

# HC32A4A8 Series

## 32-bit ARM<sup>®</sup> Cortex<sup>®</sup>-M4 Microcontroller

---

# Reference Manual

Rev1.00  
May 2025

## Statement

**Copyright ©2025 Xiaohua Semiconductor Co., Ltd. All Rights Reserved.**

All intellectual property rights related to the information contained in this document and its annexes belongs to Xiaohua Semiconductor Co., Ltd. (hereinafter referred to as "XHSC"). The customer only has the right to use the information contained in this document and its annexes. Without the written permission of XHSC, it is forbidden to extract, copy, modify, use or spread any part or all of this document in any form.

### Trademark Statement

**XHSC** XIAOHUA SEMICONDUCTOR, **XHSC**, and other trademarks are the trademarks of XHSC. All other products or services shown on XHSC products are the property of their respective owners.

### Precautions

- XHSC reserves the right to change, correct, enhance and modify the product and/or this document at any time without notice. Users can get the latest information before placing orders. XHSC products are sold in accordance with the sales terms and conditions stated in the basic purchase and sale contract.
- It is the customer's responsibility to select the appropriate XHSC product for your application and to design, validate and test your application to ensure that your application meets the appropriate standards and any safety, security or other requirements. The customer shall be solely responsible for this.
- XHSC hereby acknowledges that no intellectual property license has been granted by express or implied means.
- The resale of XHSC Products shall be invalidated by any warranty commitment of XHSC to such Products if its terms are different from those set forth herein.
- Information in this notification replaces information in previous versions.

### Xiaohua Semiconductor Co., Ltd.

|          |                                                                                                                                                                               |
|----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Address: | F4, Block A, No.1867 Zhongke Road, Pudong New Area, Shanghai, China                                                                                                           |
| URL:     | <a href="https://www.xhsc.com.cn">https://www.xhsc.com.cn</a>                                                                                                                 |
| E-mail:  | <a href="mailto:XHSC_MCU@xhsc.com.cn">XHSC_MCU@xhsc.com.cn</a> (Business Consulting)<br><a href="mailto:XHSC_FAE@xhsc.com.cn">XHSC_FAE@xhsc.com.cn</a> (Technical Consulting) |

## Preface

### Bit/Bit Field

A register contains a bit (one bit wide) or a bit field (multiple bit width), and the bit field is usually described as [y:x], starting from the x bit and ending at the y bit. Unless otherwise specified, decimal numbers are used in the bit field.

It is not recommended to use unspecified fields; otherwise, unpredictable consequences may be caused.

### Register Address

The address of each register can be obtained by the base address of the system address division and the offset address of the register (see the "Register List" of each module for details).

### Access Type

The following abbreviations are used for register description:

| Access Type | Description                                                                     |
|-------------|---------------------------------------------------------------------------------|
| RW          | Readable and writable.                                                          |
| RO          | Read only. Writes are ignored.                                                  |
| WO          | Write only. Read value undefined.                                               |
| RWA         | Readable and writable. Writable only when unlocked or in safe mode.             |
| RWC         | Readable and writable. Writing any value clears this bit.                       |
| RC          | Read only. Writes are ignored. This bit is cleared automatically after reading. |
| RS          | Read only. Writes are ignored. This bit is set automatically after reading.     |

### Number Format

- 0x prefix is used for hexadecimal numbers
- 0b prefix is used for binary numbers
- Numbers without prefix are in decimal representation

### Reserved (Res.)

Registers and bit fields marked with *Reserved* are reserved for future use. These should be written to their reset value unless otherwise stated in the Register Description. The read value of Reserved bit may be different in different devices or features.

### Reset Value (Reset)

The reset value indicates the value after the register is reset. Registers need to be initialized and reset before use; otherwise, unpredictable consequences may be caused.

### Peripheral Availability Description

For HC32A4A8 Series 32-bit ARM® Cortex®-M4 Microcontroller supported peripherals and their numbers, refer to the corresponding "Datasheet".

### Security Statement

There may be potential security problems due to the use of a certain function or protocol, which need to be declared to remind users and avoid security risks.

# Contents

|                                                          |            |
|----------------------------------------------------------|------------|
| <b>Preface</b> .....                                     | <b>iii</b> |
| <b>1 Product Introduction</b> .....                      | <b>50</b>  |
| <b>2 Memory Mapping</b> .....                            | <b>51</b>  |
| 2.1 Memory Mapping.....                                  | 51         |
| 2.2 External Space Mapping.....                          | 56         |
| 2.3 Bit Segment Space.....                               | 56         |
| 2.4 Address Remapping.....                               | 56         |
| 2.5 Register Description.....                            | 58         |
| 2.5.1 Register List.....                                 | 58         |
| 2.5.2 Access Protection Register (MMF_REMPRT).....       | 59         |
| 2.5.3 Remapping Register (MMF_REMCrx) (x=0-1).....       | 60         |
| <b>3 Bus Architecture (BUS)</b> .....                    | <b>61</b>  |
| 3.1 Introduction.....                                    | 61         |
| 3.2 Bus Architecture.....                                | 62         |
| 3.3 Bus Function.....                                    | 63         |
| <b>4 Reset Module Unit (RMU)</b> .....                   | <b>64</b>  |
| 4.1 Introduction.....                                    | 64         |
| 4.2 Functional Description.....                          | 64         |
| 4.2.1 Reset Mode and Reset Flag Bit.....                 | 64         |
| 4.2.2 Reset Timing.....                                  | 66         |
| 4.2.3 Reset Mode Judgment.....                           | 72         |
| 4.2.4 Module Function Reset.....                         | 72         |
| 4.2.5 Reset Condition.....                               | 72         |
| 4.3 Register Description.....                            | 75         |
| 4.3.1 Register List.....                                 | 75         |
| 4.3.2 Function Reset Control Register 0 (RMU_FRST0)..... | 76         |
| 4.3.3 Function Reset Control Register 1 (RMU_FRST1)..... | 78         |
| 4.3.4 Function Reset Control Register 2 (RMU_FRST2)..... | 80         |
| 4.3.5 Function Reset Control Register 3 (RMU_FRST3)..... | 82         |
| 4.3.6 Reset Control Register 0 (RMU_PRSTCR0).....        | 84         |
| 4.3.7 Reset Flag Register 0 (RMU_RSTF0).....             | 85         |
| <b>5 Clock Module Unit (CMU)</b> .....                   | <b>88</b>  |
| 5.1 Introduction.....                                    | 88         |
| 5.2 Features.....                                        | 88         |
| 5.3 Functional Description.....                          | 90         |
| 5.3.1 System Block Diagram.....                          | 90         |
| 5.3.2 Working Clock Specifications.....                  | 91         |
| 5.3.3 Crystal Oscillator Circuit.....                    | 93         |
| 5.3.4 Internal RC Clock.....                             | 97         |
| 5.3.5 PLL Clock.....                                     | 99         |
| 5.3.6 Clock Switching Steps.....                         | 99         |
| 5.3.7 Clock Output Function.....                         | 101        |



|                                                                                    |            |
|------------------------------------------------------------------------------------|------------|
| 5.3.8 Clock Frequency Measurement.....                                             | 102        |
| 5.4 Register Description.....                                                      | 104        |
| 5.4.1 Register List.....                                                           | 104        |
| 5.4.2 CMU XTAL Config Register (CMU_XTALCFGR).....                                 | 106        |
| 5.4.3 CMU XTAL Stable Control Register (CMU_XTALSTBCR).....                        | 107        |
| 5.4.4 CMU XTAL Fractional Frequency Division Config Register (CMU_XTALDIVR).....   | 108        |
| 5.4.5 CMU XTAL Fractional Frequency Division Control Register (CMU_XTALDIVCR)..... | 108        |
| 5.4.6 CMU XTAL Control Register (CMU_XTALCR).....                                  | 109        |
| 5.4.7 CMU XTAL Clock Stop Detect Control Register (CMU_XTALSTDCR).....             | 110        |
| 5.4.8 CMU XTAL Clock Stop Detect Status Register (CMU_XTALSTDSR).....              | 111        |
| 5.4.9 CMU XTAL32 Config Register (CMU_XTAL32CFGR).....                             | 112        |
| 5.4.10 CMU XTAL32 Filter Control Register (CMU_XTAL32NFR).....                     | 113        |
| 5.4.11 CMU XTAL32 Control Register (CMU_XTAL32CR).....                             | 113        |
| 5.4.12 CMU HRC Trimming Register (CMU_HRCTRM).....                                 | 114        |
| 5.4.13 CMU HRC Control Register (CMU_HRCCR).....                                   | 115        |
| 5.4.14 CMU MRC Trimming Register (CMU_MRCTRM).....                                 | 116        |
| 5.4.15 CMU MRC Control Register (CMU_MRCCR).....                                   | 117        |
| 5.4.16 CMU LRC Trimming Register (CMU_LRCTRM).....                                 | 118        |
| 5.4.17 CMU LRC Control Register (CMU_LRCCR).....                                   | 119        |
| 5.4.18 CMU RTCLRC Trimming Register (CMU_RTCLRCTRM).....                           | 120        |
| 5.4.19 CMU PLLH Config Register (CMU_PLLHCFGR).....                                | 121        |
| 5.4.20 CMU PLLH Control Register (CMU_PLLHCR).....                                 | 123        |
| 5.4.21 CMU PLLA Config Register (CMU_PLLACFGR).....                                | 124        |
| 5.4.22 CMU PLLA Control Register (CMU_PLLACR).....                                 | 126        |
| 5.4.23 CMU Clock Source Stable Status Register (CMU_OSCSTBSR).....                 | 127        |
| 5.4.24 CMU System Clock Source Switch Register (CMU_CKSWR).....                    | 128        |
| 5.4.25 CMU Clock Divider Config Register (CMU_SCFGR).....                          | 129        |
| 5.4.26 CMU USB Clock Config Register (CMU_USBCKCFGR).....                          | 132        |
| 5.4.27 CMU CAN Clock Config Register (CMU_CANCKCFGR).....                          | 133        |
| 5.4.28 CMU ADC/DAC Clock Config Register (CMU_PERICKSEL).....                      | 136        |
| 5.4.29 CMU Debug Clock Config Register (CMU_TPIUCKCFGR).....                       | 137        |
| 5.4.30 CMU I2S Clock Config Register (CMU_I2SCKCFGR).....                          | 138        |
| 5.4.31 CMU MCO1 Clock Output Config Register (CMU_MCO1CFGR).....                   | 140        |
| 5.4.32 CMU MCO2 Clock Output Config Register (CMU_MCO2CFGR).....                   | 141        |
| 5.4.33 FCM Lower Limit Control Register (FCMn_LVR) (n=1-4).....                    | 142        |
| 5.4.34 FCM Upper Limit Control Register (FCMn_UVR) (n=1-4).....                    | 142        |
| 5.4.35 FCM Counter Register (FCMn_CNTR) (n=1-4).....                               | 143        |
| 5.4.36 FCM Start Control Register (FCM_STR).....                                   | 143        |
| 5.4.37 FCM Measurement Control Register (FCMn_MCCR) (n=1-4).....                   | 144        |
| 5.4.38 FCM Reference Control Register (FCMn_RCCR) (n=1-4).....                     | 145        |
| 5.4.39 FCM Reset Interrupt Control Register (FCMn_RIER) (n=1-4).....               | 147        |
| 5.4.40 FCM Flag Register (FCM_SR).....                                             | 148        |
| 5.4.41 FCM Flag Clear Register (FCMn_CLR) (n=1-4).....                             | 150        |
| <b>6 Power Control (PWC).....</b>                                                  | <b>151</b> |
| 6.1 Introduction.....                                                              | 151        |

|                                                                          |     |
|--------------------------------------------------------------------------|-----|
| 6.2 Functional Description.....                                          | 151 |
| 6.2.1 Power Distribution.....                                            | 151 |
| 6.2.2 Power Voltage Detection Unit (PVD) description.....                | 153 |
| 6.2.3 Wake-up Timer.....                                                 | 160 |
| 6.2.4 Action Mode and Low Power Mode.....                                | 160 |
| 6.2.5 Methods to Reduce Power Consumption.....                           | 169 |
| 6.3 Register Description.....                                            | 172 |
| 6.3.1 Register List.....                                                 | 172 |
| 6.3.2 Power Mode Control Register 0 (PWC_PWRC0).....                     | 174 |
| 6.3.3 Power Mode Control Register 1 (PWC_PWRC1).....                     | 175 |
| 6.3.4 Power Mode Control Register 2 (PWC_PWRC2).....                     | 177 |
| 6.3.5 Power Mode Control Register 3 (PWC_PWRC3).....                     | 177 |
| 6.3.6 Power Mode Control Register 4 (PWC_PWRC4).....                     | 178 |
| 6.3.7 Power-down Wake-up Enable Register 0 (PWC_PDWKE0).....             | 179 |
| 6.3.8 Power-down Wake-up Enable Register 1 (PWC_PDWKE1).....             | 180 |
| 6.3.9 Power-down Wake-up Enable Register 2 (PWC_PDWKE2).....             | 181 |
| 6.3.10 Power-down Wake-up Event Edge Select Register (PWC_PDWKES).....   | 182 |
| 6.3.11 Power-down Wake-up Flag Register 0 (PWC_PDWKFO).....              | 183 |
| 6.3.12 Power-down Wake-up Flag Register 1 (PWC_PDWKFI).....              | 184 |
| 6.3.13 RAM Power Control Register 0 (PWC_RAMPC0).....                    | 185 |
| 6.3.14 RAM Operating Condition Register (PWC_RAMOPM).....                | 187 |
| 6.3.15 Peripheral RAM Low Power Control Register (PWC_PRAMLPC).....      | 188 |
| 6.3.16 Low Power Mode Status Register (PWC_LPMCSR).....                  | 190 |
| 6.3.17 Low Power Mode Status Clear Register (PWC_LPMCSCR).....           | 190 |
| 6.3.18 PVD Control Register 0 (PWC_PVDCR0).....                          | 191 |
| 6.3.19 PVD Control Register 1 (PWC_PVDCR1).....                          | 192 |
| 6.3.20 PVD Filter Control Register (PWC_PVDFCR).....                     | 194 |
| 6.3.21 PVD Level Control Register (PWC_PVDLCR).....                      | 195 |
| 6.3.22 PVD Interrupt Control Register (PWC_PVDICR).....                  | 196 |
| 6.3.23 PVD Detection Status Register (PWC_PVDDSR).....                   | 197 |
| 6.3.24 Function Clock Control Register 0 (PWC_FCG0).....                 | 198 |
| 6.3.25 Function Clock Control Register 1 (PWC_FCG1).....                 | 200 |
| 6.3.26 Function Clock Control Register 2 (PWC_FCG2).....                 | 202 |
| 6.3.27 Function Clock Control Register 3 (PWC_FCG3).....                 | 204 |
| 6.3.28 PWC_FCG0 Protection Control Register (PWC_FCG0PC).....            | 206 |
| 6.3.29 Function Protection Control Register (PWC_FPRC).....              | 207 |
| 6.3.30 STOP Mode Control Register (PWC_STPMCR).....                      | 208 |
| 6.3.31 Backup Domain Reset Register (PWC_VBATRSTR).....                  | 208 |
| 6.3.32 Backup Domain Control Register (PWC_VBATCR).....                  | 209 |
| 6.3.33 Backup Domain Power-on Flag Register (PWC_VBATPORF).....          | 210 |
| 6.3.34 Backup Domain Power-on Flag Clear Register (PWC_VBATPORFCLR)..... | 210 |
| 6.3.35 Wake-up Timer Control Register 0 (PWC_WKTC0).....                 | 211 |
| 6.3.36 Wake-up Timer Control Register 1 (PWC_WKTC1).....                 | 211 |
| 6.3.37 Wake-up Timer Control Register 2 (PWC_WKTC2).....                 | 212 |
| 6.3.38 VBAT Backup Register n (PWC_BKRn) (n=0-127).....                  | 212 |

|                                                                       |            |
|-----------------------------------------------------------------------|------------|
| <b>7 Initialization Configuration (ICG).....</b>                      | <b>213</b> |
| 7.1 Introduction.....                                                 | 213        |
| 7.2 Register Description.....                                         | 213        |
| 7.2.1 Register List.....                                              | 213        |
| 7.2.2 Initialization Configuration Register 0 (ICG0).....             | 214        |
| 7.2.3 Initialization Configuration Register 1 (ICG1).....             | 217        |
| 7.2.4 Initialization Configuration Register 3 (ICG3).....             | 218        |
| 7.2.5 Initialization Configuration Register 8x (ICG8x) (x=A/B/C)..... | 219        |
| <b>8 Embedded Flash Memory (EFM).....</b>                             | <b>220</b> |
| 8.1 Introduction.....                                                 | 220        |
| 8.2 Features.....                                                     | 220        |
| 8.3 Functional Description.....                                       | 221        |
| 8.3.1 Embedded Flash.....                                             | 221        |
| 8.3.2 Read Interface.....                                             | 222        |
| 8.3.3 Flash Read Acceleration Cache.....                              | 223        |
| 8.3.4 Flash Programming and Erasing Operations.....                   | 223        |
| 8.3.5 One-Time Programmable Byte (OTP).....                           | 228        |
| 8.3.6 Boot Swap.....                                                  | 229        |
| 8.3.7 Data Reliability and Integrity.....                             | 231        |
| 8.3.8 Interrupt.....                                                  | 232        |
| 8.4 Register Description.....                                         | 233        |
| 8.4.1 Register List.....                                              | 233        |
| 8.4.2 Flash Access Protect Register (EFM_FAPRT).....                  | 235        |
| 8.4.3 Flash Key 1 Register (EFM_KEY1).....                            | 235        |
| 8.4.4 Flash Key 2 Register (EFM_KEY2).....                            | 236        |
| 8.4.5 Flash Stop Register (EFM_FSTP).....                             | 236        |
| 8.4.6 Flash Read Mode Register (EFM_FRMC).....                        | 237        |
| 8.4.7 Flash Write Mode Register (EFM_FWMC).....                       | 238        |
| 8.4.8 Flash State Register (EFM_FSR).....                             | 240        |
| 8.4.9 Flash State Clear Register (EFM_FSCLR).....                     | 244        |
| 8.4.10 Flash Interrupt Enable Register (EFM_FITE).....                | 246        |
| 8.4.11 Flash Boot Swap Status Register (EFM_FSWP).....                | 247        |
| 8.4.12 Flash Read Mode Register 2 (EFM_FRMC2).....                    | 248        |
| 8.4.13 Chip Identification Register (EFM_CHIPID).....                 | 249        |
| 8.4.14 UNIQUE ID Register (EFM_UQID0).....                            | 249        |
| 8.4.15 UNIQUE ID Register (EFM_UQID1).....                            | 250        |
| 8.4.16 UNIQUE ID Register (EFM_UQID2).....                            | 250        |
| 8.4.17 Flash Write Lock Register (EFM_WLOCK).....                     | 251        |
| 8.4.18 Flash0 Write Protection Register0(EFM_F0NWPRT0).....           | 253        |
| 8.4.19 Flash0 Write Protection Register 1 (EFM_F0NWPRT1).....         | 253        |
| 8.4.20 Flash0 Write Protection Register 2 (EFM_F0NWPRT2).....         | 254        |
| 8.4.21 Flash0 Write Protection Register 3 (EFM_F0NWPRT3).....         | 254        |
| 8.4.22 Flash1 Write Protection Register 0 (EFM_F1NWPRT0).....         | 255        |
| 8.4.23 Flash1 Write Protection Register 1 (EFM_F1NWPRT1).....         | 255        |
| 8.4.24 Flash1 Write Protection Register 2 (EFM_F1NWPRT2).....         | 256        |

|                                                                               |            |
|-------------------------------------------------------------------------------|------------|
| 8.4.25 Flash1 Write Protection Register 3 (EFM_F1NWPRT3).....                 | 256        |
| 8.4.26 Flash Check Control Register (EFM_CKCR).....                           | 257        |
| 8.4.27 Flash Check Protection Register (EFM_CKPR).....                        | 259        |
| 8.4.28 Flash Check Status Register (EFM_CKSR).....                            | 260        |
| 8.4.29 Flash Error Injection Enable Register (EFM_EIEN).....                  | 261        |
| 8.4.30 Flash0 ECC Control Register (EFM_F0ECCCR).....                         | 261        |
| 8.4.31 Flash0 ECC Data Register (EFM_F0ECCDR).....                            | 262        |
| 8.4.32 Flash0 Error Record Register n (EFM_F0ERRn) (n=0-1).....               | 263        |
| 8.4.33 Flash0 Error Injection Enable Register n (EFM_F0EIRn) (n=0-4).....     | 264        |
| 8.4.34 Flash1 ECC Control Register (EFM_F1ECCCR).....                         | 265        |
| 8.4.35 Flash1 ECC Data Register (EFM_F1ECCDR).....                            | 265        |
| 8.4.36 Flash1 Error Log Register n (EFM_F1ERRn) (n=0-1).....                  | 266        |
| 8.4.37 Flash1 Error Injection Bit Enable Register n (EFM_F1EIRn) (n=0-4)..... | 267        |
| 8.5 Precautions.....                                                          | 268        |
| <b>9 Built-in SRAM (SRAM).....</b>                                            | <b>269</b> |
| 9.1 Introduction.....                                                         | 269        |
| 9.2 Functional Description.....                                               | 269        |
| 9.2.1 Waiting Period.....                                                     | 269        |
| 9.2.2 SRAM Space Allocation.....                                              | 270        |
| 9.2.3 ECC Check Error Injection.....                                          | 270        |
| 9.3 Register Description.....                                                 | 272        |
| 9.3.1 Register List.....                                                      | 272        |
| 9.3.2 SRAM Wait Time Control Register (SRAMC_WTCR).....                       | 274        |
| 9.3.3 SRAM Wait Time Protect Register (SRAMC_WTPR).....                       | 276        |
| 9.3.4 SRAM Check Control Register (SRAMC_CKCR).....                           | 277        |
| 9.3.5 SRAM Check Protect Register (SRAMC_CKPR).....                           | 282        |
| 9.3.6 SRAM Check Status Register (SRAMC_CKSR).....                            | 283        |
| 9.3.7 SRAM1 Error Injection Enable Register (SRAM1_EIEN).....                 | 285        |
| 9.3.8 SRAM1 Error Injection Bit Enable Register 0 (SRAM1_EIBIT0).....         | 285        |
| 9.3.9 SRAM1 Error Injection Bit Enable Register 1 (SRAM1_EIBIT1).....         | 286        |
| 9.3.10 SRAM1 Error Address Register (SRAM1_ECCERRADDR).....                   | 286        |
| 9.3.11 SRAM2 Error Injection Enable Register (SRAM2_EIEN).....                | 287        |
| 9.3.12 SRAM2 Error Injection Bit Enable Register 0 (SRAM2_EIBIT0).....        | 287        |
| 9.3.13 SRAM2 Error Injection Bit Enable Register 1 (SRAM2_EIBIT1).....        | 288        |
| 9.3.14 SRAM2 Error Address Register (SRAM2_ECCERRADDR).....                   | 288        |
| 9.3.15 SRAM3 Error Injection Enable Register (SRAM3_EIEN).....                | 289        |
| 9.3.16 SRAM3 Error Injection Bit Enable Register 0 (SRAM3_EIBIT0).....        | 289        |
| 9.3.17 SRAM3 Error Injection Bit Enable Register 1 (SRAM3_EIBIT1).....        | 290        |
| 9.3.18 SRAM3 Error Address Register (SRAM3_ECCERRADDR).....                   | 290        |
| 9.3.19 SRAM4 Error Injection Enable Register (SRAM4_EIEN).....                | 291        |
| 9.3.20 SRAM4 Error Injection Bit Enable Register 0 (SRAM4_EIBIT0).....        | 291        |
| 9.3.21 SRAM4 Error Injection Bit Enable Register 1 (SRAM4_EIBIT1).....        | 292        |
| 9.3.22 SRAM4 Error Address Register (SRAM4_ECCERRADDR).....                   | 292        |
| 9.3.23 SRAMH Error Injection Enable Register (SRAMH_EIEN).....                | 293        |
| 9.3.24 SRAMH Error Injection Bit Enable Register 0 (SRAMH_EIBIT0).....        | 293        |

|                                                                           |            |
|---------------------------------------------------------------------------|------------|
| 9.3.25 SRAMH Error Injection Bit Enable Register 1 (SRAMH_EIBIT0).....    | 294        |
| 9.3.26 SRAMH Error Address Register (SRAMH_ECCERRADDR).....               | 294        |
| 9.3.27 Ret SRAM Error Injection Enable Register (SRAMB_EIEN).....         | 295        |
| 9.3.28 Ret SRAM Error Injection Bit Enable Register 0 (SRAMB_EIBIT0)..... | 295        |
| 9.3.29 Ret SRAM Error Injection Bit Enable Register 1 (SRAMB_EIBIT1)..... | 296        |
| 9.3.30 Ret SRAM Error Address Register (SRAMB_ECCERRADDR).....            | 296        |
| 9.3.31 CACHERAM Error Injection Enable Register (CACHE_EIEN).....         | 297        |
| 9.3.32 CACHERAM Error Injection Bit Enable Register 0 (CACHE_EIBIT0)..... | 297        |
| 9.3.33 CACHERAM Error Injection Bit Enable Register 1 (CACHE_EIBIT1)..... | 298        |
| 9.3.34 CACHERAM Error Injection Bit Enable Register 2 (CACHE_EIBIT2)..... | 298        |
| 9.3.35 CACHERAM Error Injection Bit Enable Register 3 (CACHE_EIBIT3)..... | 299        |
| 9.3.36 CACHERAM Error Injection Bit Enable Register 4 (CACHE_EIBIT4)..... | 299        |
| 9.3.37 CACHERAM Error Address Register (CACHE_ECCERRADDR).....            | 301        |
| 9.4 Precautions.....                                                      | 301        |
| <b>10 General-Purpose I/O (GPIO).....</b>                                 | <b>302</b> |
| 10.1 Introduction.....                                                    | 302        |
| 10.2 Features.....                                                        | 302        |
| 10.3 Functional Description.....                                          | 303        |
| 10.3.1 Function Block Diagram.....                                        | 303        |
| 10.3.2 General Input/Output GPIO Function.....                            | 303        |
| 10.3.3 Peripheral Function.....                                           | 304        |
| 10.3.4 Dual Peripheral Function.....                                      | 304        |
| 10.3.5 Event Port Input and Output Functions.....                         | 304        |
| 10.3.6 External Interrupt EIRQ Input Function.....                        | 305        |
| 10.3.7 Analog Function.....                                               | 305        |
| 10.3.8 General Control.....                                               | 305        |
| 10.4 Register Description.....                                            | 307        |
| 10.4.1 Register List.....                                                 | 307        |
| 10.4.2 General Input Data Register (PIDRx).....                           | 309        |
| 10.4.3 General Output Data Register (PODRx).....                          | 309        |
| 10.4.4 General Output Enable Register (POERx).....                        | 310        |
| 10.4.5 General Output Set Register (POSRx).....                           | 310        |
| 10.4.6 General Output Reset Register (PORRx).....                         | 311        |
| 10.4.7 General Output Toggle Register (POTRx).....                        | 311        |
| 10.4.8 Special Control Register (PSPCR).....                              | 312        |
| 10.4.9 Common Control Register (PCCR).....                                | 313        |
| 10.4.10 Input Control Register (PINAER).....                              | 314        |
| 10.4.11 Write Protection Register (PWPR).....                             | 315        |
| 10.4.12 General Control Register (PCRxy).....                             | 316        |
| 10.4.13 Function Select Register (PFSRxy).....                            | 318        |
| 10.4.14 Event Port Direction Selection Register (PEVNTDIRRm).....         | 319        |
| 10.4.15 Event Port Input Data Register (PEVNTIDRm).....                   | 319        |
| 10.4.16 Event Port Output Data Register (PEVNTODRm).....                  | 320        |
| 10.4.17 Event Port Output Data Reset Register (PEVNTORRm).....            | 320        |
| 10.4.18 Event Port Output Data Set Register (PEVNTOSRm).....              | 322        |

|                                                                                |            |
|--------------------------------------------------------------------------------|------------|
| 10.4.19 Event Port Rising Edge Input Enable Register (PEVNTRISRm).....         | 323        |
| 10.4.20 Event Port Falling Edge Input Enable Register (PEVNTFALRm).....        | 324        |
| 10.4.21 Event Port Input Filter Control Register (PEVNTNFCR).....              | 325        |
| 10.5 Precautions.....                                                          | 326        |
| <b>11 Interrupt Controller (INTC).....</b>                                     | <b>327</b> |
| 11.1 Introduction.....                                                         | 327        |
| 11.2 Vector Digraph.....                                                       | 328        |
| 11.2.1 Interrupt Vector Digraph.....                                           | 328        |
| 11.2.2 Interrupt Event List.....                                               | 333        |
| 11.3 Functional Description.....                                               | 347        |
| 11.3.1 Function Block Diagram.....                                             | 347        |
| 11.3.2 Non-maskable Interrupt.....                                             | 347        |
| 11.3.3 External Interruption.....                                              | 348        |
| 11.3.4 Interrupt Event Multiplexer.....                                        | 348        |
| 11.3.5 Software Interrupt.....                                                 | 348        |
| 11.3.6 Event Input.....                                                        | 348        |
| 11.3.7 WFE Event Management.....                                               | 349        |
| 11.3.8 Noise Filter.....                                                       | 349        |
| 11.3.9 Low Power Mode Return.....                                              | 350        |
| 11.4 Register Description.....                                                 | 351        |
| 11.4.1 Register List.....                                                      | 351        |
| 11.4.2 External Interrupt Filter Control Register (INTC_NOCCR).....            | 352        |
| 11.4.3 Non-maskable Interrupt Enable Register (INTC_NMIER).....                | 353        |
| 11.4.4 Non-maskable Interrupt Flag Register (INTC_NMIFR).....                  | 355        |
| 11.4.5 Non-maskable Interrupt Flag Clear Register (INTC_NMIFCR).....           | 357        |
| 11.4.6 External Interrupt Control Register (INTC_EIRQCRx) (x=0-15).....        | 359        |
| 11.4.7 External Interrupt Flag Register (INTC_EIFR).....                       | 360        |
| 11.4.8 External Interrupt Flag Clear Register (INTC_EIFCR).....                | 360        |
| 11.4.9 Interrupt Event Select Register (INTC_INTSELx) (x=0-31).....            | 361        |
| 11.4.10 Interrupt Select Register (INTC_INTSELy) (y=32-127).....               | 362        |
| 11.4.11 Vector Shared Interrupt Select Register (INTC_VSSELz) (z=128-143)..... | 364        |
| 11.4.12 Stop Mode Wake-up Event Enable Register (INTC_WKEN).....               | 365        |
| 11.4.13 Software Interrupt Event Register (INTC_SWIER).....                    | 367        |
| 11.4.14 Event Enable Register (INTC_EVTER).....                                | 367        |
| 11.4.15 Interrupt Enable Register (INTC_IER).....                              | 368        |
| 11.4.16 FPU Interrupt Enable Register (INTC_FPUIER).....                       | 368        |
| 11.5 Precautions.....                                                          | 369        |
| <b>12 Automatic Operation System (AOS).....</b>                                | <b>370</b> |
| 12.1 Introduction.....                                                         | 370        |
| 12.2 Features.....                                                             | 370        |
| 12.3 Functional Description.....                                               | 371        |
| 12.3.1 Function Block Diagram.....                                             | 371        |
| 12.3.2 AOS Source Events and Targets.....                                      | 371        |
| 12.3.3 Dedicated Trigger Source.....                                           | 373        |
| 12.3.4 Common Trigger Source.....                                              | 373        |

|                                                                                                     |            |
|-----------------------------------------------------------------------------------------------------|------------|
| 12.3.5 Programmable Logic Unit (PLU).....                                                           | 374        |
| 12.4 Register Description.....                                                                      | 375        |
| 12.4.1 Register List.....                                                                           | 375        |
| 12.4.2 Software Trigger Event Register (INTSFTTRG).....                                             | 377        |
| 12.4.3 DCU Trigger Source Selection Register (DCU_TRGSELx) (x=1-4).....                             | 378        |
| 12.4.4 DMA1 Transfer Start Trigger Source Selection Register (DMA1_TRGSELx) (x=0-7).....            | 379        |
| 12.4.5 DMA2 Transfer Start Trigger Source Selection Register (DMA2_TRGSELx) (x=0-7).....            | 380        |
| 12.4.6 DMA Re-config Trigger Source Selection Register (DMA_RC_TRGSEL).....                         | 381        |
| 12.4.7 Timer6 Hardware Trigger Event Selection Register (TMR6_TRGSELx) (x=0-3).....                 | 382        |
| 12.4.8 Event Port Trigger Source Selection Register (PEVNT_TRGSEL12/PEVNT_TRGSEL34).....            | 383        |
| 12.4.9 Timer0 Hardware Trigger Event Selection Register (TMR0_TRGSEL).....                          | 384        |
| 12.4.10 Timer2 Hardware Trigger Event Selection Register (TMR2_TRGSEL).....                         | 385        |
| 12.4.11 HASH Hardware Trigger Event Selection Register (HASH_TRGSELx) (x=A/B).....                  | 386        |
| 12.4.12 TimerA Hardware Trigger Event Selection Register (TMRA_TRGSELx) (x=0-3).....                | 388        |
| 12.4.13 OTS Hardware Trigger Event Selection Register (OTS_TRGSEL).....                             | 389        |
| 12.4.14 ADC1 Conversion Start On-Chip Trigger Source Selection Register (ADC1_TRGSELx) (x=0-1)..... | 390        |
| 12.4.15 ADC2 Conversion Start On-Chip Trigger Source Selection Register (ADC2_TRGSELx) (x=0-1)..... | 391        |
| 12.4.16 ADC3 Conversion Start On-Chip Trigger Source Selection Register (ADC3_TRGSELx) (x=0-1)..... | 392        |
| 12.4.17 Common Trigger Source Selection Register (AOS_COMTRGSELx) (x=1-2).....                      | 393        |
| 12.4.18 PLU Control Register (AOS_PLUX_CR) (x=0-3).....                                             | 394        |
| 12.4.19 PLU Trigger Source Selection Register A (AOS_PLUX_TRGSELA) (x=0-3).....                     | 396        |
| 12.4.20 PLU Trigger Source Selection Register B (AOS_PLUX_TRGSELB) (x=0-3).....                     | 396        |
| 12.4.21 PLU Trigger Source Selection Register C (AOS_PLUX_TRGSELC) (x=0-3).....                     | 397        |
| 12.4.22 PLU Trigger Source Selection Register D (AOS_PLUX_TRGSELD) (x=0-3).....                     | 397        |
| <b>13 Memory Protection Unit (MPU).....</b>                                                         | <b>398</b> |
| 13.1 Introduction.....                                                                              | 398        |
| 13.2 Functional Description.....                                                                    | 399        |
| 13.2.1 Locale Setting.....                                                                          | 399        |
| 13.2.2 Permission Setting.....                                                                      | 399        |
| 13.2.3 MPU Action Selection.....                                                                    | 399        |
| 13.2.4 Start MPU.....                                                                               | 399        |
| 13.3 Operation Example.....                                                                         | 400        |
| 13.3.1 Only Enable Partial Space Access.....                                                        | 400        |
| 13.3.2 Only Disable Partial Space Access.....                                                       | 400        |
| 13.4 Register Description.....                                                                      | 401        |
| 13.4.1 Register List.....                                                                           | 401        |
| 13.4.2 Area Region Description Register MPU_RGDn (n=0-15).....                                      | 403        |
| 13.4.3 Status Register (MPU_SR).....                                                                | 404        |
| 13.4.4 Flag Clear Register (MPU_ECLR).....                                                          | 405        |
| 13.4.5 Write Protection Register (MPU_WP).....                                                      | 406        |
| 13.4.6 IP Access Protection Register (MPU_IPPR).....                                                | 407        |
| 13.4.7 SMPU1 Region Enable Register (MPU_S1RGE).....                                                | 410        |
| 13.4.8 SMPU1 Region Write Permission Register (MPU_S1RGWP).....                                     | 412        |
| 13.4.9 SMPU1 Region Read Permission Register (MPU_S1RGRP).....                                      | 414        |
| 13.4.10 SMPU1 Control Register (MPU_S1CR).....                                                      | 416        |



|                                                                                                                               |            |
|-------------------------------------------------------------------------------------------------------------------------------|------------|
| 13.4.11 SMPU2 Region Enable Register (MPU_S2RGE).....                                                                         | 417        |
| 13.4.12 SMPU2 Region Write Permission Register (MPU_S2RGWP).....                                                              | 419        |
| 13.4.13 SMPU2 Region Read Permission Register (MPU_S2RGRP).....                                                               | 421        |
| 13.4.14 SMPU2 Control Register (MPU_S2CR).....                                                                                | 423        |
| 13.4.15 FMPU Region Enable Register (MPU_FRGE).....                                                                           | 424        |
| 13.4.16 FMPU Region Write Permission Register (MPU_FRGWP).....                                                                | 425        |
| 13.4.17 FMPU Region Read Permission Register (MPU_FRGRP).....                                                                 | 426        |
| 13.4.18 FMPU Control Register (MPU_FCR).....                                                                                  | 427        |
| 13.4.19 HMPU Region Enable Register (MPU_HRGE).....                                                                           | 428        |
| 13.4.20 HMPU Region Write Permission Register (MPU_HRGWP).....                                                                | 429        |
| 13.4.21 HMPU Region Read Permission Register (MPU_HRGRP).....                                                                 | 430        |
| 13.4.22 HMPU Control Register (MPU_HCR).....                                                                                  | 431        |
| 13.4.23 EMPU Region Enable Register (MPU_ERGE).....                                                                           | 432        |
| 13.4.24 EMPU Region Write Permission Register (MPU_ERGWP).....                                                                | 433        |
| 13.4.25 EMPU Region Read Permission Register (MPU_ERGRP).....                                                                 | 434        |
| 13.4.26 EMPU Control Register (MPU_ECR).....                                                                                  | 435        |
| <b>14 Keyboard Scan Control Module (KEYSCAN).....</b>                                                                         | <b>436</b> |
| 14.1 Introduction.....                                                                                                        | 436        |
| 14.2 Features.....                                                                                                            | 436        |
| 14.3 Functional Description.....                                                                                              | 437        |
| 14.3.1 Function Block Diagram.....                                                                                            | 437        |
| 14.3.2 Key Recognition Function.....                                                                                          | 437        |
| 14.3.3 Keyboard Scan Function.....                                                                                            | 438        |
| 14.4 Register Description.....                                                                                                | 439        |
| 14.4.1 Register List.....                                                                                                     | 439        |
| 14.4.2 Scan Control Register (KEYSCAN_SCR).....                                                                               | 440        |
| 14.4.3 Scan Enable Register (KEYSCAN_SER).....                                                                                | 441        |
| 14.4.4 Scan Status Register (KEYSCAN_SSR).....                                                                                | 442        |
| 14.5 Precautions.....                                                                                                         | 443        |
| <b>15 Clock Trimming Controller (CTC).....</b>                                                                                | <b>444</b> |
| 15.1 Introduction.....                                                                                                        | 444        |
| 15.2 Features.....                                                                                                            | 444        |
| 15.3 Functional Description.....                                                                                              | 444        |
| 15.3.1 Function Block Diagram.....                                                                                            | 444        |
| 15.3.2 Reference Clock.....                                                                                                   | 444        |
| 15.3.3 Frequency Calibration.....                                                                                             | 446        |
| 15.3.4 Interrupt.....                                                                                                         | 447        |
| 15.4 Operation Example.....                                                                                                   | 447        |
| 15.4.1 Use XTAL32 as the Reference Clock to Calibrate HRC to 20MHz with a Calibration Target Deviation of Less Than 0.5%..... | 447        |
| 15.5 Register Description.....                                                                                                | 449        |
| 15.5.1 Register List.....                                                                                                     | 449        |
| 15.5.2 CTC Control Register 1 (CTC_CR1).....                                                                                  | 450        |
| 15.5.3 CTC Control Register 2 (CTC_CR2).....                                                                                  | 452        |
| 15.5.4 CTC Status Register (CTC_STR).....                                                                                     | 453        |



|                                                                                     |            |
|-------------------------------------------------------------------------------------|------------|
| <b>16 DMA Controller (DMA)</b>                                                      | <b>454</b> |
| 16.1 Introduction                                                                   | 454        |
| 16.2 Features                                                                       | 454        |
| 16.3 Functional Description                                                         | 455        |
| 16.3.1 DMA Block Diagram                                                            | 455        |
| 16.3.2 Enabling the DMA Controller                                                  | 455        |
| 16.3.3 Channel Selection and Channel Priority                                       | 455        |
| 16.3.4 Start DMA                                                                    | 456        |
| 16.3.5 Data Block                                                                   | 456        |
| 16.3.6 Transfer Address Control                                                     | 456        |
| 16.3.7 Transfer Count                                                               | 456        |
| 16.3.8 Chain Transmission                                                           | 457        |
| 16.3.9 NonSequence Address Transfer                                                 | 458        |
| 16.3.10 Channel Reset                                                               | 459        |
| 16.3.11 Early Termination of Transfer                                               | 461        |
| 16.3.12 Interrupt                                                                   | 461        |
| 16.4 Operation Example                                                              | 462        |
| 16.4.1 Memory-to-Memory Transfer                                                    | 462        |
| 16.4.2 Memory-to-Peripheral Transfer                                                | 463        |
| 16.4.3 Memory-to-Memory Chain Transfer                                              | 464        |
| 16.5 Register Description                                                           | 467        |
| 16.5.1 Register List                                                                | 467        |
| 16.5.2 DMA Enable Register (DMA_EN)                                                 | 469        |
| 16.5.3 Interrupt Status Register 0 (DMA_INTSTAT0)                                   | 470        |
| 16.5.4 Interrupt Status Register 1 (DMA_INTSTAT1)                                   | 471        |
| 16.5.5 Interrupt Mask Register 0 (DMA_INTMASK0)                                     | 472        |
| 16.5.6 Interrupt Mask Register 1 (DMA_INTMASK1)                                     | 473        |
| 16.5.7 Interrupt Status Clear Register 0 (DMA_INTCLR0)                              | 474        |
| 16.5.8 Interrupt Status Clear Register 1 (DMA_INTCLR1)                              | 475        |
| 16.5.9 Channel Enable Register (DMA_CHEN)                                           | 476        |
| 16.5.10 Transfer Request Status Register (DMA_REQSTAT)                              | 477        |
| 16.5.11 Channel Status Register (DMA_CHSTAT)                                        | 478        |
| 16.5.12 Channel Reset Control Register (DMA_RCFGCTL)                                | 479        |
| 16.5.13 Software Trigger Register (DMA_SWREQ)                                       | 481        |
| 16.5.14 Channel Enable Reset Register (DMA_CHENCLR)                                 | 482        |
| 16.5.15 Transmission Source Address Register (DMA_SARx) (x=0-7)                     | 482        |
| 16.5.16 Transmission Destination Address Register (DMA_DARx) (x=0-7)                | 483        |
| 16.5.17 Transfer Data Control Register (DMA_DTCTLx) (x=0-7)                         | 484        |
| 16.5.18 Address Repeat Control Register (DMA_RPTx) (x=0-7)                          | 485        |
| 16.5.19 Address Repeat Control Register B (DMA_RPTBx) (x=0-7)                       | 486        |
| 16.5.20 Source Address NonSequence Control Register (DMA_SNSEQCTLx) (x=0-7)         | 487        |
| 16.5.21 Source Address NonSequence Control Register B (DMA_SNSEQCTLBx) (x=0-7)      | 488        |
| 16.5.22 Destination Address NonSequence Control Register (DMA_DNSEQCTLx) (x=0-7)    | 489        |
| 16.5.23 Destination Address NonSequence Control Register B (DMA_DNSEQCTLBx) (x=0-7) | 491        |
| 16.5.24 Linked List Pointer Register (DMA_LL Px) (x=0-7)                            | 492        |

|                                                                                                   |            |
|---------------------------------------------------------------------------------------------------|------------|
| 16.5.25 Channel Control Register (DMA_CHCTLx) (x=0-7).....                                        | 493        |
| 16.5.26 Source Address Monitor Register (DMA_MONSARx) (x=0-7).....                                | 495        |
| 16.5.27 Destination Address Monitor Register (DMA_MONDARx) (x=0-7).....                           | 496        |
| 16.5.28 Transfer Data Control Monitor Register (DMA_MONDTCTLx) (x=0-7).....                       | 497        |
| 16.5.29 Address Repeat Control Monitor Register (DMA_MONRPTx) (x=0-7).....                        | 498        |
| 16.5.30 Source Address NonSequence Control Monitor Register (DMA_MONSNSEQCTLx) (x=0-7).....       | 499        |
| 16.5.31 Destination Address NonSequence Control Monitor Register (DMA_MONDNSEQCTLx) (x=0-7).....  | 500        |
| 16.6 Precautions.....                                                                             | 500        |
| <b>17 Voltage Comparator (CMP).....</b>                                                           | <b>501</b> |
| 17.1 Introduction.....                                                                            | 501        |
| 17.2 Features.....                                                                                | 501        |
| 17.3 Functional Description.....                                                                  | 502        |
| 17.3.1 Function Block Diagram.....                                                                | 502        |
| 17.3.2 Input Selection.....                                                                       | 503        |
| 17.3.3 Normal Compare.....                                                                        | 503        |
| 17.3.4 Window Compare.....                                                                        | 504        |
| 17.3.5 Blank Window.....                                                                          | 505        |
| 17.3.6 Hysteresis Function.....                                                                   | 506        |
| 17.3.7 Noise Filter.....                                                                          | 506        |
| 17.3.8 External Pin Output.....                                                                   | 506        |
| 17.3.9 Interrupt and Event.....                                                                   | 506        |
| 17.4 Register Description.....                                                                    | 508        |
| 17.4.1 Register List.....                                                                         | 508        |
| 17.4.2 Comparator Working Mode Register (CMP_MDR).....                                            | 509        |
| 17.4.3 Comparator Filter and Interrupt Register (CMP_FIR).....                                    | 511        |
| 17.4.4 Comparator Output Control Register (CMP_OCR).....                                          | 512        |
| 17.4.5 Comparator Positive and Negative Input Selection Register (CMP_PMSR).....                  | 513        |
| 17.4.6 Comparator Voltage Input Selection Register (CMP_VISR) (Only exists in CMP1 and CMP3)..... | 514        |
| 17.4.7 Comparator Blank Window Output Source Register (CMP_BWSR).....                             | 515        |
| 17.4.8 Comparator Blank Window Output Polarity Register (CMP_BWPR).....                           | 516        |
| 17.5 Precautions.....                                                                             | 517        |
| 17.5.1 Module Stop Function.....                                                                  | 517        |
| 17.5.2 Power Consumption During Module Stop.....                                                  | 517        |
| 17.5.3 Power Consumption During Stop Low Power Mode.....                                          | 517        |
| <b>18 Analog-to-Digital Converter (ADC).....</b>                                                  | <b>518</b> |
| 18.1 Introduction.....                                                                            | 518        |
| 18.2 Features.....                                                                                | 518        |
| 18.3 Functional Description.....                                                                  | 519        |
| 18.3.1 Function Block Diagram.....                                                                | 519        |
| 18.3.2 ADC Clock.....                                                                             | 520        |
| 18.3.3 Channel Selection.....                                                                     | 521        |
| 18.3.4 Trigger Source Selection.....                                                              | 522        |
| 18.3.5 Sequence A Single Scan Mode.....                                                           | 522        |
| 18.3.6 Sequence A Continuous Scan Mode.....                                                       | 523        |
| 18.3.7 Sequence A Data Buffer Mode.....                                                           | 524        |

|                                                                                                    |            |
|----------------------------------------------------------------------------------------------------|------------|
| 18.3.8 Dual Sequence Scan Mode.....                                                                | 525        |
| 18.3.9 Analog Watchdog Function.....                                                               | 527        |
| 18.3.10 Sampling Time and Conversion Time for Analog Inputs.....                                   | 528        |
| 18.3.11 ADC Data Register Automatic Clear Function.....                                            | 529        |
| 18.3.12 Conversion Data Averaging Function.....                                                    | 529        |
| 18.3.13 Oversampling.....                                                                          | 530        |
| 18.3.14 Programmable Gain Amplifier PGA.....                                                       | 530        |
| 18.3.15 Dedicated Sample Hold Circuit (SH) for Channels.....                                       | 531        |
| 18.3.16 Multi-ADC Cooperative Working Mode.....                                                    | 532        |
| 18.3.17 Interrupt and Events.....                                                                  | 537        |
| 18.4 Register Description.....                                                                     | 538        |
| 18.4.1 Register List.....                                                                          | 538        |
| 18.4.2 Start Register (ADC_STR).....                                                               | 540        |
| 18.4.3 Control Register 0 (ADC_CR0).....                                                           | 541        |
| 18.4.4 Control Register 1 (ADC_CR1).....                                                           | 543        |
| 18.4.5 Control Register 2 (ADC_CR2).....                                                           | 544        |
| 18.4.6 Conversion Start Trigger Register (ADC_TRGSR).....                                          | 545        |
| 18.4.7 Channel Selection Register A (ADC_CHSELRA).....                                             | 547        |
| 18.4.8 Channel Selection Register B (ADC_CHSELRB).....                                             | 548        |
| 18.4.9 Average Channel Selection Register (ADC_AVCHSELR).....                                      | 549        |
| 18.4.10 Extended Channel Selection Register (ADC_EXCHSELR).....                                    | 550        |
| 18.4.11 Sample Cycle Register (ADC_SSTRx) (x=0-15, L).....                                         | 551        |
| 18.4.12 Channel Mapping Control Register (ADC_CHMUXRx) (x=0-3).....                                | 552        |
| 18.4.13 Interrupt Status Register (ADC_ISR).....                                                   | 554        |
| 18.4.14 Interrupt Enable Register (ADC_ICR).....                                                   | 555        |
| 18.4.15 Interrupt Status Clear Register (ADC_ISCLRR).....                                          | 556        |
| 18.4.16 Cooperative Mode Control Register (ADC_SYNCCR).....                                        | 557        |
| 18.4.17 ADC Data Register (ADC_DRx).....                                                           | 559        |
| 18.4.18 Analog Watchdog Control Register (ADC_AWDCR).....                                          | 561        |
| 18.4.19 Analog Watchdog Status Register (ADC_AWDSR).....                                           | 563        |
| 18.4.20 Analog Watchdog Status Clear Register (ADC_AWDSCLRR).....                                  | 564        |
| 18.4.21 Analog Watchdog Window x Threshold Register (ADC_AWDxDy) (x=0-1, y=0-1).....               | 565        |
| 18.4.22 Analog Watchdog Window x Comparator Channel Selection Register (ADC_AWDxCHSR) (x=0-1)..... | 566        |
| 18.4.23 Dedicated Sample Hold Circuit Control Register (ADC_SHCR).....                             | 567        |
| 18.4.24 Programmable Gain Amplifier Control Register (ADC_PGACRx) (x=1-3).....                     | 568        |
| 18.4.25 Programmable Gain Amplifier Reference Ground Selection Register (ADC_PGAVSSENr).....       | 569        |
| 18.5 Precautions.....                                                                              | 570        |
| 18.5.1 Precautions for Reading Data Registers.....                                                 | 570        |
| 18.5.2 Precautions for Handling Scan Completion Interrupts.....                                    | 570        |
| 18.5.3 Precautions for Module Stop and Low Power Settings.....                                     | 570        |
| 18.5.4 ADC Conversion Analog Channel Input Pin Settings.....                                       | 570        |
| 18.5.5 Noise Control.....                                                                          | 570        |
| <b>19 Digital-to-Analog Converter (DAC).....</b>                                                   | <b>571</b> |
| 19.1 Introduction.....                                                                             | 571        |
| 19.2 Features.....                                                                                 | 571        |

|                                                                    |            |
|--------------------------------------------------------------------|------------|
| 19.3 Functional Description.....                                   | 571        |
| 19.3.1 Function Block Diagram.....                                 | 571        |
| 19.3.2 DAC Conversion.....                                         | 572        |
| 19.3.3 Synchronization Conversion.....                             | 572        |
| 19.3.4 External Data Conversion.....                               | 573        |
| 19.3.5 ADC Conversion Priority Mode.....                           | 573        |
| 19.4 Register Description.....                                     | 575        |
| 19.4.1 Register List.....                                          | 575        |
| 19.4.2 Data Register (DAC_DADRY) (y=1-2).....                      | 576        |
| 19.4.3 Control Register (DAC_DACR).....                            | 577        |
| 19.4.4 Analog Output Control Register (DAC_DAOCR).....             | 579        |
| 19.4.5 ADC Conversion Priority Control Register (DAC_DAADPCR)..... | 580        |
| 19.5 Precautions.....                                              | 581        |
| 19.5.1 Module Stop Function Settings.....                          | 581        |
| 19.5.2 DAC Action in Module Stop State.....                        | 581        |
| 19.5.3 DAC Action in Stop Low Power Mode.....                      | 581        |
| 19.5.4 DAC Action in Power-Down Low Power Mode.....                | 581        |
| 19.5.5 Using the Output Amplifier.....                             | 581        |
| <b>20 On-chip Temperature Sensor (OTS).....</b>                    | <b>582</b> |
| 20.1 Introduction.....                                             | 582        |
| 20.2 Functional Description.....                                   | 582        |
| 20.2.1 Function Block Diagram.....                                 | 582        |
| 20.2.2 Usage Method.....                                           | 582        |
| 20.3 Register Description.....                                     | 585        |
| 20.3.1 Register List.....                                          | 585        |
| 20.3.2 OTS Control Register (OTS_CTL).....                         | 586        |
| 20.3.3 OTS Data Register 1 (OTS_DR1).....                          | 587        |
| 20.3.4 OTS Data Register 2 (OTS_DR2).....                          | 587        |
| 20.3.5 OTS Error Compensation Register (OTS_ECR).....              | 588        |
| 20.3.6 OTS Preset Temperature Data Register 1 (OTS_PDR1).....      | 588        |
| 20.3.7 OTS Preset Temperature Data Register 2 (OTS_PDR2).....      | 589        |
| 20.3.8 OTS Preset Temperature Data Register 3 (OTS_PDR3).....      | 590        |
| <b>21 Advanced Control Timer (Timer6).....</b>                     | <b>591</b> |
| 21.1 Introduction.....                                             | 591        |
| 21.2 Features.....                                                 | 591        |
| 21.3 Functional Description.....                                   | 592        |
| 21.3.1 Timer6 Block Diagram.....                                   | 592        |
| 21.3.2 Basic Actions.....                                          | 593        |
| 21.3.3 Clock Source Selection.....                                 | 595        |
| 21.3.4 Count Direction.....                                        | 595        |
| 21.3.5 Counter Refresh.....                                        | 595        |
| 21.3.6 Digital Filtering.....                                      | 596        |
| 21.3.7 Software Synchronization.....                               | 597        |
| 21.3.8 Hardware Synchronization.....                               | 598        |
| 21.3.9 Pulse Width Measurement.....                                | 600        |

|                                                                            |     |
|----------------------------------------------------------------------------|-----|
| 21.3.10 Period Measurement.....                                            | 601 |
| 21.3.11 Buffer Function.....                                               | 601 |
| 21.3.12 General PWM Output.....                                            | 607 |
| 21.3.13 Quadrature Encoding Count.....                                     | 612 |
| 21.3.14 Periodic Interval Response.....                                    | 617 |
| 21.3.15 EMB Control.....                                                   | 617 |
| 21.3.16 PTO Function.....                                                  | 618 |
| 21.3.17 Function Summary Table.....                                        | 626 |
| 21.3.18 Interrupt and Events.....                                          | 627 |
| 21.4 Operation Example.....                                                | 629 |
| 21.4.1 Basic Counting and Interrupt Action.....                            | 629 |
| 21.4.2 Compare Output and Interrupt Action.....                            | 629 |
| 21.4.3 Capture Input and Interrupt Action.....                             | 629 |
| 21.4.4 Buffer Transmission Action (Period Reference Value).....            | 630 |
| 21.4.5 Buffer Transmission Action (General Compare Reference Value).....   | 630 |
| 21.4.6 Buffer Transmission Action (Special Compare Reference Value).....   | 630 |
| 21.4.7 Buffer Transmission Action (Dead Zone Reference Value).....         | 631 |
| 21.4.8 Synchronization Start Action (Software Mode).....                   | 631 |
| 21.4.9 Synchronization Start Action (Hardware Mode).....                   | 631 |
| 21.4.10 Quadrature Encoding Count Action (2-phase).....                    | 632 |
| 21.4.11 Quadrature Encoding Count Action (3-phase).....                    | 632 |
| 21.4.12 Single Channel PWM Output.....                                     | 633 |
| 21.4.13 Complementary PWM Output (Software Dead Zone).....                 | 633 |
| 21.4.14 Complementary PWM Output (Hardware Dead Zone).....                 | 633 |
| 21.4.15 EMB Monitoring and Interrupt Action.....                           | 633 |
| 21.5 Register Description.....                                             | 635 |
| 21.5.1 Register List.....                                                  | 635 |
| 21.5.2 General Count Value Register (TMR6_CNTER).....                      | 639 |
| 21.5.3 General Refresh Value Register (TMR6_UPDAR).....                    | 639 |
| 21.5.4 General Period Reference Value Register (TMR6_PERmR) (m=A-C).....   | 640 |
| 21.5.5 General Compare Reference Value Register (TMR6_GCMmR) (m=A-F).....  | 640 |
| 21.5.6 Special Compare Reference Value Register (TMR6_SCMmR) (m=A-F).....  | 641 |
| 21.5.7 Dead Time Reference Value Register (TMR6_DTmnR) (m=U/D, n=A/B)..... | 641 |
| 21.5.8 General Control Register (TMR6_GCONR).....                          | 642 |
| 21.5.9 Interrupt Control Register (TMR6_ICONR).....                        | 644 |
| 21.5.10 Buffer Control Register (TMR6_BCONR).....                          | 646 |
| 21.5.11 Dead Zone Control Register (TMR6_DCONR).....                       | 650 |
| 21.5.12 Port Control Register A (TMR6_PCNAR).....                          | 651 |
| 21.5.13 Port Control Register B (TMR6_PCNBR).....                          | 655 |
| 21.5.14 Filter Control Register (TMR6_FCNGR).....                          | 659 |
| 21.5.15 Valid Period Register (TMR6_VPERR).....                            | 660 |
| 21.5.16 Status Flag Register (TMR6_STFLR).....                             | 661 |
| 21.5.17 Hardware Start Event Selection Register (TMR6_HSTAR).....          | 663 |
| 21.5.18 Hardware Stop Event Selection Register (TMR6_HSTPR).....           | 666 |
| 21.5.19 Hardware Clear Event Selection Register (TMR6_HCLRR).....          | 669 |

|                                                                                    |            |
|------------------------------------------------------------------------------------|------------|
| 21.5.20 Hardware Refresh Event Selection Register (TMR6_HUPDR).....                | 672        |
| 21.5.21 Hardware Capture Event Selection Register A (TMR6_HCPAR).....              | 675        |
| 21.5.22 Hardware Capture Event Selection Register B (TMR6_HCPBR).....              | 678        |
| 21.5.23 Hardware Count Up Event Select Register (TMR6_HCUPR).....                  | 681        |
| 21.5.24 Hardware Count Down Event Select Register (TMR6_HCDOR).....                | 685        |
| 21.5.25 PTO Pulse Number Register (TMR6_PTNUMR).....                               | 689        |
| 21.5.26 PTO Pulse Period Register (TMR6_PTPERR).....                               | 689        |
| 21.5.27 PTO Algorithm Period Remainder Register (TMR6_PTRMDR).....                 | 690        |
| 21.5.28 PTO Mode Control Register (TMR6_PTCTLR).....                               | 691        |
| 21.5.29 PTO A-phase Output Position Register (TMR6_PTAOPR).....                    | 691        |
| 21.5.30 PTO B-phase Output Position Register (TMR6_PTBOPR).....                    | 692        |
| 21.5.31 PTO Z-phase Output Start Position Register (TMR6_PTZOSPR).....             | 692        |
| 21.5.32 PTO Z-phase Output End Position Register (TMR6_PTZOEPR).....               | 692        |
| 21.5.33 PTO Z-phase Output Pulse Width Compression B Register (TMR6_PTZCMPBR)..... | 693        |
| 21.5.34 PTO Full Circle Pulse Count Register (TMR6_PTZPERR).....                   | 693        |
| 21.5.35 PTO Pulse Initial Position Register (TMR6_PTPSPR).....                     | 694        |
| 21.5.36 PTO Pulse Initial Port Register (TMR6_PTPSTR).....                         | 694        |
| 21.5.37 PTO Pulse Current Position Register (TMR6_PTPCPR).....                     | 695        |
| 21.5.38 PTO Pulse Current Port Register (TMR6_PTPCTR).....                         | 695        |
| 21.5.39 Common Filter Control Register (TMR6_FCNTN).....                           | 696        |
| 21.5.40 Common Software Synchronous Start Control Register (TMR6_SSTAR).....       | 698        |
| 21.5.41 Common Software Synchronous Stop Control Register (TMR6_SSTPR).....        | 700        |
| 21.5.42 Common Software Synchronous Clear Control Register (TMR6_SCLRR).....       | 701        |
| 21.5.43 Common Software Synchronous Refresh Control Register (TMR6_SUPDR).....     | 702        |
| 21.6 Precautions.....                                                              | 703        |
| <b>22 General Control Timer (Timer4).....</b>                                      | <b>706</b> |
| 22.1 Introduction.....                                                             | 706        |
| 22.2 Features.....                                                                 | 706        |
| 22.3 Functional Description.....                                                   | 707        |
| 22.3.1 Timer4 Block Diagram.....                                                   | 707        |
| 22.3.2 Basic Actions.....                                                          | 708        |
| 22.3.3 Buffer Function.....                                                        | 711        |
| 22.3.4 General PWM Output.....                                                     | 717        |
| 22.3.5 Periodic Interval Response.....                                             | 722        |
| 22.3.6 EMB Control.....                                                            | 724        |
| 22.3.7 Monitoring Output.....                                                      | 725        |
| 22.3.8 Synchronous Starting.....                                                   | 726        |
| 22.3.9 Interrupt and Event.....                                                    | 726        |
| 22.4 Register Description.....                                                     | 728        |
| 22.4.1 Register List.....                                                          | 728        |
| 22.4.2 Count Value Register (TMR4_CNTR).....                                       | 731        |
| 22.4.3 Period Reference Register (TMR4_CPSR).....                                  | 731        |
| 22.4.4 Control Status Register (TMR4_CCSR).....                                    | 732        |
| 22.4.5 Valid Period Register (TMR4_CVPR).....                                      | 734        |
| 22.4.6 General Compare Reference Register (TMR4_OCCRM).....                        | 735        |

|                                                                                                       |            |
|-------------------------------------------------------------------------------------------------------|------------|
| 22.4.7 General Control Status Register (TMR4_OCSRn).....                                              | 736        |
| 22.4.8 General Extended Control Register (TMR4_OCERn).....                                            | 738        |
| 22.4.9 General Extended Control Register (TMR4_OCMRm).....                                            | 741        |
| 22.4.10 Special Compare Reference Register (TMR4_SCCRm).....                                          | 749        |
| 22.4.11 General Control Status Register (TMR4_SCSRm).....                                             | 750        |
| 22.4.12 Special Extended Control Register (TMR4_SCER).....                                            | 752        |
| 22.4.13 Special Mode Control Register (TMR4_SCMRm).....                                               | 753        |
| 22.4.14 PWM Control Register (TMR4_POCRn).....                                                        | 754        |
| 22.4.15 PWM Status Control Register (TMR4_PSCR).....                                                  | 755        |
| 22.4.16 PWM Filter Control Register (TMR4_PFSRn).....                                                 | 759        |
| 22.4.17 PWM Dead Zone Control Register A (TMR4_PDARn).....                                            | 760        |
| 22.4.18 PWM Dead Zone Control Register B (TMR4_PDBRn).....                                            | 760        |
| 22.4.19 Reload Control Status Register (TMR4_RCSR).....                                               | 761        |
| <b>23 Emergency Brake Module (EMB).....</b>                                                           | <b>763</b> |
| 23.1 Introduction.....                                                                                | 763        |
| 23.2 Functional Description.....                                                                      | 763        |
| 23.2.1 EMB Channel Function Block Diagram.....                                                        | 763        |
| 23.2.2 EMB Structure Block Diagram.....                                                               | 765        |
| 23.2.3 Control PWM Signal Output When External Port Input Level Changes.....                          | 765        |
| 23.2.4 Stop PWM Signal Output When PWM Output Port Level Occurs In-phase (same high or same low)..... | 766        |
| 23.2.5 Stop PWM Signal Output According to Voltage Comparator Compare Result.....                     | 767        |
| 23.2.6 Stop PWM Signal Output when System Error Occurs.....                                           | 768        |
| 23.2.7 Write Register Software Control PWM Signal Output.....                                         | 768        |
| 23.3 Register Description.....                                                                        | 769        |
| 23.3.1 Register List.....                                                                             | 769        |
| 23.3.2 EMB Control Register 1 (EMB_CTL1) (Group0-3).....                                              | 770        |
| 23.3.3 EMB Control Register 1 (EMB_CTL1) (Group4-6).....                                              | 773        |
| 23.3.4 EMB Control Register 2 (EMB_CTL2) (Group0-3).....                                              | 776        |
| 23.3.5 EMB Control Register 2 (EMB_CTL2) (Group4-6).....                                              | 778        |
| 23.3.6 EMB Software Output Enable Control Register (EMB_SOE).....                                     | 779        |
| 23.3.7 EMB Status Register (EMB_STAT).....                                                            | 780        |
| 23.3.8 EMB Status Reset Register (EMB_STATCLR).....                                                   | 782        |
| 23.3.9 EMB Interrupt License Register (EMB_INTEN).....                                                | 783        |
| 23.3.10 EMB Control PWM Output Release Mode Selection Register (EMB_RLSSEL).....                      | 784        |
| <b>24 General Timer (TimerA).....</b>                                                                 | <b>785</b> |
| 24.1 Introduction.....                                                                                | 785        |
| 24.2 Features.....                                                                                    | 785        |
| 24.3 Functional Description.....                                                                      | 786        |
| 24.3.1 TimerA Block Diagram.....                                                                      | 786        |
| 24.3.2 Symmetric Units.....                                                                           | 787        |
| 24.3.3 Basic Actions.....                                                                             | 787        |
| 24.3.4 Clock Source Selection.....                                                                    | 790        |
| 24.3.5 Count Direction.....                                                                           | 790        |
| 24.3.6 Digital Filtering.....                                                                         | 791        |
| 24.3.7 Synchronous Starting.....                                                                      | 791        |



|                                                                        |            |
|------------------------------------------------------------------------|------------|
| 24.3.8 Buffer Function.....                                            | 792        |
| 24.3.9 Cascade Counting.....                                           | 793        |
| 24.3.10 General PWM Output.....                                        | 794        |
| 24.3.11 Quadrature Encoding Count.....                                 | 795        |
| 24.3.12 Interrupt and Events.....                                      | 798        |
| 24.4 Register Description.....                                         | 800        |
| 24.4.1 Register List.....                                              | 800        |
| 24.4.2 General Count Value Register (TMRA_CNTER).....                  | 802        |
| 24.4.3 Period Reference Value Register (TMRA_PERAR).....               | 803        |
| 24.4.4 Compare Reference Value Register (TMRA_CMPARm) (m=1-4).....     | 804        |
| 24.4.5 Control Status Register L (TMRA_BCSTRL).....                    | 805        |
| 24.4.6 Control Status Register H (TMRA_BCSTRH).....                    | 807        |
| 24.4.7 Interrupt Control Register (TMRA_ICONR).....                    | 808        |
| 24.4.8 Event Control Register (TMRA_ECONR).....                        | 809        |
| 24.4.9 Filter Control Register (TMRA_FCONR).....                       | 810        |
| 24.4.10 Status Flag Register (TMRA_STFLR).....                         | 811        |
| 24.4.11 Buffer Control Register (TMRA_BCONRm) (m=1-2).....             | 813        |
| 24.4.12 Capture Control Register (TMRA_CCONRm) (m=1-4).....            | 815        |
| 24.4.13 Port Control Register (TMRA_PCONRm) (m=1-4).....               | 818        |
| 24.4.14 Hardware Trigger Event Selection Register (TMRA_HCONR).....    | 820        |
| 24.4.15 Hardware Count Up Event Selection Register (TMRA_HCUPR).....   | 823        |
| 24.4.16 Hardware Count Down Event Selection Register (TMRA_HCDOR)..... | 826        |
| <b>25 General Timer (Timer2).....</b>                                  | <b>829</b> |
| 25.1 Introduction.....                                                 | 829        |
| 25.2 Features.....                                                     | 829        |
| 25.3 Functional Description.....                                       | 829        |
| 25.3.1 Basic Block Diagram.....                                        | 829        |
| 25.3.2 Clock Source Selection.....                                     | 830        |
| 25.3.3 Compare Output.....                                             | 831        |
| 25.3.4 Hardware Triggering.....                                        | 832        |
| 25.3.5 Digital Filtering.....                                          | 834        |
| 25.3.6 Software Synchronization Start.....                             | 835        |
| 25.3.7 Interrupt and Events.....                                       | 835        |
| 25.4 Register Description.....                                         | 836        |
| 25.4.1 Register List.....                                              | 836        |
| 25.4.2 Count Value Register m (TMR2_CNTmR) (m=A/B).....                | 837        |
| 25.4.3 Reference Value Register m (TMR2_CMPmR) (m=A/B).....            | 837        |
| 25.4.4 Basic Control Register (TMR2_BCONR).....                        | 838        |
| 25.4.5 Interrupt Control Register (TMR2_ICONR).....                    | 841        |
| 25.4.6 Port Control Register (TMR2_PCONR).....                         | 842        |
| 25.4.7 Hardware Control Register (TMR2_HCONR).....                     | 845        |
| 25.4.8 Status Flag Register (TMR2_STFLR).....                          | 849        |
| 25.5 Precautions.....                                                  | 851        |
| <b>26 General Timer (Timer0).....</b>                                  | <b>852</b> |
| 26.1 Introduction.....                                                 | 852        |



|                                                                   |            |
|-------------------------------------------------------------------|------------|
| 26.2 Functional Description.....                                  | 852        |
| 26.2.1 Timer0 Block Diagram.....                                  | 852        |
| 26.2.2 Clock Source Selection.....                                | 852        |
| 26.2.3 Basic Counting.....                                        | 853        |
| 26.2.4 Hardware Triggering.....                                   | 853        |
| 26.2.5 Software Synchronization Start.....                        | 853        |
| 26.2.6 Interrupt and Event.....                                   | 854        |
| 26.3 Register Description.....                                    | 856        |
| 26.3.1 Register List.....                                         | 856        |
| 26.3.2 Count Value Register m (TMR0_CNTmR) (m=A/B).....           | 857        |
| 26.3.3 Reference Value Register m (TMR0_CMPmR) (m=A/B).....       | 857        |
| 26.3.4 Basic Control Register (TMR0_BCONR).....                   | 858        |
| 26.3.5 Status Flag Register (TMR0_STFLR).....                     | 862        |
| 26.3.6 Count Synchronization Enable Register 1 (TMR_SYNENR1)..... | 864        |
| 26.3.7 Count Synchronization Enable Register 2 (TMR_SYNENR2)..... | 867        |
| 26.4 Precautions.....                                             | 870        |
| <b>27 Real Time Clock (RTC).....</b>                              | <b>871</b> |
| 27.1 Introduction.....                                            | 871        |
| 27.2 Features.....                                                | 871        |
| 27.3 Functional Description.....                                  | 872        |
| 27.3.1 RTC Block Diagram.....                                     | 872        |
| 27.3.2 Power-on Setting.....                                      | 872        |
| 27.3.3 System Low Power Mode Switching.....                       | 872        |
| 27.3.4 Clock Error Compensation.....                              | 873        |
| 27.3.5 Intrusion Detection.....                                   | 873        |
| 27.3.6 Timestamp Function.....                                    | 873        |
| 27.3.7 Interrupt.....                                             | 873        |
| 27.4 Operation Example.....                                       | 874        |
| 27.4.1 RTC Count Start Setting.....                               | 874        |
| 27.4.2 Read Count Register.....                                   | 874        |
| 27.4.3 Write Count Register.....                                  | 874        |
| 27.4.4 Alarm Clock Setting.....                                   | 875        |
| 27.4.5 1Hz Output.....                                            | 875        |
| 27.5 Register Description.....                                    | 877        |
| 27.5.1 Register List.....                                         | 877        |
| 27.5.2 Control Register 0 (RTC_CR0).....                          | 878        |
| 27.5.3 Control Register 1 (RTC_CR1).....                          | 879        |
| 27.5.4 Control Register 2 (RTC_CR2).....                          | 880        |
| 27.5.5 Control Register 3 (RTC_CR3).....                          | 882        |
| 27.5.6 Second Count Register (RTC_SEC).....                       | 883        |
| 27.5.7 Minute Count Register (RTC_MIN).....                       | 883        |
| 27.5.8 Hour Count Register (RTC_HOUR).....                        | 884        |
| 27.5.9 Day Count Register (RTC_DAY).....                          | 885        |
| 27.5.10 Week Count Register (RTC_WEEK).....                       | 886        |
| 27.5.11 Month Count Register (RTC_MON).....                       | 887        |

|                                                                                |            |
|--------------------------------------------------------------------------------|------------|
| 27.5.12 Year Count Register (RTC_YEAR).....                                    | 888        |
| 27.5.13 Minute Alarm Register (RTC_ALMMIN).....                                | 888        |
| 27.5.14 Hour Alarm Register (RTC_ALMHOUR).....                                 | 889        |
| 27.5.15 Week Alarm Register (RTC_ALMWEEK).....                                 | 889        |
| 27.5.16 Clock Error Compensation Register (RTC_ERRCRH, RTC_ERRCRL).....        | 890        |
| 27.5.17 Intrusion Control Register0 (RTC_TPCR0).....                           | 892        |
| 27.5.18 Intrusion Control Register1 (RTC_TPCR1).....                           | 893        |
| 27.5.19 Intrusion Status Register (RTC_TPSR).....                              | 894        |
| 27.5.20 Second Timestamp Register (RTC_SECTP).....                             | 895        |
| 27.5.21 Minute Timestamp Register (RTC_MINTP).....                             | 895        |
| 27.5.22 Hour Timestamp Register (RTC_HOURTP).....                              | 896        |
| 27.5.23 Day Timestamp Register (RTC_DAYTP).....                                | 896        |
| 27.5.24 Month Timestamp Register (RTC_MONTP).....                              | 897        |
| <b>28 Watchdog Timer (WDT/SWDT).....</b>                                       | <b>898</b> |
| 28.1 Introduction.....                                                         | 898        |
| 28.2 Features.....                                                             | 898        |
| 28.3 Functional Description.....                                               | 898        |
| 28.3.1 Start the Watchdog.....                                                 | 898        |
| 28.3.2 Hardware Startup Mode.....                                              | 899        |
| 28.3.3 Software Startup Mode.....                                              | 899        |
| 28.3.4 Refresh Action.....                                                     | 900        |
| 28.3.5 Flag Bit.....                                                           | 901        |
| 28.3.6 Interrupt/Reset.....                                                    | 901        |
| 28.3.7 Count Underflow.....                                                    | 901        |
| 28.3.8 Refresh Error.....                                                      | 903        |
| 28.4 Register Description.....                                                 | 904        |
| 28.4.1 Register List.....                                                      | 904        |
| 28.4.2 Control Register (SWDT_CR, WDT_CR).....                                 | 905        |
| 28.4.3 Status Register (SWDT_SR, WDT_SR).....                                  | 906        |
| 28.4.4 Refresh Register (SWDT_RR, WDT_RR).....                                 | 907        |
| 28.5 Precautions.....                                                          | 907        |
| <b>29 Universal Synchronous Asynchronous Receiver/Transmitter (USART).....</b> | <b>908</b> |
| 29.1 Introduction.....                                                         | 908        |
| 29.2 Features.....                                                             | 908        |
| 29.3 Functional Description.....                                               | 910        |
| 29.3.1 System Block Diagram.....                                               | 910        |
| 29.3.2 UART Function.....                                                      | 911        |
| 29.3.3 Multiprocessor Communication.....                                       | 921        |
| 29.3.4 UART-LIN.....                                                           | 925        |
| 29.3.5 Smart Card.....                                                         | 927        |
| 29.3.6 Clock Synchronization Mode.....                                         | 930        |
| 29.3.7 Digital Filter Function.....                                            | 935        |
| 29.3.8 Interrupt.....                                                          | 937        |
| 29.4 Register Description.....                                                 | 938        |
| 29.4.1 Register List.....                                                      | 938        |

|                                                                    |             |
|--------------------------------------------------------------------|-------------|
| 29.4.2 Status Register (USART_SR).....                             | 939         |
| 29.4.3 Transmit Data Register (USART_TDR).....                     | 944         |
| 29.4.4 Receive Data Register (USART_RDR).....                      | 944         |
| 29.4.5 Baud Rate Register (USART_BRR).....                         | 945         |
| 29.4.6 Control Register 1 (USART_CR1).....                         | 947         |
| 29.4.7 Control Register 2 (USART_CR2).....                         | 952         |
| 29.4.8 Control Register 3 (USART_CR3).....                         | 956         |
| 29.4.9 Prescaler Register (USART_PR).....                          | 959         |
| 29.4.10 LIN Baud Rate Measurement Count Register (USART_LBMC)..... | 960         |
| 29.4.11 USART1 Filter Control Register (USART1_NFC).....           | 961         |
| 29.5 Precautions.....                                              | 962         |
| 29.5.1 UART Precautions.....                                       | 962         |
| 29.5.2 Clock Synchronous Mode Precautions.....                     | 962         |
| 29.5.3 Other Precautions.....                                      | 962         |
| <b>30 Inter-Integrated Circuit (I2C).....</b>                      | <b>963</b>  |
| 30.1 Introduction.....                                             | 963         |
| 30.2 Features.....                                                 | 963         |
| 30.3 Functional Description.....                                   | 964         |
| 30.3.1 I2C System Block Diagram.....                               | 964         |
| 30.3.2 I2C Protocol.....                                           | 965         |
| 30.3.3 Address Matching.....                                       | 972         |
| 30.3.4 SMBus Action.....                                           | 975         |
| 30.3.5 Reset.....                                                  | 976         |
| 30.3.6 Programmable Digital Filtering.....                         | 977         |
| 30.3.7 Interrupt and Event.....                                    | 977         |
| 30.4 Operation Example.....                                        | 978         |
| 30.4.1 I2C Initialization Process.....                             | 978         |
| 30.5 Register Description.....                                     | 980         |
| 30.5.1 Register List.....                                          | 980         |
| 30.5.2 I2C Control Register 1 (I2C_CR1).....                       | 981         |
| 30.5.3 I2C Control Register 2 (I2C_CR2).....                       | 983         |
| 30.5.4 I2C Control Register 3 (I2C_CR3).....                       | 985         |
| 30.5.5 I2C Control Register 4 (I2C_CR4).....                       | 986         |
| 30.5.6 I2C Slave Address Register 0 (I2C_SLR0).....                | 987         |
| 30.5.7 I2C Slave Address Register 1 (I2C_SLR1).....                | 989         |
| 30.5.8 I2C SCL Level Timeout Control Register (I2C_SLTR).....      | 990         |
| 30.5.9 I2C Status Register (I2C_SR).....                           | 991         |
| 30.5.10 I2C Status Clear Register (I2C_CLR).....                   | 997         |
| 30.5.11 I2C Data Transmitting Register (I2C_DTR).....              | 999         |
| 30.5.12 I2C Data Receiving Register (I2C_DRR).....                 | 1000        |
| 30.5.13 I2C Clock Control Register (I2C_CCR).....                  | 1001        |
| 30.5.14 I2C Filter Control Register (I2C_FLTR).....                | 1002        |
| 30.5.15 I2C FIFO Status Register (I2C_FSTR).....                   | 1003        |
| 30.5.16 I2C Slave Address Receive Register (I2C_SLVADRR).....      | 1004        |
| <b>31 Serial Peripheral Interface (SPI).....</b>                   | <b>1005</b> |

|                                                                        |             |
|------------------------------------------------------------------------|-------------|
| 31.1 Introduction.....                                                 | 1005        |
| 31.2 Features.....                                                     | 1005        |
| 31.3 Functional Description.....                                       | 1007        |
| 31.3.1 System Block Diagram.....                                       | 1007        |
| 31.3.2 SPI Action System Description.....                              | 1008        |
| 31.3.3 Data Communication Description.....                             | 1010        |
| 31.3.4 Operating Instructions.....                                     | 1017        |
| 31.3.5 Self-diagnosis of Parity.....                                   | 1022        |
| 31.3.6 Error Detection.....                                            | 1023        |
| 31.3.7 SPI Initialization.....                                         | 1027        |
| 31.3.8 Interrupt and Events.....                                       | 1028        |
| 31.4 Register Description.....                                         | 1029        |
| 31.4.1 Register List.....                                              | 1029        |
| 31.4.2 SPI Data Register (SPI_DR).....                                 | 1030        |
| 31.4.3 SPI Control Register (SPI_CR).....                              | 1031        |
| 31.4.4 SPI Communication Configuration Register 1 (SPI_CFG1).....      | 1033        |
| 31.4.5 SPI Status Register (SPI_SR).....                               | 1035        |
| 31.4.6 SPI Communication Configuration Register 2 (SPI_CFG2).....      | 1037        |
| <b>32 Quad Serial Peripheral Interface (QSPI).....</b>                 | <b>1039</b> |
| 32.1 Introduction.....                                                 | 1039        |
| 32.2 Features.....                                                     | 1039        |
| 32.3 Memory Map.....                                                   | 1040        |
| 32.3.1 Internal Bus Space.....                                         | 1040        |
| 32.3.2 ROM Space and Bus Address Width.....                            | 1040        |
| 32.4 Functional Description.....                                       | 1042        |
| 32.4.1 Basic Block Diagram.....                                        | 1042        |
| 32.4.2 QSPI Bus.....                                                   | 1042        |
| 32.4.3 Timing Adjustment of QSPI Bus.....                              | 1044        |
| 32.4.4 Introduction to SPI Instructions for ROM Access.....            | 1047        |
| 32.4.5 Scheduling of QSPI Bus Period.....                              | 1055        |
| 32.4.6 XIP Control.....                                                | 1057        |
| 32.4.7 Pin Status of QSI02 and QSI03.....                              | 1059        |
| 32.4.8 Direct Communication Mode.....                                  | 1059        |
| 32.4.9 Interrupt.....                                                  | 1060        |
| 32.5 Register Description.....                                         | 1061        |
| 32.5.1 Register List.....                                              | 1061        |
| 32.5.2 QSPI Control Register (QSPI_CR).....                            | 1062        |
| 32.5.3 QSPI Chip Select Control Register (QSPI_CSCR).....              | 1064        |
| 32.5.4 QSPI Format Control Register (QSPI_FCR).....                    | 1065        |
| 32.5.5 QSPI Status Register (QSPI_SR).....                             | 1067        |
| 32.5.6 QSPI Direct Communication Instruction Register (QSPI_DCOM)..... | 1068        |
| 32.5.7 QSPI Instruction Code Register (QSPI_CCMD).....                 | 1068        |
| 32.5.8 QSPI XIP Mode Code Register (QSPI_XCMD).....                    | 1069        |
| 32.5.9 QSPI Flag Clear Register (QSPI_CLR).....                        | 1069        |
| 32.5.10 QSPI External Extended Address Register (QSPI_EXAR).....       | 1070        |

|                                                                |             |
|----------------------------------------------------------------|-------------|
| 32.6 Precautions.....                                          | 1071        |
| 32.6.1 QSPI Registers Setting.....                             | 1071        |
| 32.6.2 Module Stop Bit Settings.....                           | 1071        |
| <b>33 Inter_IC Sound Bus (I2S).....</b>                        | <b>1072</b> |
| 33.1 Introduction.....                                         | 1072        |
| 33.2 Features.....                                             | 1072        |
| 33.3 Functional Description.....                               | 1073        |
| 33.3.1 System Block Diagram.....                               | 1073        |
| 33.3.2 Pin Description.....                                    | 1073        |
| 33.3.3 Communication Method.....                               | 1074        |
| 33.3.4 Supported Audio Protocols.....                          | 1074        |
| 33.3.5 Clock Generator.....                                    | 1079        |
| 33.3.6 Master/Slave Mode.....                                  | 1080        |
| 33.3.7 Interrupt.....                                          | 1082        |
| 33.4 Register Description.....                                 | 1084        |
| 33.4.1 Register List.....                                      | 1084        |
| 33.4.2 I2S Control Register (I2S_CTRL).....                    | 1085        |
| 33.4.3 I2S Status Register (I2S_SR).....                       | 1087        |
| 33.4.4 I2S Error Status Register (I2S_ER).....                 | 1088        |
| 33.4.5 I2S Configuration Register (I2S_CFGR).....              | 1089        |
| 33.4.6 I2S Transmit Buffer FIFO Data Register (I2S_TXBUF)..... | 1090        |
| 33.4.7 I2S Receive Buffer FIFO Data Register (I2S_RXBUF).....  | 1091        |
| 33.4.8 I2S Prescaler Register (I2S_PR).....                    | 1091        |
| 33.5 Precautions.....                                          | 1092        |
| 33.5.1 Precautions when Using as a Master.....                 | 1092        |
| 33.5.2 Precautions when Using as a Slave.....                  | 1094        |
| 33.5.3 Anti-interference Support Function.....                 | 1094        |
| 33.5.4 Data Channel Identification Function.....               | 1094        |
| 33.5.5 Four-way Synchronous Control Function.....              | 1094        |
| 33.5.6 Other Precautions.....                                  | 1095        |
| <b>34 USB2.0 High-Speed Module (USBHS).....</b>                | <b>1096</b> |
| 34.1 Introduction.....                                         | 1096        |
| 34.2 Features.....                                             | 1096        |
| 34.2.1 General Features.....                                   | 1096        |
| 34.2.2 Host Mode Features.....                                 | 1097        |
| 34.2.3 Device Mode Features.....                               | 1097        |
| 34.3 Functional Description.....                               | 1098        |
| 34.3.1 System Block Diagram.....                               | 1098        |
| 34.3.2 Pin Description.....                                    | 1098        |
| 34.3.3 Clock and Work Mode.....                                | 1099        |
| 34.3.4 Mode Decision.....                                      | 1100        |
| 34.3.5 Host Function.....                                      | 1100        |
| 34.3.6 Device Function.....                                    | 1102        |
| 34.3.7 SOF Pulse Pin Output Function.....                      | 1105        |
| 34.3.8 Power Consumption Control.....                          | 1106        |

|                                                   |             |
|---------------------------------------------------|-------------|
| 34.3.9 Dynamic Update of USBHS_HFIR Register..... | 1106        |
| 34.3.10 Data FIFO.....                            | 1107        |
| 34.3.11 Host FIFO Architecture.....               | 1107        |
| 34.3.12 Device FIFO Architecture.....             | 1109        |
| 34.3.13 FIFO RAM Allocation.....                  | 1110        |
| 34.3.14 System Performance.....                   | 1111        |
| 34.3.15 Interrupt and Event.....                  | 1111        |
| 34.4 Programming Module.....                      | 1113        |
| 34.4.1 Module Initialization.....                 | 1113        |
| 34.4.2 Host Initialization.....                   | 1113        |
| 34.4.3 Device Initialization.....                 | 1114        |
| 34.4.4 DMA Mode.....                              | 1114        |
| 34.4.5 Host Programming Model.....                | 1114        |
| 34.4.6 Device Programming Model.....              | 1116        |
| 34.5 Operational Module.....                      | 1119        |
| 34.5.1 SETUP and OUT Data Transmission.....       | 1119        |
| 34.5.2 IN Data Transmission.....                  | 1125        |
| 34.6 Register Description.....                    | 1133        |
| 34.6.1 Register List.....                         | 1134        |
| 34.6.2 USB System Control Register.....           | 1137        |
| 34.6.3 USBHS Global Register.....                 | 1139        |
| 34.6.4 USBHS Host Mode Register.....              | 1171        |
| 34.6.5 USBHS Device Mode Register.....            | 1189        |
| 34.6.6 USBHS Clock Gating Control Register.....   | 1227        |
| <b>35 USB2.0 Full-Speed Module (USBFS).....</b>   | <b>1228</b> |
| 35.1 Introduction.....                            | 1228        |
| 35.2 Features.....                                | 1228        |
| 35.2.1 General Features.....                      | 1228        |
| 35.2.2 Host Mode Features.....                    | 1228        |
| 35.2.3 Device Mode Features.....                  | 1229        |
| 35.3 Functional Description.....                  | 1230        |
| 35.3.1 System Block Diagram.....                  | 1230        |
| 35.3.2 Pin Description.....                       | 1230        |
| 35.3.3 Clock and Work Mode.....                   | 1231        |
| 35.3.4 Mode Decision.....                         | 1231        |
| 35.3.5 Host Function.....                         | 1231        |
| 35.3.6 Device Function.....                       | 1234        |
| 35.3.7 SOF Pulse Pin Output Function.....         | 1237        |
| 35.3.8 Power Consumption Control.....             | 1238        |
| 35.3.9 Dynamic Update of USBFS_HFIR Register..... | 1238        |
| 35.3.10 Data FIFO.....                            | 1239        |
| 35.3.11 Host FIFO Architecture.....               | 1239        |
| 35.3.12 Device FIFO Architecture.....             | 1241        |
| 35.3.13 FIFO RAM Allocation.....                  | 1242        |
| 35.3.14 System Performance.....                   | 1243        |

|                                                                  |             |
|------------------------------------------------------------------|-------------|
| 35.3.15 Interrupt and Event.....                                 | 1243        |
| 35.4 Programming Module.....                                     | 1244        |
| 35.4.1 Module Initialization.....                                | 1244        |
| 35.4.2 Host Initialization.....                                  | 1245        |
| 35.4.3 Device Initialization.....                                | 1245        |
| 35.4.4 DMA Mode.....                                             | 1246        |
| 35.4.5 Host Programming Model.....                               | 1246        |
| 35.4.6 Device Programming Model.....                             | 1247        |
| 35.5 Operational Module.....                                     | 1250        |
| 35.5.1 SETUP and OUT Data Transmission.....                      | 1250        |
| 35.5.2 IN Data Transmission.....                                 | 1256        |
| 35.6 Register Description.....                                   | 1264        |
| 35.6.1 Register List.....                                        | 1265        |
| 35.6.2 USB System Control Register.....                          | 1268        |
| 35.6.3 USBFS Global Register.....                                | 1270        |
| 35.6.4 USBFS Host Mode Register.....                             | 1300        |
| 35.6.5 USBFS Device Mode Register.....                           | 1316        |
| 35.6.6 USBFS Clock Gating Control Register.....                  | 1347        |
| <b>36 CAN FD Controller (CAN FD).....</b>                        | <b>1348</b> |
| 36.1 Introduction.....                                           | 1348        |
| 36.2 Features.....                                               | 1348        |
| 36.3 Functional Description.....                                 | 1349        |
| 36.3.1 System Block Diagram.....                                 | 1349        |
| 36.3.2 Pin Description.....                                      | 1349        |
| 36.3.3 Operating Mode.....                                       | 1349        |
| 36.3.4 Baud Rate Setting.....                                    | 1351        |
| 36.3.5 Transmission Processing.....                              | 1353        |
| 36.3.6 Reception Processing.....                                 | 1355        |
| 36.3.7 Error Handling.....                                       | 1356        |
| 36.3.8 BUS OFF.....                                              | 1356        |
| 36.3.9 Arbitration lost Position Capture.....                    | 1357        |
| 36.3.10 Time-Triggered TTCAN.....                                | 1357        |
| 36.3.11 TDC and RDC.....                                         | 1359        |
| 36.3.12 Upward Compatibility with CAN-FD Function.....           | 1359        |
| 36.3.13 RAM ECC Function.....                                    | 1360        |
| 36.3.14 Software Reset Function.....                             | 1360        |
| 36.3.15 Interrupt.....                                           | 1362        |
| 36.4 Register Description.....                                   | 1363        |
| 36.4.1 Register List.....                                        | 1363        |
| 36.4.2 CAN Receive BUF Register (CAN_RBUF).....                  | 1365        |
| 36.4.3 CAN Transmit BUF Register (CAN_TBUF).....                 | 1368        |
| 36.4.4 CAN Configuration and Status Register (CAN_CFG_STAT)..... | 1370        |
| 36.4.5 CAN Command Register (CAN_TCMD).....                      | 1372        |
| 36.4.6 CAN Transmit Control Register (CAN_TCTRL).....            | 1374        |
| 36.4.7 CAN Receive Control Register (CAN_RCTRL).....             | 1376        |

|                                                                             |             |
|-----------------------------------------------------------------------------|-------------|
| 36.4.8 CAN Receive and Transmit Interrupt Enable Register (CAN_RTIE).....   | 1377        |
| 36.4.9 CAN Receive and Transmit Interrupt Flag Register (CAN_RTIF).....     | 1378        |
| 36.4.10 CAN Error Interrupt Enable and Flag Register (CAN_ERRINT).....      | 1380        |
| 36.4.11 CAN Slow Bit Timing Register (CAN_SBT).....                         | 1381        |
| 36.4.12 CAN Fast Bit Timing Register (CAN_FBT).....                         | 1382        |
| 36.4.13 CAN Transmitter Delay Compensation Register (CAN_TDC).....          | 1383        |
| 36.4.14 CAN Error and Arbitration Lost Capture Register (CAN_EALCAP).....   | 1384        |
| 36.4.15 CAN Warning Limits Register (CAN_LIMIT).....                        | 1385        |
| 36.4.16 CAN Receive Error Counter Register (CAN_RECNT).....                 | 1385        |
| 36.4.17 CAN Transmit Error Counter Register (CAN_TECNT).....                | 1386        |
| 36.4.18 CAN Filter Group Control Register (CAN_ACFCTRL).....                | 1387        |
| 36.4.19 CAN Filter Group Enable Register (CAN_ACFEN).....                   | 1388        |
| 36.4.20 CAN Filter Group Code and Mask Register (CAN_ACF).....              | 1390        |
| 36.4.21 TTCAN TB Slot Pointer Register (CAN_TB SLOT).....                   | 1391        |
| 36.4.22 TTCAN Time Trigger Configuration Register (CAN_TTCFG).....          | 1392        |
| 36.4.23 TTCAN Reference Message Register (CAN_REF_MSG).....                 | 1394        |
| 36.4.24 TTCAN Trigger Configuration Register (CAN_TRG_CFG).....             | 1395        |
| 36.4.25 TTCAN Trigger Time Register (CAN_TT_TRIG).....                      | 1396        |
| 36.4.26 TTCAN Trigger Watchdog Time Register (CAN_TT_WTRIG).....            | 1396        |
| 36.4.27 CAN RAM Check Control Register (CAN_RAM_CKCR).....                  | 1397        |
| 36.4.28 CAN RAM Check Status Register (CAN_RAM_CKSR).....                   | 1398        |
| 36.4.29 CAN RAM Error Injection Enable Register (CAN_RAM_EIEN).....         | 1398        |
| 36.4.30 CAN RAM Error Injection Bit Enable Register 0 (CAN_RAM_EIBIT0)..... | 1399        |
| 36.4.31 CAN RAM Error Injection Bit Enable Register 1 (CAN_RAM_EIBIT1)..... | 1399        |
| 36.4.32 CAN RAM ECC Error Address Register (CAN_RAM_ECCERRADDR).....        | 1400        |
| 36.4.33 CAN System Control Register (CAN_SYCTLREG).....                     | 1400        |
| 36.5 Precautions.....                                                       | 1401        |
| 36.5.1 CAN Bus Anti-Interference Measures.....                              | 1401        |
| 36.5.2 CAN Controller Noise Constraints.....                                | 1401        |
| <b>37 Controller Area Network (MCAN).....</b>                               | <b>1402</b> |
| 37.1 Introduction.....                                                      | 1402        |
| 37.2 Features.....                                                          | 1402        |
| 37.3 Functional Description.....                                            | 1403        |
| 37.3.1 MCAN System Block Diagram.....                                       | 1403        |
| 37.3.2 Pin Description.....                                                 | 1404        |
| 37.3.3 Working Mode.....                                                    | 1404        |
| 37.3.4 Timestamp Generation.....                                            | 1412        |
| 37.3.5 Timeout Counter.....                                                 | 1413        |
| 37.3.6 Reception Processing.....                                            | 1413        |
| 37.3.7 Transmission Processing.....                                         | 1420        |
| 37.3.8 FIFO Confirmation Processing.....                                    | 1424        |
| 37.3.9 Message RAM.....                                                     | 1425        |
| 37.4 Register Description.....                                              | 1437        |
| 37.4.1 Register List.....                                                   | 1437        |
| 37.4.2 MCAN Byte Order Register (MCAN_ENDN).....                            | 1440        |



|                                                                                             |      |
|---------------------------------------------------------------------------------------------|------|
| 37.4.3 MCAN Data Bit Time and Prescaler Register (MCAN_DBTP).....                           | 1441 |
| 37.4.4 MCAN Test Register (MCAN_TEST).....                                                  | 1443 |
| 37.4.5 MCAN RAM Watchdog Register (MCAN_RWD).....                                           | 1445 |
| 37.4.6 MCAN CC Control Register (MCAN_CCCR).....                                            | 1446 |
| 37.4.7 MCAN Nominal Bit Time and Prescaler Register (MCAN_NBTP).....                        | 1449 |
| 37.4.8 MCAN Timestamp Counter Configuration Register (MCAN_TSCC).....                       | 1450 |
| 37.4.9 MCAN Timestamp Counter Value Register (MCAN_TSCV).....                               | 1451 |
| 37.4.10 MCAN Timeout Counter Configuration Register (MCAN_TOCC).....                        | 1452 |
| 37.4.11 MCAN Timeout Counter Value Register (MCAN_TOCV).....                                | 1453 |
| 37.4.12 MCAN Error Counter Register (MCAN_ECR).....                                         | 1454 |
| 37.4.13 MCAN Protocol Status Register (MCAN_PSR).....                                       | 1455 |
| 37.4.14 MCAN Transmit Delay Compensation Register (MCAN_TDCR).....                          | 1459 |
| 37.4.15 MCAN Interrupt Register (MCAN_IR).....                                              | 1460 |
| 37.4.16 MCAN Interrupt Enable Register (MCAN_IE).....                                       | 1464 |
| 37.4.17 MCAN Interrupt Line Selection Register (MCAN_ILS).....                              | 1467 |
| 37.4.18 MCAN Interrupt Line Enable Register (MCAN_ILE).....                                 | 1470 |
| 37.4.19 MCAN Global Filter Configuration Register (MCAN_GFC).....                           | 1471 |
| 37.4.20 MCAN Standard ID Filter Configuration Register (MCAN_SIDFC).....                    | 1472 |
| 37.4.21 MCAN Extended ID Filter Configuration Register (MCAN_XIDFC).....                    | 1473 |
| 37.4.22 MCAN Extended ID and Mask Register (MCAN_XIDAM).....                                | 1474 |
| 37.4.23 MCAN High Priority Message Status Register (MCAN_HPMS).....                         | 1475 |
| 37.4.24 MCAN New Data 1 Register (MCAN_NDAT1).....                                          | 1476 |
| 37.4.25 MCAN New Data 2 Register (MCAN_NDAT2).....                                          | 1477 |
| 37.4.26 MCAN Receive FIFO0 Configuration Register (MCAN_RXF0C).....                         | 1478 |
| 37.4.27 MCAN Receive FIFO0 Status Register (MCAN_RXF0S).....                                | 1479 |
| 37.4.28 MCAN Receive FIFO0 Acknowledge Register (MCAN_RXF0A).....                           | 1480 |
| 37.4.29 MCAN Receive Buffer Configuration Register (MCAN_RXBC).....                         | 1480 |
| 37.4.30 MCAN Receive FIFO1 Configuration Register (MCAN_RXF1C).....                         | 1481 |
| 37.4.31 MCAN Receive FIFO1 Status Register (MCAN_RXF1S).....                                | 1482 |
| 37.4.32 MCAN Receive FIFO1 Acknowledge Register (MCAN_RXF1A).....                           | 1483 |
| 37.4.33 MCAN Receive Buffer and FIFO Element Size Configuration Registers (MCAN_RXESC)..... | 1484 |
| 37.4.34 MCAN Transmit Buffer Configuration Register (MCAN_TXBC).....                        | 1485 |
| 37.4.35 MCAN Transmit FIFO/Queue Status Register (MCAN_TXFQS).....                          | 1486 |
| 37.4.36 MCAN Transmit Buffer Element Size Configuration Register (MCAN_TXESC).....          | 1487 |
| 37.4.37 MCAN Transmit Buffer Request Pending Register (MCAN_TXBRP).....                     | 1488 |
| 37.4.38 MCAN Transmit Buffer Add Request Register (MCAN_TXBAR).....                         | 1489 |
| 37.4.39 MCAN Transmit Buffer Cancel Request Register (MCAN_TXBCR).....                      | 1490 |
| 37.4.40 MCAN Transmit Buffer Transmit Occurred Register (MCAN_TXBTO).....                   | 1491 |
| 37.4.41 MCAN Transmit Buffer Cancel Completed Register (MCAN_TXBCF).....                    | 1492 |
| 37.4.42 MCAN Transmit Buffer Transmit Interrupt Enable Register (MCAN_TXBTIE).....          | 1493 |
| 37.4.43 MCAN Transmit Buffer Cancel Completed Interrupt Enable Register (MCAN_TXBCIE).....  | 1493 |
| 37.4.44 MCAN Transmit Event FIFO Configuration Register (MCAN_TXEFC).....                   | 1494 |
| 37.4.45 MCAN Transmit Event FIFO Status Register (MCAN_TXEFS).....                          | 1495 |
| 37.4.46 MCAN Transmit Event FIFO Acknowledge Register (MCAN_TXEFA).....                     | 1496 |
| 37.4.47 MCAN Message RAM Check Control Register (MCANRAM_CKCR).....                         | 1497 |

|                                                                                      |             |
|--------------------------------------------------------------------------------------|-------------|
| 37.4.48 MCAN Message RAM Check Status Register (MCANRAM_CKSR).....                   | 1498        |
| 37.4.49 MCAN Message RAM Error Injection Enable Register (MCANRAM_EIEN).....         | 1498        |
| 37.4.50 MCAN Message RAM Error Injection Bit Enable Register 0 (MCANRAM_EIBIT0)..... | 1499        |
| 37.4.51 MCAN Message RAM Error Injection Bit Enable Register 1 (MCANRAM_EIBIT1)..... | 1499        |
| 37.4.52 MCAN Message RAM Check Error Address Register (MCANRAM_ECCERRADDR).....      | 1500        |
| 37.5 Precautions.....                                                                | 1500        |
| 37.5.1 CAN Bus Anti-Interference Measures.....                                       | 1500        |
| 37.5.2 CAN Controller Noise Constraints.....                                         | 1500        |
| <b>38 Ethernet MAC Controller (ETHMAC).....</b>                                      | <b>1501</b> |
| 38.1 Introduction.....                                                               | 1501        |
| 38.2 Features.....                                                                   | 1501        |
| 38.2.1 ETH_MAC Features.....                                                         | 1501        |
| 38.2.2 ETH_PTP Features.....                                                         | 1503        |
| 38.2.3 ETH_DMA Features.....                                                         | 1503        |
| 38.3 Functional Description.....                                                     | 1504        |
| 38.3.1 Basic Block Diagram.....                                                      | 1504        |
| 38.3.2 Interface Description.....                                                    | 1505        |
| 38.3.3 ETH_MAC Function.....                                                         | 1509        |
| 38.3.4 ETH_PTP Function.....                                                         | 1526        |
| 38.3.5 ETH_DMA Function.....                                                         | 1531        |
| 38.3.6 Interrupt.....                                                                | 1556        |
| 38.4 Register Description.....                                                       | 1558        |
| 38.4.1 Register List.....                                                            | 1558        |
| 38.4.2 ETH_MAC Register.....                                                         | 1562        |
| 38.4.3 ETH_PTP Register.....                                                         | 1598        |
| 38.4.4 ETH_DMA Register.....                                                         | 1610        |
| 38.4.5 ETH_MMC Register.....                                                         | 1631        |
| <b>39 SDIO Controller (SDIOC).....</b>                                               | <b>1649</b> |
| 39.1 Introduction.....                                                               | 1649        |
| 39.2 Features.....                                                                   | 1649        |
| 39.3 Functional Description.....                                                     | 1650        |
| 39.3.1 Port Assignment.....                                                          | 1650        |
| 39.3.2 Basic Access Method.....                                                      | 1650        |
| 39.3.3 Data Transmission.....                                                        | 1651        |
| 39.3.4 SD Clock.....                                                                 | 1651        |
| 39.3.5 Interrupts and DMA Start Requests.....                                        | 1651        |
| 39.4 Operation Example.....                                                          | 1652        |
| 39.4.1 Host and Device Initialization.....                                           | 1652        |
| 39.4.2 SD/MMC Single Block Read and Write.....                                       | 1653        |
| 39.4.3 SD/MMC Multi Block Read and Write.....                                        | 1654        |
| 39.4.4 Transfer Abort, Suspend and Resume.....                                       | 1655        |
| 39.4.5 Read Wait.....                                                                | 1656        |
| 39.4.6 Wake-up.....                                                                  | 1656        |
| 39.5 Register Description.....                                                       | 1658        |
| 39.5.1 Register List.....                                                            | 1658        |

|                                                                            |             |
|----------------------------------------------------------------------------|-------------|
| 39.5.2 Data Block Size Register (SDIOC_BLKSIZE).....                       | 1660        |
| 39.5.3 Data Block Count Register (SDIOC_BLKCNT).....                       | 1660        |
| 39.5.4 Argument Register 0 (SDIOC_ARG0).....                               | 1660        |
| 39.5.5 Argument Register 1 (SDIOC_ARG1).....                               | 1661        |
| 39.5.6 Transmission Mode Register (SDIOC_TRANSMODE).....                   | 1661        |
| 39.5.7 Command Register (SDIOC_CMD).....                                   | 1662        |
| 39.5.8 Response Register 0 (SDIOC_RESP0).....                              | 1663        |
| 39.5.9 Response Register 1 (SDIOC_RESP1).....                              | 1664        |
| 39.5.10 Response Register 2 (SDIOC_RESP2).....                             | 1664        |
| 39.5.11 Response Register 3 (SDIOC_RESP3).....                             | 1664        |
| 39.5.12 Response Register 4 (SDIOC_RESP4).....                             | 1665        |
| 39.5.13 Response Register 5 (SDIOC_RESP5).....                             | 1665        |
| 39.5.14 Response Register 6 (SDIOC_RESP6).....                             | 1665        |
| 39.5.15 Response Register 7 (SDIOC_RESP7).....                             | 1666        |
| 39.5.16 Data Buffer Register 0 (SDIOC_BUF0).....                           | 1666        |
| 39.5.17 Data Buffer Register 1 (SDIOC_BUF1).....                           | 1666        |
| 39.5.18 Host Status Register (SDIOC_PSTAT).....                            | 1667        |
| 39.5.19 Host Control Register (SDIOC_HOSTCON).....                         | 1669        |
| 39.5.20 Power Control Register (SDIOC_PWRCON).....                         | 1670        |
| 39.5.21 Data Block Gap Control Register (SDIOC_BLKGPCON).....              | 1671        |
| 39.5.22 Clock Control Register (SDIOC_CLKCON).....                         | 1672        |
| 39.5.23 Timeout Control Register (SDIOC_TOUTCON).....                      | 1673        |
| 39.5.24 Software Reset Register (SDIOC_SFTRST).....                        | 1674        |
| 39.5.25 Normal Interrupt Status Register (SDIOC_NORINTST).....             | 1675        |
| 39.5.26 Error Interrupt Status Register (SDIOC_ERRINTST).....              | 1676        |
| 39.5.27 Normal Interrupt State Enable Register (SDIOC_NORINTSGEN).....     | 1677        |
| 39.5.28 Error Interrupt Status Enable Register (SDIOC_ERRINTSTEN).....     | 1678        |
| 39.5.29 Normal Interrupt Signal Enable Register (SDIOC_NORINTSGEN).....    | 1679        |
| 39.5.30 Error Interrupt Signal Enable Register (SDIOC_ERRINTSGEN).....     | 1680        |
| 39.5.31 Auto-Command Error Status Register (SDIOC_ATCERRST).....           | 1681        |
| 39.5.32 Forced Auto-Command Error Status Control Register (SDIOC_FEA)..... | 1682        |
| 39.5.33 Forced Error Status Control Register (SDIOC_FEE).....              | 1683        |
| 39.5.34 System Control Register (SDIOC_SYCTLREG).....                      | 1685        |
| <b>40 External Memory Controller (EXMC).....</b>                           | <b>1686</b> |
| 40.1 Introduction.....                                                     | 1686        |
| 40.2 Features.....                                                         | 1686        |
| 40.3 Functional Description.....                                           | 1687        |
| 40.3.1 Controller Architecture.....                                        | 1687        |
| 40.3.2 Basic Access Specification.....                                     | 1687        |
| 40.3.3 Address Mapping.....                                                | 1689        |
| 40.3.4 Protocol Interface.....                                             | 1691        |
| 40.3.5 SMC-SRAM/PSRAM/NOR Flash Controller.....                            | 1694        |
| 40.3.6 DMC-SDRAM Controller.....                                           | 1715        |
| 40.3.7 NFC-NAND Flash Controller.....                                      | 1724        |
| 40.3.8 Interrupt Description.....                                          | 1726        |

|                                                                 |             |
|-----------------------------------------------------------------|-------------|
| 40.4 Register Description.....                                  | 1727        |
| 40.4.1 Register List.....                                       | 1727        |
| 40.4.2 EXMC Common Registers.....                               | 1729        |
| 40.4.3 SMC-SRAM/PSRAM/NOR Flash Register.....                   | 1730        |
| 40.4.4 DMC-SDRAM Registers.....                                 | 1741        |
| 40.4.5 NFC-NAND Flash Register.....                             | 1757        |
| 40.5 Precautions.....                                           | 1775        |
| <b>41 Digital Video Port (DVP).....</b>                         | <b>1776</b> |
| 41.1 Introduction.....                                          | 1776        |
| 41.2 Features.....                                              | 1776        |
| 41.3 Functional Description.....                                | 1777        |
| 41.3.1 System Block Diagram.....                                | 1777        |
| 41.3.2 Video Data Format.....                                   | 1778        |
| 41.3.3 Parallel Port Storage Format.....                        | 1778        |
| 41.3.4 Mode Selection.....                                      | 1780        |
| 41.3.5 Synchronization Control.....                             | 1781        |
| 41.3.6 Window Cropping.....                                     | 1782        |
| 41.3.7 FIFO Control.....                                        | 1784        |
| 41.3.8 DMA Control.....                                         | 1784        |
| 41.3.9 Interrupt and Events.....                                | 1784        |
| 41.4 Register Description.....                                  | 1786        |
| 41.4.1 Register List.....                                       | 1786        |
| 41.4.2 Control Register (DVP_CTR).....                          | 1787        |
| 41.4.3 Data Register (DVP_DTR).....                             | 1788        |
| 41.4.4 Status Register (DVP_STR).....                           | 1789        |
| 41.4.5 Interrupt Register (DVP_IER).....                        | 1790        |
| 41.4.6 DMA Data Transfer Register (DVP_DMR).....                | 1790        |
| 41.4.7 Software Synchronization Data Register (DVP_SSYNDR)..... | 1791        |
| 41.4.8 Software Synchronization Mask Register (DVP_SSYNMR)..... | 1791        |
| 41.4.9 Window Cropping Offset Register (DVP_CPSFTR).....        | 1792        |
| 41.4.10 Window Cropping Size Register (DVP_CPSZER).....         | 1793        |
| <b>42 Cryptographic Coprocessing Module (CPM).....</b>          | <b>1794</b> |
| 42.1 Introduction.....                                          | 1794        |
| 42.2 Features.....                                              | 1794        |
| 42.2.1 PKE Features.....                                        | 1794        |
| 42.2.2 SKE Features.....                                        | 1794        |
| 42.3 Functional Description.....                                | 1795        |
| 42.3.1 Secure Hash Algorithm (HASH).....                        | 1795        |
| 42.3.2 True Random Number Generator (TRNG).....                 | 1801        |
| 42.3.3 Public Key Engine (PKE).....                             | 1802        |
| 42.3.4 Symmetric Encryption Algorithm Module (SKE).....         | 1806        |
| 42.4 Register Description.....                                  | 1811        |
| 42.4.1 Register List.....                                       | 1811        |
| 42.4.2 HASH Register.....                                       | 1814        |
| 42.4.3 TRNG Register.....                                       | 1818        |

|                                                                          |             |
|--------------------------------------------------------------------------|-------------|
| 42.4.4 PKE Register.....                                                 | 1821        |
| 42.4.5 SKE Register.....                                                 | 1839        |
| <b>43 CRC Operation (CRC).....</b>                                       | <b>1850</b> |
| 43.1 Introduction.....                                                   | 1850        |
| 43.2 Functional Description.....                                         | 1850        |
| 43.2.1 Function Block Diagram.....                                       | 1850        |
| 43.2.2 CRC Parameter Model.....                                          | 1850        |
| 43.2.3 CRC16 Encoding Mode.....                                          | 1851        |
| 43.2.4 CRC16 Check Mode.....                                             | 1851        |
| 43.2.5 CRC32 Encoding Mode.....                                          | 1852        |
| 43.2.6 CRC32 Check Mode.....                                             | 1852        |
| 43.2.7 CRC64 Encoding Mode.....                                          | 1852        |
| 43.2.8 CRC64 Check Mode.....                                             | 1853        |
| 43.3 Register Description.....                                           | 1854        |
| 43.3.1 Register List.....                                                | 1854        |
| 43.3.2 Control Register (CRC_CR).....                                    | 1855        |
| 43.3.3 Result Register 1 (CRC_RESULT1).....                              | 1856        |
| 43.3.4 Result Register 2 (CRC_RESULT2).....                              | 1856        |
| 43.3.5 Data Register (CRC_DAT).....                                      | 1857        |
| <b>44 Data Computing Unit (DCU).....</b>                                 | <b>1858</b> |
| 44.1 Introduction.....                                                   | 1858        |
| 44.2 Features.....                                                       | 1858        |
| 44.3 Functional Description.....                                         | 1858        |
| 44.3.1 Addition Mode.....                                                | 1858        |
| 44.3.2 Subtraction Mode.....                                             | 1859        |
| 44.3.3 Hardware Trigger Start Mode.....                                  | 1859        |
| 44.3.4 Comparative Mode.....                                             | 1860        |
| 44.3.5 Triangle Wave Output Mode.....                                    | 1860        |
| 44.3.6 Increment Sawtooth Wave Output Mode.....                          | 1862        |
| 44.3.7 Decrement Sawtooth Wave Output Mode.....                          | 1863        |
| 44.3.8 Interrupt and Event.....                                          | 1864        |
| 44.4 Register Description.....                                           | 1865        |
| 44.4.1 Register List.....                                                | 1865        |
| 44.4.2 DCU Control Register (DCU_CTL).....                               | 1866        |
| 44.4.3 DCU Flag Register (DCU_FLAG) (DCU1-DCU4).....                     | 1868        |
| 44.4.4 DCU Flag Register (DCU_FLAG) (DCU5-DCU8).....                     | 1870        |
| 44.4.5 DCU Data Register (DCU_DATAx) (x=0-2).....                        | 1871        |
| 44.4.6 DCU Flag Reset Register (DCU_FLAGCLR) (DCU1-DCU4).....            | 1875        |
| 44.4.7 DCU Flag Reset Register (DCU_FLAGCLR) (DCU5-DCU8).....            | 1876        |
| 44.4.8 DCU Interrupt and Event Register (DCU_INTEVTSEL) (DCU1-DCU4)..... | 1877        |
| 44.4.9 DCU Interrupt and Event Register (DCU_INTEVTSEL) (DCU5-DCU8)..... | 1880        |
| <b>45 Math Arithmetical Unit (MAU).....</b>                              | <b>1883</b> |
| 45.1 Introduction.....                                                   | 1883        |
| 45.2 Functional Description.....                                         | 1883        |
| 45.2.1 Square Root Operation.....                                        | 1883        |

|                                                                 |             |
|-----------------------------------------------------------------|-------------|
| 45.2.2 Sine Operation.....                                      | 1884        |
| 45.2.3 Interrupt and Event Output.....                          | 1884        |
| 45.3 Register Description.....                                  | 1885        |
| 45.3.1 Register List.....                                       | 1885        |
| 45.3.2 Control State Register (MAU_CSR).....                    | 1886        |
| 45.3.3 Data Input Register 0 (MAU_DTR0).....                    | 1887        |
| 45.3.4 Result Output Register 0 (MAU_RTR0).....                 | 1888        |
| 45.3.5 Data Input Register 1 (MAU_DTR1).....                    | 1888        |
| 45.3.6 Result Output Register 1 (MAU_RTR1).....                 | 1889        |
| <b>46 Filter Math Accelerator (FMAC).....</b>                   | <b>1890</b> |
| 46.1 Introduction.....                                          | 1890        |
| 46.2 Features.....                                              | 1890        |
| 46.3 Functional Description.....                                | 1890        |
| 46.3.1 Basic Block Diagram.....                                 | 1890        |
| 46.3.2 Operating Procedures.....                                | 1891        |
| 46.3.3 Module Enable.....                                       | 1891        |
| 46.3.4 Coefficient Normalization.....                           | 1891        |
| 46.3.5 Interrupt and Events.....                                | 1891        |
| 46.4 Register Description.....                                  | 1892        |
| 46.4.1 Register List.....                                       | 1892        |
| 46.4.2 Module Enable Register (FMAC_ENR).....                   | 1893        |
| 46.4.3 Basic Control Register (FMAC_CTR).....                   | 1894        |
| 46.4.4 Interrupt Control Register (FMAC_IER).....               | 1895        |
| 46.4.5 Data Input Register (FMAC_DTR).....                      | 1895        |
| 46.4.6 Filter Coefficient Register n (FMAC_CORn) (n=0-16).....  | 1896        |
| 46.4.7 Result Output Register 0 (FMAC_RTR0).....                | 1896        |
| 46.4.8 Result Output Register 1 (FMAC_RTR1).....                | 1897        |
| 46.4.9 Operation Status Register (FMAC_STR).....                | 1897        |
| <b>47 Error Management Unit (ERMU).....</b>                     | <b>1898</b> |
| 47.1 Introduction.....                                          | 1898        |
| 47.2 Features.....                                              | 1898        |
| 47.3 Functional Description.....                                | 1899        |
| 47.3.1 Function Block Diagram.....                              | 1899        |
| 47.3.2 Error Output Channel.....                                | 1900        |
| 47.3.3 Wait Timer.....                                          | 1902        |
| 47.3.4 Reset Request.....                                       | 1902        |
| 47.3.5 Interrupt Request.....                                   | 1902        |
| 47.3.6 Pseudo Error Trigger.....                                | 1903        |
| 47.3.7 Interrupt.....                                           | 1903        |
| 47.3.8 Error Source.....                                        | 1903        |
| 47.4 Operation Example.....                                     | 1906        |
| 47.4.1 ERMU Error Management Example.....                       | 1906        |
| 47.5 Register Description.....                                  | 1907        |
| 47.5.1 Register List.....                                       | 1907        |
| 47.5.2 Error Output y Control Register (ERMU_EOyC) (y=0-3)..... | 1908        |

|                                                                                 |             |
|---------------------------------------------------------------------------------|-------------|
| 47.5.3 Error Output y Status Register (ERMU_EOyS) (y=0-3).....                  | 1909        |
| 47.5.4 Error Output y Clear Timer Count Register (ERMU_EOyCTCNT) (y=0-3).....   | 1909        |
| 47.5.5 Error Output y Clear Timer Compare Register (ERMU_EOyCTCMP) (y=0-3)..... | 1910        |
| 47.5.6 Error Output y Flip Timer Control Register (ERMU_EOyTTC) (y=0-3).....    | 1910        |
| 47.5.7 Error Output y Flip Timer Compare Register (ERMU_EOyTTCMP) (y=0-3).....  | 1911        |
| 47.5.8 Error Output y Mask 0 Register (ERMU_EOyOM0) (y=0-3).....                | 1911        |
| 47.5.9 Error Output y Mask 1 Register (ERMU_EOyOM1) (y=0-3).....                | 1912        |
| 47.5.10 Wait Timer z Control Register (ERMU_WTzC) (z= 0-3).....                 | 1912        |
| 47.5.11 Wait Timer z Status Register (ERMU_WTzS) (z= 0-3).....                  | 1913        |
| 47.5.12 Wait Timer z Count Register (ERMU_WTzCNT) (z= 0-3).....                 | 1913        |
| 47.5.13 Wait Timer z Compare Register (ERMU_WTzCMP) (z=0-3).....                | 1914        |
| 47.5.14 Wait Timer z Start Enable Register (ERMU_WTzSE) (z=0-3).....            | 1914        |
| 47.5.15 Error Source Status j Register (ERMU_ESSj) (j=0-1).....                 | 1915        |
| 47.5.16 Error Source Status Clear j Register (ERMU_ESSCj) (j=0-1).....          | 1915        |
| 47.5.17 Pseudo-error Source Trigger Register (ERMU_PETj) (j=0-1).....           | 1916        |
| 47.5.18 Reset Enable j Register (ERMU_REj) (j=0-1).....                         | 1916        |
| 47.5.19 Low Priority Interrupt Enable j Register (ERMU_LPIEj) (j=0-1).....      | 1917        |
| 47.5.20 High Priority Interrupt Enable j Register (ERMU_HPIEj) (j=0-1).....     | 1917        |
| 47.5.21 Count Clock Prescaler Register (ERMU_CCPS).....                         | 1918        |
| <b>48 Debug Controller (DBGC).....</b>                                          | <b>1919</b> |
| 48.1 Introduction.....                                                          | 1919        |
| 48.2 Functional Description.....                                                | 1919        |
| 48.2.1 Function Block Diagram.....                                              | 1919        |
| 48.2.2 Reference Documentation.....                                             | 1920        |
| 48.2.3 SWJ-DP Debug Port (SWD and JTAG).....                                    | 1920        |
| 48.2.4 Pin Arrangement and Debug Port Pins.....                                 | 1921        |
| 48.2.5 SW Debug Port-SW Protocol Overview.....                                  | 1922        |
| 48.2.6 TPIU (Trace Port Interface Unit).....                                    | 1923        |
| 48.3 Register Description.....                                                  | 1926        |
| 48.3.1 Register List.....                                                       | 1926        |
| 48.3.2 Chip Identification Register (CHIPID).....                               | 1927        |
| 48.3.3 DBG Status Register (MCUDBGSTAT).....                                    | 1927        |
| 48.3.4 Peripheral Debug Stop Control Register (MCUSTPCTL).....                  | 1928        |
| 48.3.5 Debug Component Trace Control Register (MCUTRACECTL).....                | 1931        |
| 48.3.6 Peripheral Debug Stop Control Register 2 (MCUSTPCTL2).....               | 1932        |
| <b>A Glossary.....</b>                                                          | <b>1934</b> |
| <b>Revision History .....</b>                                                   | <b>1935</b> |



## List of Figures

|             |                                                                                 |     |
|-------------|---------------------------------------------------------------------------------|-----|
| Figure 2-1  | QSPI Address Space Allocation.....                                              | 56  |
| Figure 3-1  | Bus Architecture Diagram.....                                                   | 62  |
| Figure 4-1  | Power-on Reset Timing Diagram.....                                              | 66  |
| Figure 4-2  | NRST Reset Timing Diagram.....                                                  | 67  |
| Figure 4-3  | Brown-out Reset Timing Diagram.....                                             | 67  |
| Figure 4-4  | Programmable Voltage Detection 1 Reset Timing Diagram.....                      | 68  |
| Figure 4-5  | Programmable Voltage Detection 2 Reset Timing Diagram.....                      | 68  |
| Figure 4-6  | Watchdog and Special Watchdog Reset Timing Diagram.....                         | 69  |
| Figure 4-7  | Power-off Wake-up Reset Timing Diagram.....                                     | 69  |
| Figure 4-8  | Software Reset Timing Diagram.....                                              | 69  |
| Figure 4-9  | MPU Error Reset Timing Diagram.....                                             | 70  |
| Figure 4-10 | RAM ECC Reset Timing Diagram.....                                               | 70  |
| Figure 4-11 | FLASH ECC Reset Timing Diagram.....                                             | 70  |
| Figure 4-12 | ERMU Reset Timing Diagram.....                                                  | 71  |
| Figure 4-13 | Clock Frequency Error Reset Timing Diagram.....                                 | 71  |
| Figure 4-14 | XTAL Error Reset Timing Diagram.....                                            | 71  |
| Figure 4-15 | Cortex-M4 Lockup Reset Timing Diagram.....                                      | 72  |
| Figure 4-16 | Module Function Reset.....                                                      | 72  |
| Figure 5-1  | Clock System Block Diagram.....                                                 | 90  |
| Figure 5-2  | Clock Frequency Measurement Block Diagram.....                                  | 91  |
| Figure 5-3  | External High Speed Oscillator Connection Example.....                          | 94  |
| Figure 5-4  | External Clock Input Connection Example.....                                    | 94  |
| Figure 5-5  | External High-speed Oscillator Fault Detection Example.....                     | 95  |
| Figure 5-6  | System Clock Selected as XTAL, Example of Detecting XTAL Oscillation Fault..... | 96  |
| Figure 5-7  | External Low-speed Oscillator Connection Example.....                           | 97  |
| Figure 5-8  | Clock Source Switching Flowchart.....                                           | 100 |
| Figure 5-9  | Clock Division Switching Flowchart.....                                         | 101 |
| Figure 5-10 | Clock Frequency Measurement Timing Diagram.....                                 | 102 |
| Figure 6-1  | Power Supply Configuration Diagram.....                                         | 152 |
| Figure 6-2  | Battery Backup Domain Power Switching Diagram.....                              | 153 |
| Figure 6-3  | Power-on Reset, Power-off Reset Waveforms.....                                  | 154 |
| Figure 6-4  | Brown-out Reset Waveform.....                                                   | 154 |
| Figure 6-5  | PVD1 Interrupt/Reset Block Diagram.....                                         | 155 |
| Figure 6-6  | PVD2 Interrupt/Reset Block Diagram.....                                         | 156 |
| Figure 6-7  | Power Monitor 1 Interrupt Timing Diagram.....                                   | 157 |
| Figure 6-8  | Power Monitor 1 Reset Timing Diagram.....                                       | 157 |
| Figure 6-9  | Power Monitor 2 Interrupt Timing Diagram.....                                   | 158 |
| Figure 6-10 | Power Monitor 2 Reset Timing Diagram.....                                       | 158 |
| Figure 6-11 | Internal Voltage Sampling Diagram.....                                          | 159 |
| Figure 6-12 | PTWKn Structure Diagram.....                                                    | 169 |
| Figure 8-1  | Flash Address Distribution.....                                                 | 221 |
| Figure 8-2  | Flash Address Distribution for 2MBytes Products.....                            | 230 |
| Figure 8-3  | Boot Swap Activation 1.....                                                     | 230 |
| Figure 8-4  | Boot Swap Activation 2.....                                                     | 230 |
| Figure 9-1  | Error Injection Diagram for SRAM 39-bit.....                                    | 270 |
| Figure 9-2  | Error Injection Diagram for CACHE RAM 137-bit.....                              | 271 |
| Figure 10-1 | Port Basic Structure Diagram.....                                               | 303 |
| Figure 11-1 | Interrupt System Block Diagram.....                                             | 347 |
| Figure 11-2 | Interrupt Request/Event Input Selection.....                                    | 349 |
| Figure 11-3 | Digital Filter Operation Diagram (taking NFSEL=0x02 as an example).....         | 350 |



|              |                                                                                              |     |
|--------------|----------------------------------------------------------------------------------------------|-----|
| Figure 12-1  | AOS Module Block Diagram.....                                                                | 371 |
| Figure 12-2  | PLU Block Diagram.....                                                                       | 374 |
| Figure 13-1  | Schematic Diagram of Partial Memory Access Permissions.....                                  | 400 |
| Figure 13-2  | Schematic Diagram of only Prohibiting Access to Part of Space.....                           | 400 |
| Figure 14-1  | KEYSCAN System Block Diagram.....                                                            | 437 |
| Figure 14-2  | Keyboard Scan Function Diagram.....                                                          | 438 |
| Figure 15-1  | CTC Function Block Diagram.....                                                              | 444 |
| Figure 15-2  | CTC Calibration Diagram.....                                                                 | 446 |
| Figure 16-1  | DMA Architecture Diagram.....                                                                | 455 |
| Figure 16-2  | Chain Transmission Diagram.....                                                              | 458 |
| Figure 16-3  | Schematic Diagram of NonSequence Address Transfer (Taking source address as an example)..... | 459 |
| Figure 16-4  | Schematic Diagram of NonSequence Reset.....                                                  | 461 |
| Figure 16-5  | Application Example 1: Memory-to-Memory Transfer.....                                        | 463 |
| Figure 16-6  | Application Example 2: Memory-to-Peripheral Transfer.....                                    | 464 |
| Figure 17-1  | CMP Function Block Diagram.....                                                              | 502 |
| Figure 17-2  | Voltage Comparison Diagram.....                                                              | 504 |
| Figure 17-3  | Window Compare Diagram.....                                                                  | 505 |
| Figure 17-4  | Blank Window Output Diagram.....                                                             | 506 |
| Figure 18-1  | ADC Block Diagram.....                                                                       | 519 |
| Figure 18-2  | Channel Mapping Diagram.....                                                                 | 521 |
| Figure 18-3  | Internal Analog Signal Selection.....                                                        | 522 |
| Figure 18-4  | Sequence A Single Scan Mode.....                                                             | 523 |
| Figure 18-5  | Continuous Scan.....                                                                         | 523 |
| Figure 18-6  | Data Buffer Mode.....                                                                        | 524 |
| Figure 18-7  | Dual Sequence Scan Mode (Sequence A Restarts from Interrupted Channel).....                  | 526 |
| Figure 18-8  | Dual Sequence Scan Mode (Sequence A Restarts from First Channel).....                        | 526 |
| Figure 18-9  | Analog Watchdog Protected Area (Comparative Mode).....                                       | 527 |
| Figure 18-10 | ADC Conversion Time Diagram.....                                                             | 528 |
| Figure 18-11 | Conversion Action When Averaging Function is Enabled.....                                    | 529 |
| Figure 18-12 | PGA and SH Channel Diagram for ADC1 and ADC2.....                                            | 531 |
| Figure 18-13 | Dedicated Sample Hold Circuit for Channels Enabled.....                                      | 532 |
| Figure 18-14 | Single Parallel Trigger Mode (Three ADCs).....                                               | 533 |
| Figure 18-15 | Single Delay Trigger Mode (Three ADCs).....                                                  | 534 |
| Figure 18-16 | Loop Parallel Trigger Mode (Three ADCs).....                                                 | 535 |
| Figure 18-17 | Loop Delay Trigger Mode (Two ADCs).....                                                      | 536 |
| Figure 18-18 | Loop Delay Trigger Mode (Three ADCs).....                                                    | 536 |
| Figure 18-19 | ADC Interrupt and Event Output Timing.....                                                   | 537 |
| Figure 19-1  | DAC Conversion Channel Block Diagram.....                                                    | 571 |
| Figure 19-2  | DAC Conversion Diagram.....                                                                  | 572 |
| Figure 19-3  | ADC Conversion Priority Mode Action Schematic Diagram.....                                   | 573 |
| Figure 20-1  | OTS Function Block Diagram.....                                                              | 582 |
| Figure 21-1  | Timer6 Basic Block Diagram.....                                                              | 592 |
| Figure 21-2  | Sawtooth Waveform (Counting Up).....                                                         | 593 |
| Figure 21-3  | Triangle Waveform.....                                                                       | 593 |
| Figure 21-4  | Compare Output Action.....                                                                   | 594 |
| Figure 21-5  | Capture Input Action.....                                                                    | 594 |
| Figure 21-6  | Hardware Refresh Action.....                                                                 | 596 |
| Figure 21-7  | Digital Filtering Function of Capture Input Ports.....                                       | 596 |
| Figure 21-8  | Software Synchronization Action.....                                                         | 598 |
| Figure 21-9  | Hardware Synchronization Action.....                                                         | 600 |
| Figure 21-10 | Pulse Width Measurement.....                                                                 | 601 |
| Figure 21-11 | Period Measurement.....                                                                      | 601 |

|              |                                                                                                     |     |
|--------------|-----------------------------------------------------------------------------------------------------|-----|
| Figure 21-12 | Timing Diagram of Compare Output in Single Buffer Mode.....                                         | 603 |
| Figure 21-13 | Capture Input Timing in Double Buffer Mode.....                                                     | 604 |
| Figure 21-14 | Count Buffer Operation in Sawtooth Mode.....                                                        | 605 |
| Figure 21-15 | Count Buffer Operation 1 in Triangle Mode.....                                                      | 606 |
| Figure 21-16 | Count Buffer Operation 2 in Triangle Mode.....                                                      | 606 |
| Figure 21-17 | Single-Edge Aligned Independent PWM.....                                                            | 607 |
| Figure 21-18 | Double-Edge Symmetrical independent PWM.....                                                        | 608 |
| Figure 21-19 | Software-Set GCMBR Complementary PWM Output.....                                                    | 609 |
| Figure 21-20 | Hardware-Set GCMBR Complementary PWM Output.....                                                    | 609 |
| Figure 21-21 | Double-Edge Asymmetrical PWM Output.....                                                            | 610 |
| Figure 21-22 | Single-Edge Aligned Independent PWM.....                                                            | 611 |
| Figure 21-23 | Double-Edge Symmetrical Complementary PWM with Dead Time.....                                       | 612 |
| Figure 21-24 | Position Mode-Basic Counting.....                                                                   | 613 |
| Figure 21-25 | Position Counting Mode-Phase Difference Counting (1x Counting).....                                 | 613 |
| Figure 21-26 | Position Counting Mode-Phase Difference Counting (2x Counting).....                                 | 614 |
| Figure 21-27 | Position Counting Mode-Phase Difference Counting (4x Counting).....                                 | 614 |
| Figure 21-28 | Position Counting Mode-Direction Counting.....                                                      | 614 |
| Figure 21-29 | Revolution Counting Mode-Z-phase Counting.....                                                      | 615 |
| Figure 21-30 | Revolution Counting Mode-Position Overflow Counting.....                                            | 615 |
| Figure 21-31 | Revolution Counting Mode-Mixed Counting.....                                                        | 615 |
| Figure 21-32 | Revolution Counting Mode-Mixed Count Z-Phase Masking Action Instances 1.....                        | 616 |
| Figure 21-33 | Revolution Counting Mode-Mixed Count Z-Phase Masking Action Instances 2.....                        | 616 |
| Figure 21-34 | Periodic Interval Active Request Signal Behavior.....                                               | 617 |
| Figure 21-35 | Forward Rotation AB Output Timing.....                                                              | 619 |
| Figure 21-36 | Reverse Rotation AB Output Timing.....                                                              | 619 |
| Figure 21-37 | Forward to Reverse Rotation.....                                                                    | 620 |
| Figure 21-38 | Forward Rotation Z Output ( $PTZOSPR < PTZOEPR$ ).....                                              | 621 |
| Figure 21-39 | Reverse Rotation Z Output ( $PTZOSPR > PTZOEPR$ ).....                                              | 621 |
| Figure 21-40 | Forward-to-Reverse Rotation Z Output ( $PTZOSPR=PTZOEPR$ ).....                                     | 622 |
| Figure 21-41 | Forward Rotation Z Output Compressed.....                                                           | 623 |
| Figure 21-42 | Reverse Z-phase Output Compressed.....                                                              | 623 |
| Figure 21-43 | Forward and Reverse Z-phase Output Compressed.....                                                  | 624 |
| Figure 21-44 | AB Output (Without Remainder Compensation).....                                                     | 624 |
| Figure 21-45 | AB Output (With Remainder Compensation).....                                                        | 625 |
| Figure 21-46 | Interrupt & Event Output Example in Sawtooth Wave Mode.....                                         | 628 |
| Figure 22-1  | Timer4 Basic Block Diagram.....                                                                     | 707 |
| Figure 22-2  | Timer4 Sawtooth Waveform.....                                                                       | 708 |
| Figure 22-3  | Timer4 Triangle Waveform.....                                                                       | 708 |
| Figure 22-4  | Timer4 Sawtooth Wave Mode Counting Action.....                                                      | 709 |
| Figure 22-5  | Timer4 Triangle Wave Mode Counting Action.....                                                      | 709 |
| Figure 22-6  | Sawtooth Mode Waveform Output Example.....                                                          | 711 |
| Figure 22-7  | Triangular Wave Mode Waveform Output Example.....                                                   | 711 |
| Figure 22-8  | Modify Sawtooth Count Period When the Buffer is disabled.....                                       | 712 |
| Figure 22-9  | Modify Sawtooth Wave Count Period when Buffer is Enabled.....                                       | 713 |
| Figure 22-10 | Modify Triangular Wave Count Period when Buffer is Enabled.....                                     | 713 |
| Figure 22-11 | OCCR Buffer Data Transmission (Period Interval Response Link Disabled).....                         | 714 |
| Figure 22-12 | OCCR Buffer Data Transmission (Period Interval Response Link Enabled).....                          | 715 |
| Figure 22-13 | Output compare Buffer Data Transfer (OCMR Buffer Enabled).....                                      | 715 |
| Figure 22-14 | SCCR Buffer Transmission Operation (Period Interval Response Link Transmission Disabled).....       | 716 |
| Figure 22-15 | SCCR Buffer Transmission Operation (when Periodic Interval Response Link Transmission Enabled)..... | 717 |
| Figure 22-16 | Sawtooth Independent PWM Output Example.....                                                        | 718 |
| Figure 22-17 | Triangle Independent PWM Output Example.....                                                        | 718 |

|              |                                                                                                                     |     |
|--------------|---------------------------------------------------------------------------------------------------------------------|-----|
| Figure 22-18 | Sawtooth Wave Extended PWM Output.....                                                                              | 719 |
| Figure 22-19 | Software Implementation of Complementary PWM Output.....                                                            | 720 |
| Figure 22-20 | Complementary PWM Output in Dead-Zone Generator Mode.....                                                           | 720 |
| Figure 22-21 | Waveform Output in Dead-Zone Timer Mode with Pulse Width Anomalies.....                                             | 721 |
| Figure 22-22 | Complementary PWM Output in Dead-Zone Timer Filter Mode.....                                                        | 722 |
| Figure 22-23 | Periodic Interval Response Timing Diagram.....                                                                      | 723 |
| Figure 22-24 | Special Event Output Signal Periodic Interval Response Output.....                                                  | 724 |
| Figure 22-25 | PWM Period Output Monitoring Example.....                                                                           | 725 |
| Figure 22-26 | Special Event Output TMR4_<t>_ADSM Timing in Compare Start Mode.....                                                | 726 |
| Figure 22-27 | Output Timing of Special Event Output Signal in Delayed Start Mode.....                                             | 727 |
| Figure 23-1  | EMB Channel Function Block Diagram.....                                                                             | 764 |
| Figure 23-2  | EMB Structure Block Diagram.....                                                                                    | 765 |
| Figure 24-1  | TimerA Basic Block Diagram.....                                                                                     | 786 |
| Figure 24-2  | Sawtooth Waveform (Counting Up).....                                                                                | 787 |
| Figure 24-3  | Triangle Waveform.....                                                                                              | 787 |
| Figure 24-4  | Compare Output Action.....                                                                                          | 788 |
| Figure 24-5  | Capture Input Action.....                                                                                           | 790 |
| Figure 24-6  | Filtering Function of Clock Input Port.....                                                                         | 791 |
| Figure 24-7  | Software Synchronization Action.....                                                                                | 792 |
| Figure 24-8  | Buffer Action in Sawtooth Mode.....                                                                                 | 793 |
| Figure 24-9  | Buffer and Special Buffer Actions in Sawtooth Mode.....                                                             | 793 |
| Figure 24-10 | 32-bit Cascade Counting Action.....                                                                                 | 794 |
| Figure 24-11 | Example of Single-Side Aligned PWM Output.....                                                                      | 794 |
| Figure 24-12 | Example of Double-Side Symmetric PWM Output.....                                                                    | 795 |
| Figure 24-13 | Position Mode-Basic Counting.....                                                                                   | 796 |
| Figure 24-14 | Position Counting Mode-Phase Difference Counting (1x Counting).....                                                 | 796 |
| Figure 24-15 | Position Counting Mode-Phase Difference Counting (2x Counting).....                                                 | 796 |
| Figure 24-16 | Position Counting Mode-Phase Difference Counting (4x Counting).....                                                 | 796 |
| Figure 24-17 | Position Counting Mode-Direction Counting.....                                                                      | 797 |
| Figure 24-18 | Revolution Counting Mode-Z-phase Counting.....                                                                      | 797 |
| Figure 24-19 | Revolution Counting Mode-Position Overflow Counting.....                                                            | 798 |
| Figure 24-20 | Revolution Counting Mode-Mixed Counting.....                                                                        | 798 |
| Figure 25-1  | Timer2 Basic Block Diagram.....                                                                                     | 830 |
| Figure 25-2  | Compare Output Action.....                                                                                          | 832 |
| Figure 25-3  | Hardware Start and Clear Action.....                                                                                | 833 |
| Figure 25-4  | Capture Input Action.....                                                                                           | 833 |
| Figure 25-5  | Pulse Width Measurement.....                                                                                        | 834 |
| Figure 25-6  | Period Measurement.....                                                                                             | 834 |
| Figure 25-7  | Digital Filtering of TRIG Input Port.....                                                                           | 835 |
| Figure 26-1  | Timer0 Basic Block Diagram.....                                                                                     | 852 |
| Figure 26-2  | Timer0 Counting Timing Diagram.....                                                                                 | 853 |
| Figure 27-1  | RTC Basic Block Diagram.....                                                                                        | 872 |
| Figure 28-1  | Hardware Startup Mode Example.....                                                                                  | 899 |
| Figure 28-2  | Software Startup Mode Example.....                                                                                  | 900 |
| Figure 28-3  | Examples of Various Refresh Action Sequences (action confirmation, refresh required signal falling edge, etc.)..... | 900 |
| Figure 28-4  | Counter Underflow Operation Example.....                                                                            | 902 |
| Figure 28-5  | Counter Refresh Operation Example.....                                                                              | 903 |
| Figure 29-1  | USART System Block Diagram.....                                                                                     | 910 |
| Figure 29-2  | UART Data Format.....                                                                                               | 912 |
| Figure 29-3  | UART Transmit Data Diagram 1.....                                                                                   | 913 |
| Figure 29-4  | UART transmit Data Diagram 2.....                                                                                   | 913 |

|              |                                                                                           |      |
|--------------|-------------------------------------------------------------------------------------------|------|
| Figure 29-5  | UART Internal Synchronization and Sample Timing.....                                      | 914  |
| Figure 29-6  | UART Receive Data Diagram 1.....                                                          | 916  |
| Figure 29-7  | UART Receive Data Diagram 2.....                                                          | 916  |
| Figure 29-8  | Half-Duplex Operation Timing.....                                                         | 919  |
| Figure 29-9  | OVER8=1 DE Asserted.....                                                                  | 919  |
| Figure 29-10 | OVER8=0 DE Asserted.....                                                                  | 920  |
| Figure 29-11 | OVER8=1 DE De-assertion.....                                                              | 920  |
| Figure 29-12 | OVER8=0 DE De-assertion.....                                                              | 920  |
| Figure 29-13 | Multiprocessor Communication Legend.....                                                  | 921  |
| Figure 29-14 | Multiprocessor Mode Data Format.....                                                      | 921  |
| Figure 29-15 | Multiprocessor Mode Transmit Data Diagram.....                                            | 923  |
| Figure 29-16 | Multiprocessor Mode Receive Data Diagram 1.....                                           | 924  |
| Figure 29-17 | Multiprocessor Mode Receive Data Diagram 2.....                                           | 924  |
| Figure 29-18 | LIN Bus Data Behavior.....                                                                | 925  |
| Figure 29-19 | Smart Card Connection Schematic Diagram.....                                              | 927  |
| Figure 29-20 | Smart Card Mode Synchronization Timing and Sample Timing Diagram.....                     | 928  |
| Figure 29-21 | Smart Card Mode Data Format.....                                                          | 928  |
| Figure 29-22 | Smart Card Mode Transmit Data Diagram.....                                                | 929  |
| Figure 29-23 | Smart Card Mode Receive Data Diagram.....                                                 | 930  |
| Figure 29-24 | Clock Synchronization Mode Data Format.....                                               | 931  |
| Figure 29-25 | Data Transmission in Clock Synchronization Mode Diagram 1 (LSB/without CTS function)..... | 933  |
| Figure 29-26 | Data Transmission in Clock Synchronization Mode Diagram 2 (LSB/with CTS function).....    | 933  |
| Figure 29-27 | Data Reception in Clock Synchronization Mode Diagram 1 (LSB/without RTS function).....    | 934  |
| Figure 29-28 | Data Reception in Clock Synchronization Mode Diagram 2 (LSB/with RTS function).....       | 934  |
| Figure 29-29 | Digital Filter Function Diagram.....                                                      | 936  |
| Figure 30-1  | I2C System Block Diagram.....                                                             | 964  |
| Figure 30-2  | Example of I2C Bus Structure.....                                                         | 964  |
| Figure 30-3  | I2C Bus Timing Diagram.....                                                               | 965  |
| Figure 30-4  | Data Format of I2C Bus.....                                                               | 966  |
| Figure 30-5  | Sequence Diagram of Master Transmitting Data in 7-bit Address Format (Example).....       | 967  |
| Figure 30-6  | Sequence Diagram of Master Receiving Data in 7-bit Address Format (Example).....          | 968  |
| Figure 30-7  | Sequence Diagram of Slave Transmitting Mode in 7-bit Address Format (Example).....        | 968  |
| Figure 30-8  | Sequence Diagram of Slave Receiving Data in 7-bit Address Format (Example).....           | 969  |
| Figure 30-9  | SCL Synchronous Timing.....                                                               | 970  |
| Figure 30-10 | Transmitting Timing Diagram of Slave (1).....                                             | 971  |
| Figure 30-11 | Transmitting Timing Diagram of Slave (2).....                                             | 971  |
| Figure 30-12 | Slave Receiving Timing Diagram.....                                                       | 972  |
| Figure 30-13 | Fast ACK/NACK Timing Diagram.....                                                         | 972  |
| Figure 30-14 | Timing when Selecting 7-bit Address Format.....                                           | 973  |
| Figure 30-15 | Timing when Selecting 10-bit Address Format.....                                          | 973  |
| Figure 30-16 | Broadcast Address Matching Timing Diagram.....                                            | 974  |
| Figure 30-17 | SMBus Master Address Matching Timing Diagram.....                                         | 974  |
| Figure 30-18 | SMBus Alarm Response Address Matching Timing Diagram.....                                 | 975  |
| Figure 30-19 | SMBus Default Address Timing Diagram.....                                                 | 975  |
| Figure 30-20 | Digital Filtering Circuit Block Diagram.....                                              | 977  |
| Figure 31-1  | SPI System Block Diagram.....                                                             | 1007 |
| Figure 31-2  | Master Mode Structure.....                                                                | 1009 |
| Figure 31-3  | Three-wire Clock Synchronization Operation.....                                           | 1010 |
| Figure 31-4  | Data Format.....                                                                          | 1011 |
| Figure 31-5  | MSB Transmitted First, Parity Invalid.....                                                | 1011 |
| Figure 31-6  | LSB Transmitted First, Parity Invalid.....                                                | 1012 |
| Figure 31-7  | MSB Transmitted First, Parity Valid.....                                                  | 1012 |

|              |                                                                                                     |      |
|--------------|-----------------------------------------------------------------------------------------------------|------|
| Figure 31-8  | LSB Transmitted First, Parity Valid.....                                                            | 1013 |
| Figure 31-9  | Data Transmission Format Diagram (CPHA=0).....                                                      | 1013 |
| Figure 31-10 | Data Transmission Format Diagram (CPHA=1).....                                                      | 1014 |
| Figure 31-11 | Full-duplex Synchronous Serial Communication.....                                                   | 1014 |
| Figure 31-12 | Transmit-only Communication.....                                                                    | 1015 |
| Figure 31-13 | Normal Communication.....                                                                           | 1016 |
| Figure 31-14 | Continuous Communication.....                                                                       | 1016 |
| Figure 31-15 | Parity Check Process.....                                                                           | 1023 |
| Figure 31-16 | Overload Error Processing.....                                                                      | 1025 |
| Figure 31-17 | Schematic Diagram of Action when Clock Automatic Stop Function Enabled (CPHA=1).....                | 1026 |
| Figure 31-18 | Schematic Diagram of Action when Clock Automatic Stop Function Enabled (CPHA=0).....                | 1026 |
| Figure 31-19 | Parity Error.....                                                                                   | 1027 |
| Figure 32-1  | Default Area Setting and AHB Bus Space Memory Mapping Relationship Diagram.....                     | 1040 |
| Figure 32-2  | QSPI-ROM Space Memory Image.....                                                                    | 1041 |
| Figure 32-3  | Module Composition Diagram of QSPI.....                                                             | 1042 |
| Figure 32-4  | Diagram of Extended SPI Protocol Action 1 (Fast Read Mode).....                                     | 1043 |
| Figure 32-5  | Diagram of Extended SPI Protocol Action 2 (Four-Wire Input/Output Fast Read Mode).....              | 1043 |
| Figure 32-6  | Diagram of Two-wire SPI Protocol Action (Fast Read Mode).....                                       | 1043 |
| Figure 32-7  | Diagram Four-Wire SPI Protocol Action (Fast Read Mode).....                                         | 1044 |
| Figure 32-8  | Basic Timing Diagram of Serial Interface.....                                                       | 1044 |
| Figure 32-9  | Diagram of Output Clock Duty Cycle Correction when HCLK is Divided by 3 as the Reference Clock..... | 1045 |
| Figure 32-10 | QSSN Setup Time Configuration Diagram.....                                                          | 1046 |
| Figure 32-11 | QSSN Hold Time Configuration Diagram.....                                                           | 1046 |
| Figure 32-12 | Diagram of Data Receiving Delay.....                                                                | 1047 |
| Figure 32-13 | Diagram of a Standard Read Bus Cycle.....                                                           | 1048 |
| Figure 32-14 | Fast Read Bus Cycle Diagram.....                                                                    | 1048 |
| Figure 32-15 | Fast Read Bus Cycle for Selecting XIP Mode Diagram.....                                             | 1049 |
| Figure 32-16 | Two-Wire Output Fast Read Bus Cycle Diagram.....                                                    | 1050 |
| Figure 32-17 | Two-Wire Output Fast Read Bus Cycle for Selecting XIP Mode Diagram.....                             | 1050 |
| Figure 32-18 | Two-Wire Input and Output Fast Read Bus Cycle Diagram.....                                          | 1051 |
| Figure 32-19 | Two-Wire Input and Output Fast Read Bus Cycle for Selecting XIP Mode Diagram.....                   | 1051 |
| Figure 32-20 | Four-Wire Output Fast Read Bus Cycle Diagram.....                                                   | 1052 |
| Figure 32-21 | Four-Wire Output Fast Read Bus Cycle for Selecting XIP Mode Diagram.....                            | 1052 |
| Figure 32-22 | Four-Wire Input and Output Fast Read Bus Cycle Diagram.....                                         | 1053 |
| Figure 32-23 | Four-Wire Input and Output Fast Read Bus Cycle for Selecting XIP Mode Diagram.....                  | 1053 |
| Figure 32-24 | Diagram of Instruction Bus Cycle when Entering 4-Bytes Mode.....                                    | 1054 |
| Figure 32-25 | Exit 4-Byte Mode Instruction Bus Period Diagram.....                                                | 1054 |
| Figure 32-26 | Diagram of Write Mode Instruction Bus Period.....                                                   | 1055 |
| Figure 32-27 | Diagram of Single Flash Memory Data Read Operation with Independent Conversion.....                 | 1055 |
| Figure 32-28 | Diagram of Data Reading Operation when Pre-fetch Function Enable.....                               | 1056 |
| Figure 32-29 | Diagram of Data Read Operation by QSPI Bus Cycle Extension Function.....                            | 1057 |
| Figure 32-30 | XIP Mode Control Diagram.....                                                                       | 1058 |
| Figure 33-1  | I2S System Block Diagram.....                                                                       | 1073 |
| Figure 33-2  | I2S Philips Protocol Waveform (16/32 bit Full Precision).....                                       | 1075 |
| Figure 33-3  | I2S Philips Protocol Waveform (16-bit Data Encapsulated in 32-bit Frame).....                       | 1075 |
| Figure 33-4  | I2S Philips Protocol Waveform (24-bit Data Encapsulated in 32-bit Frame).....                       | 1075 |
| Figure 33-5  | I2S MSB Protocol Waveform (16/32-bit Full Precision).....                                           | 1076 |
| Figure 33-6  | I2S MSB Protocol Waveform (16-bit Data Encapsulated in 32-bit Frame).....                           | 1076 |
| Figure 33-7  | I2S MSB Protocol Waveform (24-bit Data Encapsulated in 32-bit Frame).....                           | 1076 |
| Figure 33-8  | I2S LSB Protocol Waveform (16/32-bit Full Precision).....                                           | 1077 |
| Figure 33-9  | I2S LSB Protocol Waveform (16-bit Data Encapsulated in 32-bit Frame).....                           | 1077 |
| Figure 33-10 | I2S LSB Protocol Waveform (24-bit Data Encapsulated in 32-bit Frame).....                           | 1077 |

|                                                                                                   |      |
|---------------------------------------------------------------------------------------------------|------|
| Figure 33-11 I2S PCM Protocol Waveform (16/32-bit Full Precision).....                            | 1078 |
| Figure 33-12 I2S PCM Protocol Waveform (16-bit Data Encapsulated in 32-bit Frame).....            | 1078 |
| Figure 33-13 I2S PCM Protocol Waveform (24-bit Data Encapsulated in 32-bit Frame).....            | 1078 |
| Figure 33-14 Audio Sampling Frequency Definition.....                                             | 1079 |
| Figure 33-15 Clock Generator Architecture.....                                                    | 1079 |
| Figure 33-16 Master only Receives and Temporarily Stops Receiving.....                            | 1092 |
| Figure 33-17 PCM Short Frame Master Resends After Suspension of Transmission Method 1.....        | 1093 |
| Figure 33-18 PCM Short Frame Master Resends After Suspension of Transmission Method 2.....        | 1093 |
| Figure 34-1 USBHS System Block Diagram.....                                                       | 1098 |
| Figure 34-2 USBHS host Mode System Architecture Diagram.....                                      | 1100 |
| Figure 34-3 USBHS Device Mode System Architecture Diagram.....                                    | 1103 |
| Figure 34-4 Schematic Diagram of USBHS Dynamic Update for USBHS_HFIR Register.....                | 1107 |
| Figure 34-5 FIFO Architecture Diagram in Host Mode.....                                           | 1107 |
| Figure 34-6 Schematic Diagram of FIFO Architecture in USBHS Device Mode.....                      | 1109 |
| Figure 34-7 USBHS Control and Status Register Memory Mapping.....                                 | 1133 |
| Figure 35-1 USBFS System Block Diagram.....                                                       | 1230 |
| Figure 35-2 USBFS Host Mode System Architecture Diagram.....                                      | 1231 |
| Figure 35-3 USBFS Device Mode System Architecture Diagram.....                                    | 1235 |
| Figure 35-4 Schematic Diagram of USBFS Dynamic Update for USBFS_HFIR Register.....                | 1239 |
| Figure 35-5 Schematic Diagram of FIFO Architecture in Host Mode.....                              | 1239 |
| Figure 35-6 Schematic Diagram of FIFO Architecture in USBFS Device Mode.....                      | 1241 |
| Figure 35-7 USBFS Control and Status Register Memory Mapping.....                                 | 1264 |
| Figure 36-1 CAN FD System Block Diagram.....                                                      | 1349 |
| Figure 36-2 CAN FD LBMI and LBME Schematic Diagram.....                                           | 1350 |
| Figure 36-3 CAN Bit Time Definition Diagram.....                                                  | 1351 |
| Figure 36-4 CAN FD TBUF Register Write Transmit Buffer and Schematic Diagram.....                 | 1354 |
| Figure 36-5 CAN RBUF Register Read Receive Buffer Schematic Diagram.....                          | 1355 |
| Figure 36-6 CAN ACF Register Access Filter Group Schematic Diagram.....                           | 1356 |
| Figure 36-7 TDC Functional Block Diagram.....                                                     | 1359 |
| Figure 37-1 MCAN System Block Diagram.....                                                        | 1403 |
| Figure 37-2 Transceiver Delay Measurement.....                                                    | 1408 |
| Figure 37-3 Pin Control in Bus Monitoring Mode.....                                               | 1410 |
| Figure 37-4 External Loopback Mode Pin Control.....                                               | 1412 |
| Figure 37-5 Pin Control for Internal Loopback Mode.....                                           | 1412 |
| Figure 37-6 Standard Message ID Filter Path.....                                                  | 1415 |
| Figure 37-7 Extended Message ID Filter Path.....                                                  | 1416 |
| Figure 37-8 Receive FIFO Status.....                                                              | 1417 |
| Figure 37-9 Receive FIFO Overflow Processing.....                                                 | 1418 |
| Figure 37-10 Example of Mixed Configuration of Dedicated Transmit Buffers and Transmit FIFO.....  | 1423 |
| Figure 37-11 Example of Mixed Configuration of Dedicated Transmit Buffers and Transmit Queue..... | 1423 |
| Figure 37-12 Message RAM Configuration.....                                                       | 1426 |
| Figure 38-1 Ethernet MAC Controller Architecture Diagram.....                                     | 1504 |
| Figure 38-2 MII Interface Connection Diagram.....                                                 | 1505 |
| Figure 38-3 RMII Interface Connection Diagram.....                                                | 1507 |
| Figure 38-4 SMI Interface Connection Diagram.....                                                 | 1507 |
| Figure 38-5 MAC Frame Structure.....                                                              | 1510 |
| Figure 38-6 MII/RMII Transmission Bit Order.....                                                  | 1516 |
| Figure 38-7 Collision-Free Transmission Diagram.....                                              | 1516 |
| Figure 38-8 Collision Transmission Diagram.....                                                   | 1516 |
| Figure 38-9 Transmission Diagram in MII/RMII Mode.....                                            | 1517 |
| Figure 38-10 MII/RMII Reception Bit Order.....                                                    | 1519 |
| Figure 38-11 Error-Free Transmission Diagram.....                                                 | 1520 |



|                                                                                                                                                          |      |
|----------------------------------------------------------------------------------------------------------------------------------------------------------|------|
| Figure 38-12 Error Transmission Diagram.....                                                                                                             | 1520 |
| Figure 38-13 Reception Diagram Under False Carrier Indication.....                                                                                       | 1520 |
| Figure 38-14 Remote Wake-Up Frame Filter Registers.....                                                                                                  | 1524 |
| Figure 38-15 Clock Synchronization Diagram.....                                                                                                          | 1527 |
| Figure 38-16 System Time Calibration.....                                                                                                                | 1529 |
| Figure 38-17 Descriptor Structure.....                                                                                                                   | 1532 |
| Figure 38-18 TxDMA Workflow (Non-OSF Mode).....                                                                                                          | 1536 |
| Figure 38-19 TxDMA Workflow (OSF Mode).....                                                                                                              | 1537 |
| Figure 38-20 RxDMA Workflow.....                                                                                                                         | 1547 |
| Figure 38-21 ETHMAC Interrupt Scheme.....                                                                                                                | 1556 |
| Figure 38-22 DMA Interrupt Components.....                                                                                                               | 1557 |
| Figure 40-1 EXMC Architecture Diagram.....                                                                                                               | 1687 |
| Figure 40-2 External Address Space Allocation.....                                                                                                       | 1689 |
| Figure 40-3 SMC Address Space Partitioning.....                                                                                                          | 1690 |
| Figure 40-4 DMC Address Space Division.....                                                                                                              | 1690 |
| Figure 40-5 NFC Address Space Division.....                                                                                                              | 1691 |
| Figure 40-6 SMC State Diagram.....                                                                                                                       | 1695 |
| Figure 40-7 SMC Initial Setup Process.....                                                                                                               | 1696 |
| Figure 40-8 Single Read Operation Basic Timing (Asynchronous Mode RSYN=0b0 & 8-bit Data Width MW=0b00).....                                              | 1698 |
| Figure 40-9 Single Read Operation Basic Timing (Asynchronous Mode RSYN=0b0 & 16-bit Data Width MW=0b01).....                                             | 1699 |
| Figure 40-10 Single Read Operation Basic Timing (Asynchronous Mode RSYN=0b0 & 32-bit Data Width MW=0b10).....                                            | 1700 |
| Figure 40-11 Basic Timing for Single Read Action (Synchronous Mode (RSYN=0b1) & 8-bit Width (MW=0b00) ).....                                             | 1701 |
| Figure 40-12 Basic Timing for Single Read Action (Synchronous Mode (RSYN=0b1) & 16-bit Width (MW=0b01) ).....                                            | 1702 |
| Figure 40-13 Basic Timing for Single Read Action (Synchronous Mode (RSYN=0b1) & 32-bit Width (MW=0b10) ).....                                            | 1703 |
| Figure 40-14 Basic Timing for Single Read Action with Address and Data Line Multiplexing (8-bit Width (MW=0b00) ).....                                   | 1704 |
| Figure 40-15 Basic Timing for Single Read Action with Address and Data Line Multiplexing (16-bit Width (MW=0b01) )....                                   | 1705 |
| Figure 40-16 Basic Timing for Single Read Action with Address and Data Line Multiplexing (32-bit Width (MW=0b10) )....                                   | 1706 |
| Figure 40-17 Single Write Operation Basic Timing (Asynchronous Mode WSYN=0b0, 8-bit Data Width MW=0b00, BLSS=0b0).....                                   | 1707 |
| Figure 40-18 Single Write Operation Basic Timing (Asynchronous Mode WSYN=0b0, 16-bit Data Width MW=0b01, BLSS=0b1).....                                  | 1708 |
| Figure 40-19 Single Write Operation Basic Timing (Asynchronous Mode WSYN=0b0, 32-bit Data Width MW=0b10, BLSS=0b1).....                                  | 1709 |
| Figure 40-20 Basic Timing for Single Write Action (Synchronous Mode (WSYN=0b1) & 8-bit Width (MW=0b00) & BLSS=0b1).....                                  | 1710 |
| Figure 40-21 Basic Timing for Single Write Action (Synchronous Mode (WSYN=0b1) & 16-bit Width (MW=0b01) & BLSS=0b0).....                                 | 1711 |
| Figure 40-22 Basic Timing for Single Write Action (Synchronous Mode (WSYN=0b1) & 32-bit Width (MW=0b10) & BLSS=0b0).....                                 | 1712 |
| Figure 40-23 Basic Timing for Single Write Action with Address and Data Line Multiplexing (Asynchronous mMode (WSYN=0b0) & 16-bit Width (MW=0b01) )..... | 1713 |
| Figure 40-24 Basic Timing for Single Write Action with Address and Data Line Multiplexing (Synchronous Mode (WSYN=0b1) & 16-bit Width (MW=0b01) ).....   | 1714 |
| Figure 40-25 Timing from Row Activation to Read and Write Operations.....                                                                                | 1715 |
| Figure 40-26 Command-based Precharge.....                                                                                                                | 1716 |
| Figure 40-27 Auto-precharge after Read and Write.....                                                                                                    | 1716 |
| Figure 40-28 Timing for Entering and Exiting the Self-refresh Action.....                                                                                | 1717 |
| Figure 40-29 DMC State Diagram.....                                                                                                                      | 1718 |
| Figure 40-30 DMC Initial Setup Process.....                                                                                                              | 1719 |
| Figure 40-31 Basic Timing for Single Read Action (16-bit Width (DMCMW=0b00) ).....                                                                       | 1720 |
| Figure 40-32 Basic Timing for Single Write Action (32-bit Width (DMCMW=0b01) ).....                                                                      | 1721 |
| Figure 40-33 Basic Timing for Burst Read Action (32-bit Width (DMCMW=0b10) & BURST=0b010).....                                                           | 1722 |

|                                                                                                 |      |
|-------------------------------------------------------------------------------------------------|------|
| Figure 40-34 Basic Timing for Burst Write Action (32-bit Width (DMCMW=0b10) & BURST=0b011)..... | 1722 |
| Figure 40-35 DMC_CKE Port Output Control Timing.....                                            | 1723 |
| Figure 40-36 NFC timing control diagram 1.....                                                  | 1768 |
| Figure 40-37 NFC Timing Control Diagram 2.....                                                  | 1769 |
| Figure 40-38 External Feedback Clock Access.....                                                | 1775 |
| Figure 41-1 DVP Basic Block Diagram.....                                                        | 1777 |
| Figure 41-2 DVP Signal Waveform Diagram.....                                                    | 1779 |
| Figure 41-3 Single Frame Mode Data Acquisition Action.....                                      | 1780 |
| Figure 41-4 Continuous Mode Data Acquisition Action.....                                        | 1781 |
| Figure 41-5 Frame Acquisition Frequency Control.....                                            | 1781 |
| Figure 41-6 DVP Window Configuration Diagram.....                                               | 1783 |
| Figure 41-7 Window Cropping Data Acquisition Diagram.....                                       | 1783 |
| Figure 42-1 HASH Algorithm Flowchart.....                                                       | 1795 |
| Figure 42-2 TRNG System Block Diagram.....                                                      | 1801 |
| Figure 42-3 PKE System Block Diagram.....                                                       | 1804 |
| Figure 42-4 SKE System Block Diagram.....                                                       | 1807 |
| Figure 43-1 CRC Application Diagram.....                                                        | 1850 |
| Figure 44-1 Hardware Trigger Start Schematic Diagram.....                                       | 1860 |
| Figure 44-2 Triangular Wave Output Mode Waveform.....                                           | 1861 |
| Figure 44-3 Triangular Wave Output Mode.....                                                    | 1861 |
| Figure 44-4 Incremental Sawtooth Wave Output Mode Waveform.....                                 | 1862 |
| Figure 44-5 Incremental Sawtooth Wave Output Mode.....                                          | 1863 |
| Figure 44-6 Decremental Sawtooth Wave Output Mode Waveform.....                                 | 1864 |
| Figure 44-7 Decrement Sawtooth Wave Output Mode.....                                            | 1864 |
| Figure 45-1 Schematic Diagram Of Sine Operation Angle Value.....                                | 1884 |
| Figure 46-1 FMAC Basic Block Diagram.....                                                       | 1890 |
| Figure 47-1 ERMU Block Diagram.....                                                             | 1899 |
| Figure 48-1 Debug Control System Block Diagram.....                                             | 1919 |
| Figure 48-2 Debug Control System.....                                                           | 1920 |
| Figure 48-3 JTAG-DP to SW-DP Switching Timing.....                                              | 1921 |
| Figure 48-4 TPIU Block Diagram.....                                                             | 1923 |



## List of Tables

|            |                                                                                                        |     |
|------------|--------------------------------------------------------------------------------------------------------|-----|
| Table 2-1  | Memory Mapping.....                                                                                    | 51  |
| Table 2-2  | Target Address Configuration Main Flash Example (MMF_REMCR0.EN=1 or MMF_REMCR1.EN=1).....              | 57  |
| Table 2-3  | Target Address Configuration SRAMH Example (MMF_REMCR0.EN=1 or MMF_REMCR1.EN=1).....                   | 57  |
| Table 2-4  | Memory Mapping Register Offset.....                                                                    | 58  |
| Table 4-1  | Reset Mode and Generating Conditions.....                                                              | 64  |
| Table 4-2  | Reset Mode and Reset Flag.....                                                                         | 65  |
| Table 4-3  | Reset Conditions for Each Module.....                                                                  | 72  |
| Table 4-4  | Reset Module Unit (RMU) Register Offset.....                                                           | 75  |
| Table 5-1  | Clock Source Main Features.....                                                                        | 88  |
| Table 5-2  | Specifications for Each Internal Clock.....                                                            | 91  |
| Table 5-3  | Clock Frequency Measurement (FCM) Register Offset.....                                                 | 104 |
| Table 5-4  | Clock Module Unit (CMU) Register Offset.....                                                           | 104 |
| Table 6-1  | BOR Configuration.....                                                                                 | 154 |
| Table 6-2  | PVD1/PVD2 Features.....                                                                                | 155 |
| Table 6-4  | RUN Mode.....                                                                                          | 160 |
| Table 6-5  | Low Power Mode.....                                                                                    | 160 |
| Table 6-6  | Operating Conditions of Low Power Mode and Status of Each Module in Low Power Mode.....                | 161 |
| Table 6-7  | RUN Mode Description.....                                                                              | 163 |
| Table 6-8  | Power-down Mode Submodes.....                                                                          | 167 |
| Table 6-10 | RAM Module and RAM Power-down Control Bit.....                                                         | 170 |
| Table 6-11 | Register Protection List.....                                                                          | 172 |
| Table 6-12 | Power Control (PWC) Register Offset.....                                                               | 172 |
| Table 7-1  | Initialization Configuration (ICG) Register Offset.....                                                | 213 |
| Table 8-1  | Compare Table of CPU Clock Frequency (HCLK) and Flash Read Waiting Cycles.....                         | 222 |
| Table 8-2  | Actual Flash Read Cycle Count.....                                                                     | 223 |
| Table 8-3  | OTP Address Distribution Table.....                                                                    | 228 |
| Table 8-4  | Embedded Flash Memory (EFM) Register Offset.....                                                       | 233 |
| Table 9-1  | Relationship Between Waiting Period Setting of SRAM Read and Write Access and CPU Clock Frequency..... | 269 |
| Table 9-2  | SRAM Space Allocation.....                                                                             | 270 |
| Table 9-3  | Built-in SRAM (SRAM) Register Offset.....                                                              | 272 |
| Table 10-1 | PORT Register List for 32-bit Access.....                                                              | 307 |
| Table 10-2 | General-Purpose IO (GPIO) Register Offset.....                                                         | 307 |
| Table 10-3 | Event Port Register Offset.....                                                                        | 308 |
| Table 11-1 | External Interrupt Input Pin.....                                                                      | 327 |
| Table 11-2 | Interrupt Vector Digraph.....                                                                          | 328 |
| Table 11-4 | Interrupt Controller (INTC) Register Offset.....                                                       | 351 |
| Table 12-1 | AOS Target List.....                                                                                   | 371 |
| Table 12-2 | Automatic Operation System (AOS) Register Offset.....                                                  | 375 |
| Table 13-1 | MPU Module Introduction.....                                                                           | 398 |
| Table 13-2 | Memory Protection Unit (MPU) Register Offset.....                                                      | 401 |
| Table 14-1 | KEYSCAN Pin Description.....                                                                           | 437 |
| Table 14-2 | Keyboard Scan Control Module (KEYSCAN) Register Offset.....                                            | 439 |
| Table 15-1 | Measurement Error for HRC Target Frequency of 20MHz.....                                               | 445 |
| Table 15-2 | Measurement Error for HRC Target Frequency of 16MHz.....                                               | 445 |
| Table 15-3 | Clock Trimming Controller (CTC) Register Offset.....                                                   | 449 |
| Table 16-1 | Channel Reset Description.....                                                                         | 460 |
| Table 16-2 | DMA Controller (DMA) Register Offset.....                                                              | 467 |
| Table 17-1 | CMP Pin List.....                                                                                      | 502 |
| Table 17-2 | CMP Positive Input List.....                                                                           | 503 |
| Table 17-3 | CMP Negative Input List.....                                                                           | 503 |

|                                                                                                                        |     |
|------------------------------------------------------------------------------------------------------------------------|-----|
| Table 17-4 Voltage Comparator (CMP) Register Offset.....                                                               | 508 |
| Table 17-17 Blank Window PWM List.....                                                                                 | 515 |
| Table 18-1 Specifications of Each ADC Unit.....                                                                        | 519 |
| Table 18-2 Various Conflicts Between Sequences A and B.....                                                            | 525 |
| Table 18-3 ADC Conversion Time.....                                                                                    | 528 |
| Table 18-4 Analog-to-Digital Converter (ADC) Register Offset.....                                                      | 538 |
| Table 18-30 Table of Different ADC Unit CHx Mapping Relationships.....                                                 | 553 |
| Table 19-1 DAC pin.....                                                                                                | 571 |
| Table 19-2 Digital-to-Analog Converter (DAC) Register Offset.....                                                      | 575 |
| Table 19-11 DAC Conversion and Analog Output Control.....                                                              | 579 |
| Table 20-1 OTS Preset Temperature Data.....                                                                            | 583 |
| Table 20-2 E <sub>hrc</sub> Usage and Setting Method.....                                                              | 583 |
| Table 20-3 OTS Preset Data Register Offset.....                                                                        | 585 |
| Table 20-4 On-chip Temperature Sensor (OTS) Register Offset.....                                                       | 585 |
| Table 21-1 Basic Functions and Features of Timer6.....                                                                 | 591 |
| Table 21-2 Timer6 Port List.....                                                                                       | 592 |
| Table 21-3 Function Compare Table in Different Modes.....                                                              | 626 |
| Table 21-4 Advanced Control Timer (Timer6) Register Offset.....                                                        | 635 |
| Table 21-5 Timer6 Common Register Offset.....                                                                          | 637 |
| Table 21-90 CNTERR Control Priority.....                                                                               | 703 |
| Table 21-91 PWMA Port Output Control Priority.....                                                                     | 703 |
| Table 21-92 PWMB Port Output Control Priority.....                                                                     | 704 |
| Table 22-1 Basic Functions and Features of Timer4.....                                                                 | 706 |
| Table 22-2 Timer4 Port List.....                                                                                       | 707 |
| Table 22-3 Internal Signal of General Comparator Output.....                                                           | 710 |
| Table 22-4 Corresponding Relationship Between Internal Signal of General Comparator Output and Port Output Control.... | 717 |
| Table 22-5 General Control Timer (Timer4) Register Offset.....                                                         | 728 |
| Table 22-21 OCCR*h Count Match List.....                                                                               | 741 |
| Table 22-24 OCCR*h/1 Count Match List.....                                                                             | 744 |
| Table 23-1 EMB Port Allocation.....                                                                                    | 766 |
| Table 23-2 EMB Group Mapping Table.....                                                                                | 767 |
| Table 23-3 Emergency Brake Module (EMB) Register Offset.....                                                           | 769 |
| Table 24-1 Basic Functions and Features of TimerA.....                                                                 | 785 |
| Table 24-2 TimerA Port List.....                                                                                       | 786 |
| Table 24-3 TRGSEL Selection Relationship Table for Internal Trigger Events.....                                        | 789 |
| Table 24-4 General Timer (TimerA) Register Offset.....                                                                 | 800 |
| Table 25-1 Basic Functions and Features of Timer2.....                                                                 | 829 |
| Table 25-2 Timer2 Port List.....                                                                                       | 830 |
| Table 25-3 General Timer (Timer2) Register Offset.....                                                                 | 836 |
| Table 26-1 Counter and PWM Synchronization Start Status.....                                                           | 854 |
| Table 26-2 General Timer (Timer0) Register Offset.....                                                                 | 856 |
| Table 26-3 TMR_SYNENR Register Offset.....                                                                             | 856 |
| Table 27-1 RTC Main Features.....                                                                                      | 871 |
| Table 27-2 Real Time Clock (RTC) Register Offset.....                                                                  | 877 |
| Table 28-1 Basic Features of Watchdog Timer.....                                                                       | 898 |
| Table 28-2 Watchdog Timer (WDT) Register Offset.....                                                                   | 904 |
| Table 28-3 Watchdog Timer (SWDT) Register Offset.....                                                                  | 904 |
| Table 29-2 USART pin description.....                                                                                  | 910 |
| Table 29-3 Tolerance of UART Receiver.....                                                                             | 915 |
| Table 29-5 UART Interrupt/Event Table.....                                                                             | 920 |
| Table 29-6 Multiprocessor Mode Interrupt/Event Table.....                                                              | 924 |
| Table 29-7 LIN Interrupt/Event Table.....                                                                              | 927 |

|                                                                                                  |      |
|--------------------------------------------------------------------------------------------------|------|
| Table 29-8 Smart Card Mode Interrupt/Event Table.....                                            | 930  |
| Table 29-9 Interrupt/Event Table in Synchronous Clock Mode.....                                  | 935  |
| Table 29-10 USART Global Interrupt Summary Table.....                                            | 937  |
| Table 29-11 USART1 Filter Control Register Offset.....                                           | 938  |
| Table 29-12 Universal Synchronous Asynchronous Receiver/Transmitter (USART) Register Offset..... | 938  |
| Table 29-21 Baud Rate Calculation Formula (Fractional Baud Rate is disable where FBME=0).....    | 945  |
| Table 29-22 Baud Rate Calculation Formula (Fractional Baud Rate is enable where FBME=1).....     | 945  |
| Table 30-1 Input/Output Pin.....                                                                 | 965  |
| Table 30-2 Interrupt List.....                                                                   | 977  |
| Table 30-3 Event Signal Output List.....                                                         | 978  |
| Table 30-4 Inter-Integrated Circuit (I2C) Register Offset.....                                   | 980  |
| Table 31-1 SPI Main Features.....                                                                | 1005 |
| Table 31-2 Pin Description.....                                                                  | 1007 |
| Table 31-3 SPI Pin Status Description in Master Mode.....                                        | 1008 |
| Table 31-4 SPI Pin Status Description in Slave Mode.....                                         | 1008 |
| Table 31-5 Bit Speed of Some Set Values.....                                                     | 1010 |
| Table 31-6 SPI Mode and Register Setting Relationship.....                                       | 1017 |
| Table 31-7 Error Detection Correspondence Table.....                                             | 1023 |
| Table 31-8 SPI Interrupt Source and Trigger Source Description.....                              | 1028 |
| Table 31-9 Serial Peripheral Interface (SPI) Register Offset.....                                | 1029 |
| Table 32-1 QSPI Main Specifications.....                                                         | 1039 |
| Table 32-2 QSPI pin.....                                                                         | 1042 |
| Table 32-3 Reference Instruction List.....                                                       | 1047 |
| Table 32-4 Pin Status of QSIO2 and QSIO3.....                                                    | 1059 |
| Table 32-5 Quad Serial Peripheral Interface (QSPI) Register Offset.....                          | 1061 |
| Table 33-1 I2S Main Features.....                                                                | 1072 |
| Table 33-2 I2S Pin Description.....                                                              | 1073 |
| Table 33-3 Audio Frequency Accuracy (For VCO Input Frequency=1MHz).....                          | 1080 |
| Table 33-4 I2S Interrupt Request.....                                                            | 1082 |
| Table 33-5 Inter_IC Sound Bus (I2S) Register Offset.....                                         | 1084 |
| Table 34-1 USBHS Pin Description.....                                                            | 1098 |
| Table 34-2 USBHS_GLB Interrupt Event List.....                                                   | 1112 |
| Table 34-3 USB System Control Register Offset.....                                               | 1134 |
| Table 34-4 USB 2.0 High-Speed Module USBHS Register Offset.....                                  | 1134 |
| Table 35-1 USBFS Pin Description.....                                                            | 1230 |
| Table 35-2 USBFS_GLB Interrupt Event List.....                                                   | 1243 |
| Table 35-3 USB System Control Register Offset.....                                               | 1265 |
| Table 35-4 USB2.0 Full-Speed Module USBFS Register Offset.....                                   | 1265 |
| Table 36-1 CAN Pin Description.....                                                              | 1349 |
| Table 36-2 CAN Bit Time Setting Rules.....                                                       | 1352 |
| Table 36-3 Baud Rate Setting Recommendations for 20MHz Communication Clock.....                  | 1352 |
| Table 36-4 Baud Rate Setting Recommendations for 40MHz Communication Clock.....                  | 1352 |
| Table 36-5 Baud Rate Setting Recommendations for 80MHz Communication Clock.....                  | 1353 |
| Table 36-6 CAN RAM Address Mapping.....                                                          | 1360 |
| Table 36-7 Software Reset Range Table.....                                                       | 1360 |
| Table 36-8 CAN Interrupt Table.....                                                              | 1362 |
| Table 36-9 CAN Global Control Register Offset.....                                               | 1363 |
| Table 36-10 CAN FD Controller (CAN FD) Register Offset.....                                      | 1363 |
| Table 36-11 Standard Format CAN Receive Mailbox Format.....                                      | 1365 |
| Table 36-12 Extended Format CAN Receive Mailbox Format.....                                      | 1366 |
| Table 36-14 DLC Control Bit.....                                                                 | 1367 |
| Table 36-15 Standard Format CAN Transmit Mailbox Format.....                                     | 1368 |

|             |                                                                                                                 |      |
|-------------|-----------------------------------------------------------------------------------------------------------------|------|
| Table 36-16 | Extended Format CAN Transmit Mailbox Format.....                                                                | 1369 |
| Table 37-1  | MCAN Pin Description.....                                                                                       | 1404 |
| Table 37-2  | MCAN Configuration Register List.....                                                                           | 1405 |
| Table 37-3  | DLC Encoding in MCAN.....                                                                                       | 1407 |
| Table 37-4  | Receive buffer and FIFO element size.....                                                                       | 1417 |
| Table 37-5  | Receive Buffer Filter Configuration Example.....                                                                | 1419 |
| Table 37-6  | Debug Message Filter Configuration Example.....                                                                 | 1420 |
| Table 37-7  | Possible configurations of the send frame.....                                                                  | 1420 |
| Table 37-8  | Transmit Buffer, FIFO (queue) Element Size.....                                                                 | 1421 |
| Table 37-9  | Receive Buffers and FIFO Elements.....                                                                          | 1427 |
| Table 37-10 | Receive Buffer and FIFO Elements Description.....                                                               | 1427 |
| Table 37-11 | Transmit Buffer Elements.....                                                                                   | 1429 |
| Table 37-12 | Transmit Buffer Elements Description.....                                                                       | 1430 |
| Table 37-13 | Transmit Event FIFO Elements.....                                                                               | 1432 |
| Table 37-14 | Transmit Event FIFO Element Description.....                                                                    | 1432 |
| Table 37-15 | Standard Message ID Filter Element.....                                                                         | 1434 |
| Table 37-16 | Standard Message ID Filter Element Description.....                                                             | 1434 |
| Table 37-17 | Extended Message ID Filter Element.....                                                                         | 1435 |
| Table 37-18 | Extended Message ID Filter Element Description.....                                                             | 1435 |
| Table 37-19 | Controller Area Network (MCAN) Register Offset.....                                                             | 1437 |
| Table 37-20 | MCANRAM register offset.....                                                                                    | 1439 |
| Table 38-1  | MII Interface Signal Description.....                                                                           | 1505 |
| Table 38-2  | Signal State Description When Sending Data.....                                                                 | 1506 |
| Table 38-3  | Signal State Description when Receiving Data.....                                                               | 1506 |
| Table 38-4  | SMI Interface Signal Description.....                                                                           | 1508 |
| Table 38-5  | SMI Frame Format.....                                                                                           | 1508 |
| Table 38-6  | ETHMAC Port Function Allocation.....                                                                            | 1508 |
| Table 38-7  | Timestamp Snapshot Target Frames.....                                                                           | 1528 |
| Table 38-8  | Normal Tx Descriptor.....                                                                                       | 1538 |
| Table 38-13 | Enhanced Tx Descriptor.....                                                                                     | 1545 |
| Table 38-16 | Normal Rx Descriptor.....                                                                                       | 1549 |
| Table 38-21 | Enhanced Rx Descriptor.....                                                                                     | 1553 |
| Table 38-25 | ETHMAC Common Register Offset.....                                                                              | 1558 |
| Table 38-26 | Ethernet MAC Controller (ETHMAC) Register Offset.....                                                           | 1558 |
| Table 39-1  | Port Allocation Table.....                                                                                      | 1650 |
| Table 39-2  | SDIOC System Control Register Offset.....                                                                       | 1658 |
| Table 39-3  | SDIO Controller (SDIOC) Register Offset.....                                                                    | 1658 |
| Table 40-1  | Table of AHB Access Width and Memory Bit Width Corresponding Access Methods.....                                | 1688 |
| Table 40-2  | SMC Protocol Interface.....                                                                                     | 1691 |
| Table 40-3  | DMC Protocol Interface.....                                                                                     | 1692 |
| Table 40-4  | NFC Protocol Interface.....                                                                                     | 1692 |
| Table 40-5  | EXMC Port Function Mapping 1.....                                                                               | 1693 |
| Table 40-6  | EXMC Port Function Mapping 2.....                                                                               | 1693 |
| Table 40-7  | Basic Setup Example for a Single Read Action.....                                                               | 1697 |
| Table 40-8  | Basic Setup Example for a Single Read Action with Address and Data Line Multiplexing.....                       | 1704 |
| Table 40-9  | Basic Setup Example for a Single Write Action.....                                                              | 1707 |
| Table 40-10 | Basic Setup Example for a Single Write Action with Address and Data Line Multiplexing in Asynchronous Mode..... | 1713 |
| Table 40-11 | Basic Setup Example for a Single Write Action with Address and Data Line Multiplexing in Synchronous Mode.....  | 1714 |
| Table 40-12 | DMC Command Truth Table.....                                                                                    | 1717 |
| Table 40-13 | ONFI Access Commands.....                                                                                       | 1724 |

|              |                                                                          |      |
|--------------|--------------------------------------------------------------------------|------|
| Table 40-14  | EXMC Common Register Offset.....                                         | 1727 |
| Table 40-15  | External Memory Controller (EXMC) SMC Register Offset.....               | 1727 |
| Table 40-16  | External Memory Controller (EXMC) DMC Register Offset.....               | 1727 |
| Table 40-17  | External Memory Controller (EXMC) NFC Register Offset.....               | 1728 |
| Table 40-34  | CS0 Setting and Access Address Correspondence Table.....                 | 1735 |
| Table 40-55  | CS0 Setting and Access Address Correspondence Table.....                 | 1745 |
| Table 40-94  | Command parameter list.....                                              | 1759 |
| Table 40-99  | Index Register Value and MEM Address Correspondence Table.....           | 1762 |
| Table 40-116 | ECCR Register List in 4bitECC Mode.....                                  | 1770 |
| Table 40-117 | Each Segment ECCR Register in 4bitECC Mode.....                          | 1771 |
| Table 41-1   | DVP Basic Functions and Features.....                                    | 1776 |
| Table 41-2   | DVP Port List.....                                                       | 1777 |
| Table 41-3   | Monochrome Format Video Data Storage Method.....                         | 1778 |
| Table 41-4   | YCbCr Format Video Data Storage Method.....                              | 1778 |
| Table 41-5   | RGB565 Format Video Data Storage Method.....                             | 1778 |
| Table 41-6   | 8-bit Data DVP Storage Method.....                                       | 1779 |
| Table 41-7   | 10-bit Data DVP Storage Method.....                                      | 1779 |
| Table 41-8   | 12-bit Data DVP Storage Method.....                                      | 1780 |
| Table 41-9   | 14-bit Data DVP Storage Method.....                                      | 1780 |
| Table 41-10  | Frame Transfer Interrupt.....                                            | 1784 |
| Table 41-11  | Synchronization Code Error Sequence.....                                 | 1784 |
| Table 41-12  | Digital Video Port (DVP) Register Offset.....                            | 1786 |
| Table 42-2   | Operation Speed.....                                                     | 1802 |
| Table 42-3   | RAM Operand Mapping .....                                                | 1805 |
| Table 42-4   | Programming Entry Address.....                                           | 1805 |
| Table 42-5   | AES Hardware Operation Period Count (Grouping size: 128-bit).....        | 1807 |
| Table 42-6   | SM4 Hardware Operation Period Number (Grouping size: 128-bit).....       | 1807 |
| Table 42-7   | DES Hardware Operation Period Count (Grouping size: 64-bit).....         | 1807 |
| Table 42-12  | 128-bit AES ECB Mode Operation Register Example.....                     | 1810 |
| Table 42-13  | Secure Hash Algorithm (HASH) Register Offset.....                        | 1811 |
| Table 42-14  | True Random Number Generator (TRNG) Register Offset.....                 | 1811 |
| Table 42-15  | Public Key Engine (PKE) Register Offset.....                             | 1811 |
| Table 42-16  | Symmetric Encryption Algorithm Module (SKE) Register Offset.....         | 1812 |
| Table 43-1   | CRC Parameter Model.....                                                 | 1850 |
| Table 43-2   | CRC Operation (CRC) Register Offset.....                                 | 1854 |
| Table 44-1   | Data Computing Unit (DCU) Register Offset.....                           | 1865 |
| Table 45-1   | Math Unit (MAU) Register Offset.....                                     | 1885 |
| Table 46-1   | Filter Math Accelerator (FMAC) Register Offset.....                      | 1892 |
| Table 47-1   | ERMU Parameter Definition Table.....                                     | 1899 |
| Table 47-2   | ERMU Input/Output Ports.....                                             | 1899 |
| Table 47-3   | Error Output Operation.....                                              | 1900 |
| Table 47-4   | Description of ERMU_EOUT0M Output Situation.....                         | 1900 |
| Table 47-5   | ERMU_EOUTyM (y=1-3) and ERMU_EOUTyC (y=0-3) Output Port Description..... | 1901 |
| Table 47-7   | ERMU Error Source List.....                                              | 1903 |
| Table 47-8   | Error Management Unit (ERMU) Register Offset.....                        | 1907 |
| Table 48-1   | SWJ Debug Port Pin.....                                                  | 1921 |
| Table 48-2   | Flexible SWJ-DP Pin Assignment.....                                      | 1922 |
| Table 48-6   | Debug Controller (DBGC) Register Offset.....                             | 1926 |

## 1 Product Introduction

HC32A4A8 Series is a high-performance MCU with maximum operating 240MHz frequency based on 32-bit ARM® Cortex®-M4 Microcontroller. The Cortex-M4 core integrates an FPU and DSP for single-precision floating-point operations, supporting all ARM single-precision data processing instructions, data types, and full DSP instruction sets. The core integrates MPU and DMAC-dedicated MPU for system security.

HC32A4A8 Series integrates high-speed on-chip memory, including up to 2MB Flash and 512KB SRAM, both with ECC support. It integrates a Flash access acceleration unit to achieve single-cycle program execution by the CPU on Flash. A polling bus matrix allows multiple bus masters to access memory and peripherals simultaneously, enhancing performance. Bus masters include CPU, DMA, USB-dedicated DMA, and ETHMAC-dedicated DMA. In addition to the bus matrix, it supports peripheral-to-peripheral data transfer, basic arithmetic operations, and event triggering, significantly reducing CPU workload.

HC32A4A8 Series integrates rich peripheral functions. Includes 3 independent 12-bit 2.5Msps ADCs, 4 programmable gain amplifiers (PGA), 4 12-bit DACs, 4 high-speed comparators (CMP), 8 multifunction PWM Timers (Timer6) supporting 16-channel complementary PWM output, 3 motor PWM Timers (Timer4) supporting 18-channel complementary PWM output, 12 16-bit general-purpose Timers (TimerA) with 6-channel 3-phase quadrature encoder input and 48-channel Duty independently configurable PWM outputs, 22 serial interfaces (I2C/USART/SPI), 1 QSPI, 4 CAN, 4 I2S with audio PLL, 2 SDIOs, 1 ETHMAC, USBFS Controller with built-in PHY and USBHS Controller, 1 external bus controller (NFC, SMC, DMC), 1 DVP, 1 MAU, 4 FMACs, 1 SKE, 1 PKE, 1 ERMU, 1 HSM Light-Mid.

HC32A4A8 Series supports wide voltage (1.8-3.63V), wide temperature (-40-125°C), and multiple low-power modes. Supports fast wake-up from low-power modes: 2μs from STOP mode, 25μs from Power-down mode.

### Typical Application

HC32A4A8 Series is available in 100-pin, 144-pin, and 176-pin LQFP package, designed for automotive electronics (e.g., T-Box, BMS controllers, gateways, vehicle lighting control).

### About This Manual

This manual describes the chip's functionalities, operational guidelines, and usage methods. For detailed chip specifications, refer to the corresponding "Datasheet".

## 2 Memory Mapping

### 2.1 Memory Mapping

This MCU supports a 4GB linear address space, with addresses from 0x00000000 to 0xFFFFFFFF, including programs and data. Please refer to the table below for detailed memory mapping.

Table 2-1 Memory Mapping

| Memory Classification <sup>(2)</sup> |                                 | Start Address <sup>(1)</sup> | End Address <sup>(1)</sup> | Space Size | Module <sup>(3)</sup> | Protection <sup>(4)</sup> | Description                         |
|--------------------------------------|---------------------------------|------------------------------|----------------------------|------------|-----------------------|---------------------------|-------------------------------------|
| System                               | -                               | 0xE0100000                   | 0xFFFFFFFF                 | 511MB      | Reserved              | -                         | Custom space                        |
|                                      | Private Peripheral External Bus | 0xE00FF000                   | 0xE00FFFFF                 | 4KB        | ROMTABLE              | -                         | -                                   |
|                                      |                                 | 0xE0042400                   | 0xE00FEFFF                 | 755KB      | BLANK                 | -                         | -                                   |
|                                      |                                 | 0xE0042000                   | 0xE00423FF                 | 1KB        | DBGC                  | -                         | Debug control register area         |
|                                      |                                 | 0xE0041000                   | 0xE0041FFF                 | 4KB        | ETM                   | -                         | -                                   |
|                                      |                                 | 0xE0040000                   | 0xE0040FFF                 | 4KB        | TPIU                  | -                         | -                                   |
|                                      | Private Peripheral Internal Bus | 0xE000F000                   | 0xE003FFFF                 | 196KB      | BLANK                 | -                         | -                                   |
|                                      |                                 | 0xE000E000                   | 0xE000EFFF                 | 4KB        | SCS                   | -                         | System control space NVIC/MPU, etc. |
|                                      |                                 | 0xE0003000                   | 0xE000DFFF                 | 44KB       | BLANK                 | -                         | -                                   |
|                                      |                                 | 0xE0002000                   | 0xE0002FFF                 | 4KB        | FPB                   | -                         | -                                   |
|                                      |                                 | 0xE0001000                   | 0xE0001FFF                 | 4KB        | DWT                   | -                         | -                                   |
|                                      |                                 | 0xE0000000                   | 0xE0000FFF                 | 4KB        | ITM                   | -                         | -                                   |
| External Device                      | -                               | 0xA0000000                   | 0xDFFFFFFF                 | 1024MB     | Reserved              | -                         | -                                   |
| External Storage                     | AHB5 Clock: HCLK                | 0x98000000                   | 0x9FFFFFFF                 | 128MB      | QSPI                  | -                         | -                                   |
|                                      |                                 | 0x88200000                   | 0x97FFFFFFF                | 254MB      | Reserved              | -                         | -                                   |
|                                      | AHB5 Clock: BCLK                | 0x88100000                   | 0x881FFFFF                 | 1MB        | NFC                   | -                         | -                                   |
|                                      |                                 | 0x88000800                   | 0x880FFFFF                 | 1022KB     | BLANK                 | -                         | -                                   |
|                                      |                                 | 0x88000400                   | 0x880007FF                 | 1KB        | DMCR                  | -                         | -                                   |
|                                      |                                 | 0x88000000                   | 0x880003FF                 | 1KB        | SMCR                  | -                         | -                                   |
|                                      |                                 | 0x80000000                   | 0x87FFFFFFF                | 128MB      | DMC                   | -                         | -                                   |
|                                      |                                 | 0x60000000                   | 0x7FFFFFFF                 | 512MB      | SMC                   | -                         | -                                   |
| Peripheral                           | -                               | 0x44000000                   | 0x5FFFFFFF                 | 448MB      | Reserved              | -                         | -                                   |
|                                      |                                 | 0x42000000                   | 0x43FFFFFFF                | 32MB       | PeriBitBand           | -                         | -                                   |
|                                      |                                 | 0x40100000                   | 0x41FFFFFFF                | 31MB       | Reserved              | -                         | -                                   |
|                                      | AHB4 Clock: PCLK1               | 0x400C0000                   | 0x400FFFFF                 | 256KB      | USBHS                 | -                         | -                                   |
|                                      | AHB2 Clock: PCLK1               | 0x40080000                   | 0x400BFFFF                 | 256KB      | USBFS                 | -                         | -                                   |
|                                      |                                 | 0x40078800                   | 0x4007FFFF                 | 30KB       | BLANK                 | -                         | -                                   |
|                                      |                                 | 0x40078400                   | 0x400787FF                 | 1KB        | SDIOC_2               | -                         | -                                   |
|                                      |                                 | 0x40078000                   | 0x400783FF                 | 1KB        | CAN_2                 | -                         | -                                   |
|                                      | AHB3 Clock: PCLK1               | 0x40070400                   | 0x40077FFF                 | 31KB       | BLANK                 | -                         | -                                   |
|                                      |                                 | 0x40070000                   | 0x400703FF                 | 1KB        | SDIOC_1               | -                         | -                                   |
|                                      |                                 | 0x40060000                   | 0x4006FFFF                 | 64KB       | ETHMAC                | -                         | -                                   |
|                                      | AHB1 Clock: HCLK                | 0x40059000                   | 0x4005FFFF                 | 28KB       | BLANK                 | -                         | -                                   |
|                                      |                                 | 0x40058C00                   | 0x40058FFF                 | 1KB        | FMAC_4                | -                         | -                                   |



| Memory Classification <sup>(2)</sup> |                      | Start Address <sup>(1)</sup> | End Address <sup>(1)</sup> | Space Size | Module <sup>(3)</sup> | Protection <sup>(4)</sup> | Description                       |
|--------------------------------------|----------------------|------------------------------|----------------------------|------------|-----------------------|---------------------------|-----------------------------------|
| Peripheral                           | AHB1<br>Clock: HCLK  | 0x40058800                   | 0x40058BFF                 | 1KB        | FMAC_3                | -                         | -                                 |
|                                      |                      | 0x40058400                   | 0x400587FF                 | 1KB        | FMAC_2                | -                         | -                                 |
|                                      |                      | 0x40058000                   | 0x400583FF                 | 1KB        | FMAC_1                | -                         | -                                 |
|                                      |                      | 0x40057C00                   | 0x40057FFF                 | 1KB        | DCU_8                 | -                         | -                                 |
|                                      |                      | 0x40057800                   | 0x40057BFF                 | 1KB        | DCU_7                 | -                         | -                                 |
|                                      |                      | 0x40057400                   | 0x400577FF                 | 1KB        | DCU_6                 | -                         | -                                 |
|                                      |                      | 0x40057000                   | 0x400573FF                 | 1KB        | DCU_5                 | -                         | -                                 |
|                                      |                      | 0x40056C00                   | 0x40056FFF                 | 1KB        | DCU_4                 | -                         | -                                 |
|                                      |                      | 0x40056800                   | 0x40056BFF                 | 1KB        | DCU_3                 | -                         | -                                 |
|                                      |                      | 0x40056400                   | 0x400567FF                 | 1KB        | DCU_2                 | -                         | -                                 |
|                                      |                      | 0x40056000                   | 0x400563FF                 | 1KB        | DCU_1                 | -                         | -                                 |
|                                      |                      | 0x40055800                   | 0x40055BFF                 | 1KB        | BLANK                 | -                         | -                                 |
|                                      |                      | 0x40055800                   | 0x40055BFF                 | 1KB        | DVP                   | -                         | -                                 |
|                                      |                      | 0x40055400                   | 0x400557FF                 | 1KB        | PERIC                 | -                         | Peripheral block control register |
|                                      |                      | 0x40055000                   | 0x400553FF                 | 1KB        | MAU                   | -                         | -                                 |
|                                      |                      | 0x40054400                   | 0x40054FFF                 | 3KB        | BLANK                 | -                         | -                                 |
|                                      |                      | 0x40054000                   | 0x400543FF                 | 1KB        | CMU                   | With protection           | -                                 |
|                                      |                      | 0x40053800                   | 0x40053FFF                 | 2KB        | GPIO                  | -                         | -                                 |
|                                      |                      | 0x40053400                   | 0x400537FF                 | 1KB        | DMA_2                 | -                         | -                                 |
|                                      |                      | 0x40053000                   | 0x400533FF                 | 1KB        | DMA_1                 | -                         | -                                 |
|                                      |                      | 0x40052000                   | 0x40052FFF                 | 4KB        | BLANK                 | -                         | -                                 |
|                                      |                      | 0x40051000                   | 0x40051FFF                 | 4KB        | INTC                  | With protection           | -                                 |
|                                      |                      | 0x40050C00                   | 0x40050FFF                 | 1KB        | KEYSCAN               | -                         | -                                 |
|                                      |                      | 0x40050800                   | 0x40050BFF                 | 1KB        | RAMIF                 | With protection           | -                                 |
|                                      |                      | 0x40050400                   | 0x400507FF                 | 1KB        | BLANK                 | -                         | -                                 |
|                                      |                      | 0x40050000                   | 0x400503FF                 | 1KB        | MPU                   | With protection           | -                                 |
|                                      | APB4<br>Clock: PCLK3 | 0x4004F800                   | 0x4004FFFF                 | 2KB        | BLANK                 | -                         | -                                 |
|                                      |                      | 0x4004F400                   | 0x4004F7FF                 | 1KB        | I2C_6                 | -                         | -                                 |
|                                      |                      | 0x4004F000                   | 0x4004F3FF                 | 1KB        | I2C_5                 | -                         | -                                 |
|                                      |                      | 0x4004EC00                   | 0x4004EFFF                 | 1KB        | I2C_4                 | -                         | -                                 |
|                                      |                      | 0x4004E800                   | 0x4004EBFF                 | 1KB        | I2C_3                 | -                         | -                                 |
|                                      |                      | 0x4004E400                   | 0x4004E7FF                 | 1KB        | I2C_2                 | -                         | -                                 |
|                                      |                      | 0x4004E000                   | 0x4004E3FF                 | 1KB        | I2C_1                 | -                         | -                                 |
|                                      |                      | 0x4004D000                   | 0x4004DFFF                 | 1KB        | BLANK                 | -                         | -                                 |
|                                      |                      | 0x4004CC00                   | 0x4004CFFF                 | 1KB        | PWC, CMU              | With protection           | -                                 |
|                                      |                      | 0x4004C800                   | 0x4004CBFF                 | 1KB        | BLANK                 | -                         | -                                 |
|                                      |                      | 0x4004C400                   | 0x4004C7FF                 | 1KB        | VBAT                  | With protection           | -                                 |
|                                      |                      | 0x4004C000                   | 0x4004C3FF                 | 1KB        | RTC                   | With protection           | -                                 |
|                                      |                      | 0x4004AC00                   | 0x4004BFFF                 | 5KB        | BLANK                 | -                         | -                                 |
|                                      |                      | 0x4004A800                   | 0x4004ABFF                 | 1KB        | OTS                   | -                         | -                                 |

| Memory Classification <sup>(2)</sup> |                      | Start Address <sup>(1)</sup> | End Address <sup>(1)</sup> | Space Size | Module <sup>(3)</sup> | Protection <sup>(4)</sup> | Description |
|--------------------------------------|----------------------|------------------------------|----------------------------|------------|-----------------------|---------------------------|-------------|
| Peripheral                           | APB4<br>Clock: PCLK3 | 0x4004A400                   | 0x4004A7FF                 | 1KB        | CMP_3,4               | -                         | -           |
|                                      |                      | 0x4004A000                   | 0x4004A3FF                 | 1KB        | CMP_1,2               | -                         | -           |
|                                      |                      | 0x40049C00                   | 0x40049FFF                 | 1KB        | CTC                   | -                         | -           |
|                                      |                      | 0x40049800                   | 0x40049BFF                 | 1KB        | BLANK                 | -                         | -           |
|                                      |                      | 0x40049400                   | 0x400497FF                 | 1KB        | SWDT                  | With protection           | -           |
|                                      |                      | 0x40049000                   | 0x400493FF                 | 1KB        | WDT                   | With protection           | -           |
|                                      |                      | 0x400488000                  | 0x40048FFF                 | 2KB        | ERMU                  | With protection           | -           |
|                                      |                      | 0x40048400                   | 0x400487FF                 | 1KB        | FCM                   | -                         | -           |
|                                      |                      | 0x40048000                   | 0x400483FF                 | 1KB        | FCG, CMU              | With protection           | -           |
|                                      | APB3<br>Clock: PCLK4 | 0x40042400                   | 0x40047FFF                 | 23KB       | BLANK                 | -                         | -           |
|                                      |                      | 0x40042000                   | 0x400423FF                 | 1KB        | TRNG                  | With protection           | -           |
|                                      |                      | 0x40041800                   | 0x40041FFF                 | 2KB        | BLANK                 | -                         | -           |
|                                      |                      | 0x40041400                   | 0x400417FF                 | 1KB        | DAC_2                 | -                         | -           |
|                                      |                      | 0x40041000                   | 0x400413FF                 | 1KB        | DAC_1                 | -                         | -           |
|                                      |                      | 0x40040C00                   | 0x40040FFF                 | 1KB        | BLANK                 | -                         | -           |
|                                      |                      | 0x40040800                   | 0x40040BFF                 | 1KB        | ADC_3                 | -                         | -           |
|                                      |                      | 0x40040400                   | 0x400407FF                 | 1KB        | ADC_2                 | -                         | -           |
|                                      |                      | 0x40040000                   | 0x400403FF                 | 1KB        | ADC_1                 | -                         | -           |
|                                      | APB5<br>Clock: PCLK0 | 0x4003B000                   | 0x4003FFFF                 | 20KB       | BLANK                 | -                         | -           |
|                                      |                      | 0x4003AC00                   | 0x4003AFFF                 | 1KB        | TimerA_4              | -                         | -           |
|                                      |                      | 0x4003A800                   | 0x4003ABFF                 | 1KB        | TimerA_3              | -                         | -           |
|                                      |                      | 0x4003A400                   | 0x4003A7FF                 | 1KB        | TimerA_2              | -                         | -           |
|                                      |                      | 0x4003A000                   | 0x4003A3FF                 | 1KB        | TimerA_1              | -                         | -           |
|                                      |                      | 0x40038C00                   | 0x40039FFF                 | 5KB        | BLANK                 | -                         | -           |
|                                      |                      | 0x40038800                   | 0x40038BFF                 | 1KB        | Timer4_3              | -                         | -           |
|                                      |                      | 0x40038400                   | 0x400387FF                 | 1KB        | Timer4_2              | -                         | -           |
|                                      |                      | 0x40038000                   | 0x400383FF                 | 1KB        | Timer4_1              | -                         | -           |
|                                      | APB2<br>Clock: PCLK1 | 0x4002D400                   | 0x40037FFF                 | 43KB       | BLANK                 | -                         | -           |
|                                      |                      | 0x4002D000                   | 0x4002D3FF                 | 1KB        | MCAN_RA-MECC          | -                         | -           |
|                                      |                      | 0x4002C000                   | 0x4002CFFF                 | 4KB        | MCAN_RAM              | -                         | -           |
|                                      |                      | 0x4002B800                   | 0x4002BFFF                 | 2KB        | BLANK                 | -                         | -           |
|                                      |                      | 0x4002B400                   | 0x4002B7FF                 | 1KB        | MCAN_2                | -                         | -           |
|                                      |                      | 0x4002B000                   | 0x4002B3FF                 | 1KB        | MCAN_1                | -                         | -           |
|                                      |                      | 0x40029C00                   | 0x4002AFFF                 | 5KB        | BLANK                 | -                         | -           |
|                                      |                      | 0x40029800                   | 0x40029BFF                 | 1KB        | Timer0_5              | -                         | -           |
|                                      |                      | 0x40029400                   | 0x400297FF                 | 1KB        | Timer0_4              | -                         | -           |
|                                      |                      | 0x40029000                   | 0x400293FF                 | 1KB        | Timer0_3              | -                         | -           |
|                                      |                      | 0x40028000                   | 0x40028FFF                 | 4KB        | BLANK                 | -                         | -           |
|                                      |                      | 0x40027C00                   | 0x40027FFF                 | 1KB        | TimerA_12             | -                         | -           |
|                                      |                      | 0x40027800                   | 0x40027BFF                 | 1KB        | TimerA_11             | -                         | -           |
|                                      |                      | 0x40027400                   | 0x400277FF                 | 1KB        | TimerA_10             | -                         | -           |

| Memory Classification <sup>(2)</sup> |                      | Start Address <sup>(1)</sup> | End Address <sup>(1)</sup> | Space Size | Module <sup>(3)</sup> | Protection <sup>(4)</sup> | Description                            |
|--------------------------------------|----------------------|------------------------------|----------------------------|------------|-----------------------|---------------------------|----------------------------------------|
| Peripheral                           | APB2<br>Clock: PCLK1 | 0x40027000                   | 0x400273FF                 | 1KB        | TimerA_9              | -                         | -                                      |
|                                      |                      | 0x40026C00                   | 0x40026FFF                 | 1KB        | TimerA_8              | -                         | -                                      |
|                                      |                      | 0x40026800                   | 0x40026BFF                 | 1KB        | TimerA_7              | -                         | -                                      |
|                                      |                      | 0x40026400                   | 0x400267FF                 | 1KB        | TimerA_6              | -                         | -                                      |
|                                      |                      | 0x40026000                   | 0x400263FF                 | 1KB        | TimerA_5              | -                         | -                                      |
|                                      |                      | 0x40025800                   | 0x40025FFF                 | 2KB        | BLANK                 | -                         | -                                      |
|                                      |                      | 0x40025400                   | 0x400257FF                 | 1KB        | Timer2_4              | -                         | -                                      |
|                                      |                      | 0x40025000                   | 0x400253FF                 | 1KB        | Timer2_3              | -                         | -                                      |
|                                      |                      | 0x40024C00                   | 0x40024FFF                 | 1KB        | Timer2_2              | -                         | -                                      |
|                                      |                      | 0x40024800                   | 0x40024BFF                 | 1KB        | Timer2_1              | -                         | -                                      |
|                                      |                      | 0x40024400                   | 0x400247FF                 | 1KB        | Timer0_2              | -                         | -                                      |
|                                      |                      | 0x40024000                   | 0x400243FF                 | 1KB        | Timer0_1              | -                         | -                                      |
|                                      |                      | 0x40022800                   | 0x40023FFF                 | 6KB        | BLANK                 | -                         | -                                      |
|                                      |                      | 0x40022400                   | 0x400227FF                 | 1KB        | I2S_4                 | -                         | -                                      |
|                                      |                      | 0x40022000                   | 0x400223FF                 | 1KB        | I2S_3                 | -                         | -                                      |
|                                      |                      | 0x40021C00                   | 0x40021FFF                 | 1KB        | USART_10              | -                         | -                                      |
|                                      |                      | 0x40021800                   | 0x40021BFF                 | 1KB        | USART_9               | -                         | -                                      |
|                                      |                      | 0x40021400                   | 0x400217FF                 | 1KB        | USART_8               | -                         | -                                      |
|                                      |                      | 0x40021000                   | 0x400213FF                 | 1KB        | USART_7               | -                         | -                                      |
|                                      |                      | 0x40020C00                   | 0x40020FFF                 | 1KB        | USART_6               | -                         | -                                      |
|                                      |                      | 0x40020800                   | 0x40020BFF                 | 1KB        | SPI_6                 | -                         | -                                      |
|                                      |                      | 0x40020400                   | 0x400207FF                 | 1KB        | SPI_5                 | -                         | -                                      |
|                                      |                      | 0x40020000                   | 0x400203FF                 | 1KB        | SPI_4                 | -                         | -                                      |
|                                      | APB1<br>Clock: PCLK1 | 0x4001E800                   | 0x4001FFFF                 | 6KB        | BLANK                 | -                         | -                                      |
|                                      |                      | 0x4001E400                   | 0x4001E7FF                 | 1KB        | I2S_2                 | -                         | -                                      |
|                                      |                      | 0x4001E000                   | 0x4001E3FF                 | 1KB        | I2S_1                 | -                         | -                                      |
|                                      |                      | 0x4001DC00                   | 0x4001DFFF                 | 1KB        | USART_5               | -                         | -                                      |
|                                      |                      | 0x4001D800                   | 0x4001DBFF                 | 1KB        | USART_4               | -                         | -                                      |
|                                      |                      | 0x4001D400                   | 0x4001D7FF                 | 1KB        | USART_3               | -                         | -                                      |
|                                      |                      | 0x4001D000                   | 0x4001D3FF                 | 1KB        | USART_2               | -                         | -                                      |
|                                      |                      | 0x4001CC00                   | 0x4001CFFF                 | 1KB        | USART_1               | -                         | -                                      |
|                                      |                      | 0x4001C800                   | 0x4001CBFF                 | 1KB        | SPI_3                 | -                         | -                                      |
|                                      |                      | 0x4001C400                   | 0x4001C7FF                 | 1KB        | SPI_2                 | -                         | -                                      |
|                                      |                      | 0x4001C000                   | 0x4001C3FF                 | 1KB        | SPI_1                 | -                         | -                                      |
|                                      |                      | 0x40018000                   | 0x4001BFFF                 | 16KB       | Timer6                | -                         | Count clock: PCLK0                     |
|                                      |                      | 0x40017C00                   | 0x40017FFF                 | 1KB        | EMB                   | -                         | -                                      |
|                                      |                      | 0x40010C00                   | 0x40017BFF                 | 28KB       | BLANK                 | -                         | -                                      |
|                                      |                      | 0x40010800                   | 0x40010BFF                 | 1KB        | AOS                   | -                         | Internal trigger event register region |
|                                      |                      | 0x40010400                   | 0x400107FF                 | 1KB        | EFM                   | With protection           | -                                      |
|                                      |                      | 0x40010000                   | 0x400103FF                 | 1KB        | BLANK                 | -                         | -                                      |
|                                      | AHB4<br>Clock: PCLK1 | 0x4000F000                   | 0x4000FFFF                 | 4KB        | BLANK                 | -                         | -                                      |
|                                      |                      | 0x4000E000                   | 0x4000EFFF                 | 4KB        | SKE                   | With protection           | -                                      |

| Memory Classification <sup>(2)</sup> |                           | Start Address <sup>(1)</sup> | End Address <sup>(1)</sup> | Space Size | Module <sup>(3)</sup>    | Protection <sup>(4)</sup> | Description                                        |
|--------------------------------------|---------------------------|------------------------------|----------------------------|------------|--------------------------|---------------------------|----------------------------------------------------|
| Peripheral                           | AHB4<br>Clock: PCLK1      | 0x4000C000                   | 0x4000DFFF                 | 8KB        | PKE                      | With protection           | -                                                  |
|                                      |                           | 0x4000B000                   | 0x4000BFFF                 | 4KB        | PKE_RAM                  | -                         | -                                                  |
|                                      |                           | 0x40009400                   | 0x4000AFFF                 | 7KB        | BLANK                    | -                         | -                                                  |
|                                      |                           | 0x40009000                   | 0x400093FF                 | 1KB        | CAN_1                    | -                         | -                                                  |
|                                      |                           | 0x40008C00                   | 0x40008FFF                 | 1KB        | CRC                      | With protection           | -                                                  |
|                                      |                           | 0x40008800                   | 0x40008BFF                 | 1KB        | BLANK                    | -                         | -                                                  |
|                                      |                           | 0x40008400                   | 0x400087FF                 | 1KB        | HASH                     | With protection           | -                                                  |
|                                      |                           | 0x40008000                   | 0x400083FF                 | 1KB        | BLANK                    | -                         | -                                                  |
|                                      |                           | 0x40000000                   | 0x40007FFF                 | 32KB       | Reserved                 | -                         | -                                                  |
| SRAM                                 | SRAM<br>Clock: HCLK       | 0x24000000                   | 0x3FFFFFFF                 | 448MB      | Reserved                 | -                         | -                                                  |
|                                      |                           | 0x22000000                   | 0x23FFFFFF                 | 32MB       | SRAMBit-Band             | -                         | -                                                  |
|                                      |                           | 0x20100000                   | 0x21FFFFFF                 | 31MB       | Reserved                 | -                         | -                                                  |
|                                      |                           | 0x200F1000                   | 0x200FFFFF                 | 60KB       | BLANK                    | -                         | -                                                  |
|                                      |                           | 0x200F0000                   | 0x200F0FFF                 | 4KB        | RetSRAM                  | -                         | ECC RAM                                            |
|                                      |                           | 0x20060000                   | 0x200EFFFF                 | 576KB      | BLANK                    | -                         | -                                                  |
|                                      |                           | 0x20058000                   | 0x2005FFFF                 | 32KB       | SRAM4                    | -                         | ECC RAM                                            |
|                                      |                           | 0x20040000                   | 0x20057FFF                 | 96KB       | SRAM3                    | -                         | ECC RAM                                            |
|                                      |                           | 0x20020000                   | 0x2003FFFF                 | 128KB      | SRAM2                    | -                         | ECC RAM                                            |
| CODE                                 | SRAM<br>Clock: HCLK       | 0x1FFE0000                   | 0x1FFFFFFF                 | 128KB      | SRAMH                    | -                         | ECC RAM                                            |
|                                      |                           | 0x03004000                   | 0x1FFDFFFF                 | 463.86 MB  | BLANK                    | -                         | -                                                  |
|                                      | OTP, Flash<br>Clock: HCLK | 0x0300400C                   | 0x03005FFF                 | 8180B      | BLANK                    | -                         | -                                                  |
|                                      |                           | 0x03004000                   | 0x0300400B                 | 12B        | Data security protection | -                         | Used for configuration of data security protection |
|                                      |                           | 0x03002004                   | 0x03003FFF                 | 8187B      | BLANK                    | -                         | -                                                  |
|                                      |                           | 0x03002000                   | 0x03002003                 | 4B         | Boot Swap                | -                         | Only access by CPU                                 |
|                                      |                           | 0x03001FF0                   | 0x03001FFF                 | 16B        | OTP                      | -                         | Only access by CPU                                 |
|                                      |                           | 0x03001F70                   | 0x03001FEF                 | 128B       | BLANK                    | -                         | -                                                  |
|                                      |                           | 0x03000000                   | 0x03001F6F                 | 8048B      | OTP                      | -                         | Only access by CPU                                 |
|                                      | -                         | 0x02100000                   | 0x02FFFFFF                 | 15M        | BLANK                    | -                         | -                                                  |
|                                      | REMAP<br>Clock: HCLK      | 0x02080000                   | 0x020FFFFF                 | 512KB      | REMAP1                   | -                         | Address remapping region 1                         |
|                                      |                           | 0x02000000                   | 0x0207FFFF                 | 512KB      | REMAP0                   | -                         | Address remapping region 0                         |
|                                      | -                         | 0x00200000                   | 0x01FFFFFF                 | 30M        | BLANK                    | -                         | -                                                  |
|                                      | Flash<br>Clock: HCLK      | 0x00100000                   | 0x001FFFFF                 | 1MB        | Embedded-Flash1          | -                         | -                                                  |
|                                      |                           | 0x00000000                   | 0x000FFFFF                 | 1MB        | Embedded-Flash0          | -                         | -                                                  |

**Note**

1. Please refer to the "ARM Cortex-M4 Processor Technical Reference Manual Revision: r0p1".
2. For bus descriptions, please refer to the relevant descriptions in the "Bus Architecture (BUS) " chapter.
3. Reserved: Accessing the bus will cause a bus error.  
BLANK: Write access has no effect; reading returns 0.
4. Modules with protection functions only support access in CPU privileged mode when the protection function is active. For specific registers and descriptions, refer to the relevant content in the "Memory Protection Unit (MPU)" chapter.

## 2.2 External Space Mapping

The QSPI space is divided into two segments, including 64MB of QSPI register space and 64MB of external QSPI device space. Please refer to the figure below for the allocation relationship.

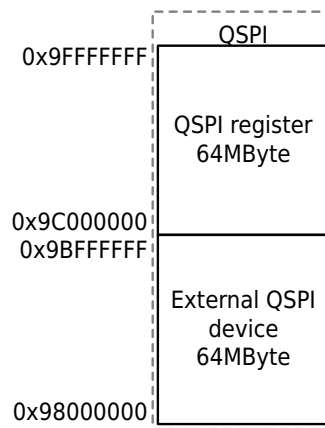


Figure 2-1 QSPI Address Space Allocation

## 2.3 Bit Segment Space

Cortex-M4 memory mapping includes two bit segment regions. These regions map each word in the memory alias region to the corresponding bit in the memory bit segment region. When you write to an alias region, it is equivalent to a read-modify-write operation on the target bit of the bit band region.

In this MCU, peripheral registers and SRAM are mapped to a bit segment region, enabling read and write operations on individual bit segments. These operations are only applicable to Cortex-M4 access and are disabled for other bus master interfaces (e.g., DMA).

## 2.4 Address Remapping

This MCU provides a memory address remapping function with two remapping addresses available for free configuration, using MMF\_REMCR0 and MMF\_REMCR1 to configure the remapping target address, which can be either the main flash address or the SRAMH address.

- Remapping Address 0:  
0x02000000-0x02080000 (depending on the set remapping space size through MMF\_REMCR0.RMSIZE[4:0] )
- Remapping Address 1:  
0x02080000-0x02100000 (depending on the set remapping space size through MMF\_REMCR1.RMSIZE[4:0])

When the remapping function is enabled (MMF\_REMCR0.EN=1 or MMF\_REMCR1.EN=1), the address correspondence table is as follows:

**Table 2-2 Target Address Configuration Main Flash Example (MMF\_REMCR0.EN=1 or MMF\_REMCR1.EN=1)**

| Register Setting                                       | Remapping Address (CPU Address-CPUADDR[31:0]) | Remapping Target Address (Main Flash Address) |                                       |               |
|--------------------------------------------------------|-----------------------------------------------|-----------------------------------------------|---------------------------------------|---------------|
|                                                        |                                               | Upper 12 Bits                                 | Middle 6 Bits                         | Lower 14 Bits |
| REMCR0.RMSIZE[4:0]=0b01110<br>(Remapping Space: 16KB)  | 0x02000000-0x02003FFF                         | All 0                                         | REMCR0.RMTADDR[5:0]                   | CPUADDR[13:0] |
| REMCR0.RMSIZE[4:0]=0b01111<br>(Remapping Space: 32KB)  | 0x02000000-0x02007FFF                         | All 0                                         | REMCR0.RMTADDR[4:0]<br>CPUADDR[14]    | CPUADDR[13:0] |
| REMCR1.RMSIZE[4:0]=0b10000<br>(Remapping Space: 64KB)  | 0x02080000-0x02087FFF                         | All 0                                         | REMCR1.RMTADDR[3:0]<br>CPUADDR[15:14] | CPUADDR[13:0] |
| REMCR1.RMSIZE[4:0]=0b10001<br>(Remapping Space: 128KB) | 0x02080000-0x0208FFFF                         | All 0                                         | REMCR1.RMTADDR[2:0]<br>CPUADDR[16:14] | CPUADDR[13:0] |

**Table 2-3 Target Address Configuration SRAMH Example (MMF\_REMCR0.EN=1 or MMF\_REMCR1.EN=1)**

| Register Setting                                     | Remapping Address (CPU Address-CPUADDR[31:0]) | Remapping Target Address (SRAM Address) |                                     |               |
|------------------------------------------------------|-----------------------------------------------|-----------------------------------------|-------------------------------------|---------------|
|                                                      |                                               | Upper 3 Bits                            | Middle 17 Bits                      | Lower 12 Bits |
| REMCR0.RMSIZE[4:0]=0b01100<br>(Remapping Space: 4KB) | 0x02000000-0x02000FFF                         | All 0                                   | REMCR0.RMTADDR[16:0]                | CPUADDR[11:0] |
| REMCR0.RMSIZE[4:0]=0b01101<br>(Remapping Space: 8KB) | 0x02000000-0x02001FFF                         | All 0                                   | REMCR0.RMTADDR[15:0]<br>CPUADDR[12] | CPUADDR[11:0] |
| REMCR1.RMSIZE[4:0]=0b01100<br>(Remapping Space: 4KB) | 0x02080000-0x02080FFF                         | All 0                                   | REMCR1.RMTADDR[16:0]                | CPUADDR[11:0] |
| REMCR1.RMSIZE[4:0]=0b01101<br>(Remapping Space: 8KB) | 0x02080000-0x02081FFF                         | All 0                                   | REMCR1.RMTADDR[15:0]<br>CPUADDR[12] | CPUADDR[11:0] |

Example:

- Using the remapping address 0 function, set the source address to the main flash address 0x00008000, with a remapping space of 32K; the register MMF\_REMCR0 needs to be set to 0x8000800F.
- Using the remapping address 1 function, set the source address to SRAMH address 0x1FFFE000, with a remapping space of 16K; the register MMF\_REMCR1 needs to be set to 0x9FFFE00E.



#### Notice

The start address of the source address should be set to an integer multiple of the remapping space.

## 2.5 Register Description

### 2.5.1 Register List

Base address: 0x40010500

Table 2-4 Memory Mapping Register Offset

| Offset       | Register   | Description                  | Width |
|--------------|------------|------------------------------|-------|
| 0x0000       | MMF_REMPRT | Access Protection Register   | 32    |
| 0x0004+0x4*x | MMF_REMCrx | Remapping Register x (x=0-1) | 32    |



## 2.5.2 Access Protection Register (MMF\_REMPRT)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |            |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|------------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0x0000 | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15         | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |            |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | MMF_REMPRT |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RW         |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit   | Name       | Function            | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | Access |
|-------|------------|---------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:16 | Reserved   | Reserved            | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | -      |
| 15:0  | MMF_REMPRT | Protection register | <p>Write protection for registers MMF_REMCRO and MMF_REMCRI: First write 0x0123, then write 0x3210 to MMF_REMPRT[15:0] to disable protection.</p> <div>  <b>Note</b> <ul style="list-style-type: none"> <li>When registers MMF_REMCRO and MMF_REMCRI are in write protection state, reading the register returns 0.</li> <li>When registers MMF_REMCRO and MMF_REMCRI are not in write protection state, reading the register returns 1.</li> </ul> </div> | RW     |

### 2.5.3 Remapping Register (MMF\_REMCrX) (x=0-1)

Offset: 0x0004+0x4\*x

| Offset | Bit Position |          |         |    |    |    |    |    |    |    |    |    |    |    |    |    |          |    |    |    |    |    |   |   |        |   |   |   |   |   |   |   |
|--------|--------------|----------|---------|----|----|----|----|----|----|----|----|----|----|----|----|----|----------|----|----|----|----|----|---|---|--------|---|---|---|---|---|---|---|
| '      | 31           | 30       | 29      | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15       | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7      | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |          |         |    |    |    |    |    |    |    |    |    |    |    |    |    |          |    |    |    |    |    |   |   |        |   |   |   |   |   |   |   |
| Name   | EN           | Reserved | RMTADDR |    |    |    |    |    |    |    |    |    |    |    |    |    | Reserved |    |    |    |    |    |   |   | RMSIZE |   |   |   |   |   |   |   |
| Access | RW           |          | RW      |    |    |    |    |    |    |    |    |    |    |    |    |    |          |    |    |    |    |    |   |   | RW     |   |   |   |   |   |   |   |

| Bit   | Name     | Function                 | Description                                                                                                                                                                                                                                                                                                  | Access |
|-------|----------|--------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31    | EN       | Remapping Enable Bit     | <ul style="list-style-type: none"> <li>0b0: Re-mapping disabled</li> <li>0b1: Re-mapping enabled</li> </ul>                                                                                                                                                                                                  | RW     |
| 30:29 | Reserved | Reserved                 | -                                                                                                                                                                                                                                                                                                            | -      |
| 28:12 | RMTADDR  | Remapping Target Address | The number of active bits is related to the RMSIZE[4:0] setting.<br>The setting can refer to <a href="#">Address Remapping</a> .                                                                                                                                                                             | RW     |
| 11:5  | Reserved | Reserved                 | -                                                                                                                                                                                                                                                                                                            | -      |
| 4:0   | RMSIZE   | Remapping Space Size     | Set the size of the re-mapping region: <ul style="list-style-type: none"> <li>0b01100: 4KB</li> <li>0b01101: 8KB</li> <li>0b01110: 16KB</li> <li>0b01111: 32KB</li> <li>0b10000: 64KB</li> <li>0b10001: 128KB</li> <li>0b10010: 256KB</li> <li>0b10011: 512KB</li> <li>Others: Setting prohibited</li> </ul> | RW     |

## 3 Bus Architecture (BUS)

### 3.1 Introduction

The main system consists of a 32-bit multi-layer AHB bus matrix, enabling the interconnection of the following host buses and slave buses:

- Host bus
  - ▶ Cortex-M4 core CPU-I bus, CPU-D bus, CPU-S bus
  - ▶ System DMA\_1 bus, System DMA\_2 bus
  - ▶ USBFS\_DMA bus
  - ▶ USBHS\_DMA bus
  - ▶ ETHMAC\_DMA bus
- Slave bus
  - ▶ Flash ICODE bus
  - ▶ Flash DCODE bus
  - ▶ Flash MCODE bus (The bus through which non-CPU hosts access the Flash memory)
  - ▶ High-speed SRAMH (SRAMH 128KB) bus
  - ▶ System SRAM (SRAM1 128KB) bus
  - ▶ System SRAM (SRAM2 128KB) bus
  - ▶ System SRAM (SRAM3 96KB, SRAM4 32KB, Ret SRAM 4KB) bus
  - ▶ APB1 peripheral bus (AOS/EMB/Timer6/SPI/USART/I2S/EFM)
  - ▶ APB2 peripheral bus (Timer0/Timer2/TimerA/SPI/USART/I2S/CAN/MCAN)
  - ▶ APB3 peripheral bus (ADC/DAC/TRNG)
  - ▶ APB4 peripheral bus (FCG/CMU/FCM/ERMU/WDT/SWDT/CMP/PWC/CTC/OTS/RTC/VBAT/WKTM/I2C)
  - ▶ APB5 peripheral bus (Timer4/TimerA)
  - ▶ AHB1 peripheral bus (MPU/RAMIF/KEYSCAN/INTC/DCU/GPIO/DMA/CMU/DVP/MAU/FMAC)
  - ▶ AHB2 peripheral bus (CAN/SDIOC/USBFS)
  - ▶ AHB3 peripheral bus (SDIOC/ETHMAC)
  - ▶ AHB4 peripheral bus (HASH/CRC/CAN/USBHS/PKE/SKE)
  - ▶ AHB5 peripheral bus (SMC/DMC/SMCR/DMCR/NFC/QSPI)

With the bus matrix, efficient concurrent access from host buses to slave buses can be achieved.

## 3.2 Bus Architecture

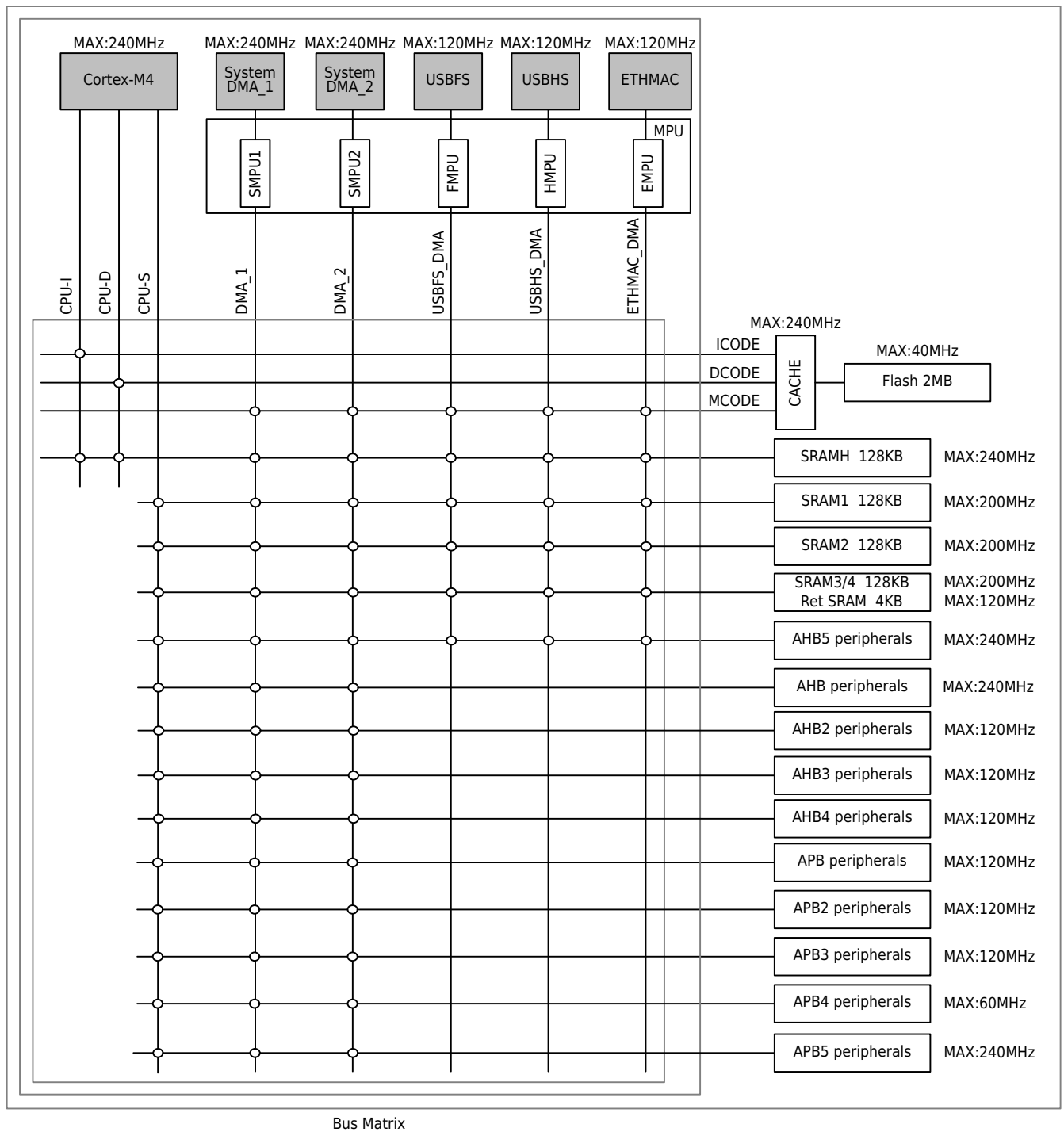


Figure 3-1 Bus Architecture Diagram

The bus matrix manages access arbitration between host buses. Arbitration uses a round-robin scheduling algorithm.

- CPU-I bus

Cortex-M4 Core Command Bus: The CPU fetches instructions through this bus. Access targets: Flash and SRAMH containing code.

- CPU-D bus  
Cortex-M4 Core Data Bus: The CPU performs immediate number loading and debug access through this bus. Access targets: Flash and SRAMH containing code or data.
- CPU-S bus  
Cortex-M4 Core System Bus: The CPU accesses peripherals or system SRAM through this bus. It can also fetch instructions and immediate numbers through this bus (though less efficient than through the CPU-I bus and CPU-D bus). Access targets: SRAM1/SRAM2/SRAM3/SRAM4/Ret SRAM, all peripherals, and external extended space.
- DMA\_1 bus, DMA\_2 bus  
System DMA\_1/system DMA\_2 dedicated bus, DMA\_1/DMA\_2 accesses data memory and peripherals through this bus. The access objects are Flash/SRAMH/SRAM1/SRAM2/SRAM3/SRAM4/Ret SRAM, all peripherals, and external extended space.
- USBFS\_DMA bus  
DMA dedicated bus for USBFS, through which USBFS accesses all memory spaces. Access targets include Flash/SRAMH/SRAM1/SRAM2/SRAM3/SRAM4/Ret SRAM, and external extended space.
- USBHS\_DMA bus  
USBHS DMA dedicated bus, allowing USBHS to access all memory spaces via this bus. Access targets include Flash/SRAMH/SRAM1/SRAM2/SRAM3/SRAM4/Ret SRAM, and external extended space.
- ETHMAC\_DMA bus  
ETHMAC DMA dedicated bus, allowing ETHMAC to access all memory spaces via this bus. Access targets include Flash/SRAMH/SRAM1/SRAM2/SRAM3/SRAM4/Ret SRAM, and external extended space.

### 3.3 Bus Function

The bus is responsible for the host's read and writes access to the slave computer. When the operating frequency of the host module is higher than that of the slave module (such as CPU-S accessing RTC), the bus will automatically perform frequency reduction synchronization processing. When the operating frequency of the host module is lower than that of the slave module (such as USBFS\_DMA accessing SRAMH), the bus automatically performs up-frequency synchronization processing.

Through the bus matrix, when the access targets of different host buses do not conflict, each access can be carried out simultaneously. For example, CPU-I accesses Flash, CPU-D accesses SRAMH, CPU-S accesses APB peripherals, DMA\_1 accesses SRAM1, DMA\_2 accesses SRAM2, USBFS\_DMA accesses the external expansion space of AHB5, and these accesses can be performed simultaneously.

## 4 Reset Module Unit (RMU)

### 4.1 Introduction

The chip is configured with 16 reset modes:

- Power-on Reset (POR)
- NRST pin Reset (NRST)
- Brown-out Reset (BOR)
- Programmable Voltage Detect 1 Reset (PVD1R)
- Programmable Voltage Detect 2 Reset (PVD2R)
- Watchdog Reset (WDTR)
- Special Watchdog Reset (SWDTR)
- Power-down Wake-up Reset (PDRST)
- Software Reset (SRST)
- MPU Error Reset (MPUR)
- RAMECC Reset (RAMECCR)
- FLASHECC Reset (FLASHECCR)
- ERMU Reset (ERMUR)
- Clock Frequency Error Reset (CKFER)
- XTAL Error Reset (XTALER)
- Cortex-M4 Lockup Reset (LKUPR)

### 4.2 Functional Description

#### 4.2.1 Reset Mode and Reset Flag Bit

The reset mode and generating conditions are shown in the following table.

**Table 4-1 Reset Mode and Generating Conditions**

| Reset Mode                             | Generating Condition                                                                                                                    |
|----------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------|
| Power-on Reset                         | VCC powered on                                                                                                                          |
| NRST Pin Reset                         | NRST pin input low level                                                                                                                |
| Brown-out Reset                        | VCC voltage drops below VBOR voltage                                                                                                    |
| Programmable Voltage Detection 1 Reset | VCC voltage drops below PVD1 voltage                                                                                                    |
| Programmable Voltage Detection 2 Reset | VCC voltage drops below PVD2 voltage                                                                                                    |
| Watchdog Reset                         | Watchdog timer generates a refresh error or an overflow error                                                                           |
| Special Watchdog Reset                 | Refresh error or overflow error occurred in special watchdog                                                                            |
| Power-down Wake-up Reset               | By setting the reset generated by the Power-down mode, the core wakes up from the reset state after the Power-down wake-up event occurs |
| Software Reset                         | Set the reset register bit (ARM register AIRCR.SYSRESETREQ bit)                                                                         |
| MPU Error Reset                        | Reset generated by MPU access error                                                                                                     |

| Reset Mode                     | Generating Condition                                                                                                                                                      |
|--------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| RAM ECC Reset                  | Reset generated when ECC error occurs in RAM                                                                                                                              |
| FLASH ECC Reset                | Reset generated when ECC error occurs in FLASH                                                                                                                            |
| ERMU Reset                     | When a reset request is issued to the ERMU                                                                                                                                |
| Clock frequency Error Reset    | When the clock frequency monitoring function (FCM) detects a clock period error                                                                                           |
| XTAL Error Reset               | Resetting of external high-speed oscillator abnormal shock                                                                                                                |
| Cortex-M4 Lockup Reset (LKUPR) | When Cortex-M4 encounters a serious exception, it will stop its PC pointer at the current address and lock it, and reset the entire chip after a few clock period delays. |

When a reset occurs, the chip will set the corresponding reset flag according to the reset method, and the reset flag is as shown. For example, pin reset occurs, pin reset flag bit PINRF is set to 1, PINRF can be reset by writing CLRF after PINRF is set.

Table 4-2 Reset Mode and Reset Flag

| Reset Flag                                                     | Reset mode     |           |                 |                           |                           |                |                        |                          |                |                 |               |                 |            |                             |                  |
|----------------------------------------------------------------|----------------|-----------|-----------------|---------------------------|---------------------------|----------------|------------------------|--------------------------|----------------|-----------------|---------------|-----------------|------------|-----------------------------|------------------|
|                                                                | Power-on Reset | Pin Reset | Brown-out Reset | Voltage Detection 1 Reset | Voltage Detection 2 Reset | Watchdog Reset | Special Watchdog Reset | Power-down Wake-up Reset | Software Reset | MPU Error Reset | RAM ECC Reset | FLASH ECC Reset | ERMU Reset | Clock Frequency Error Reset | XTAL Error Reset |
| Power-on reset flag (RMU_RSTF0.PORF)                           | ✓              | -         | -               | -                         | -                         | -              | -                      | -                        | -              | -               | -             | -               | -          | -                           | -                |
| Pin reset flag (RMU_RSTF0.PINRF)                               | ×              | ✓         | -               | -                         | -                         | -              | -                      | -                        | -              | -               | -             | -               | -          | -                           | -                |
| Brown-out reset flag (RMU_RSTF0.BORF)                          | ×              | ×         | ✓               | -                         | -                         | -              | -                      | -                        | -              | -               | -             | -               | -          | -                           | -                |
| Programmable voltage detection 1 reset flag (RMU_RSTF0.PVD1RF) | ×              | ×         | ×               | ✓                         | -                         | -              | -                      | -                        | -              | -               | -             | -               | -          | -                           | -                |
| Programmable voltage detection 2 reset flag (RMU_RSTF0.PVD2RF) | ×              | ×         | ×               | -                         | ✓                         | -              | -                      | -                        | -              | -               | -             | -               | -          | -                           | -                |
| Watchdog reset flag (RMU_RSTF0.WDRF)                           | ×              | ×         | ×               | -                         | -                         | ✓              | -                      | ×                        | -              | -               | -             | -               | -          | -                           | -                |
| Special watchdog reset flag (RMU_RSTF0.SWDRF)                  | ×              | ×         | ×               | -                         | -                         | -              | ✓                      | ×                        | -              | -               | -             | -               | -          | -                           | -                |
| Power-down wake-up reset flag (RMU_RSTF0.PDRF)                 | ×              | -         | -               | -                         | -                         | -              | -                      | ✓                        | -              | -               | -             | -               | -          | -                           | -                |
| Software reset flag (RMU_RSTF0.SWRF)                           | ×              | ×         | ×               | -                         | -                         | -              | -                      | ×                        | ✓              | -               | -             | -               | -          | -                           | -                |
| MPU error reset flag (RMU_RSTF0.MPUERF)                        | ×              | ×         | ×               | -                         | -                         | -              | -                      | ×                        | -              | ✓               | -             | -               | -          | -                           | -                |
| RAM ECC reset flag (RMU_RSTF0.RAECRF)                          | ×              | ×         | ×               | -                         | -                         | -              | -                      | ×                        | -              | -               | ✓             | -               | -          | -                           | -                |
| FLASH ECC reset flag (RMU_RSTF0.FLECRF)                        | ×              | ×         | ×               | -                         | -                         | -              | -                      | ×                        | -              | -               | -             | ✓               | -          | -                           | -                |
| ERMU reset flag (RMU_RSTF0.ERMURF)                             | ×              | ×         | ×               | -                         | -                         | -              | -                      | ×                        | -              | -               | -             | -               | ✓          | -                           | -                |



| Reset Flag                                          | Reset mode     |           |                 |                           |                           |                |                        |                          |                |                 |               |                 |            |
|-----------------------------------------------------|----------------|-----------|-----------------|---------------------------|---------------------------|----------------|------------------------|--------------------------|----------------|-----------------|---------------|-----------------|------------|
|                                                     | Power-on Reset | Pin Reset | Brown-out Reset | Voltage Detection 1 Reset | Voltage Detection 2 Reset | Watchdog Reset | Special Watchdog Reset | Power-down Wake-up Reset | Software Reset | MPU Error Reset | RAM ECC Reset | FLASH ECC Reset | ERUM Reset |
| Clock frequency error reset flag (RMU_RSTF0.CKFERF) | ×              | ×         | ×               | -                         | -                         | -              | -                      | ×                        | -              | -               | -             | -               | ✓          |
| XTAL error reset flag (RMU_RSTF0.XTALERF)           | ×              | ×         | ×               | -                         | -                         | -              | -                      | ×                        | -              | -               | -             | -               | ✓          |
| Cortex-M4 Lockup reset flag (RSTF0.LKUPRF)          | ×              | ×         | ×               | -                         | -                         | -              | -                      | ×                        | -              | -               | -             | -               | ✓          |

**Note**

"✓" indicates setting; "×" indicates clearing; "-" indicates unchanged.

## 4.2.2 Reset Timing

### 4.2.2.1 Power-on Reset

Power-on reset is an internal reset caused by the power-on reset circuit, and the timing is as follows. Power-on reset occurs when the NRST pin is connected to the power supply at a high level. After a certain period of time ( $T_{RSTPOR}$ ) after the VCC voltage is higher than the power-on reset voltage  $V_{POR}$ , the internal reset of the chip is released, and the CPU starts to execute the code. When generating power-on reset, power-on reset flag RMU\_RSTF0.PORF is set. For detailed instructions on power-on reset, see ["Power-on/Power-off Reset Action Description"](#) in the "Power Control (PWC)" chapter.

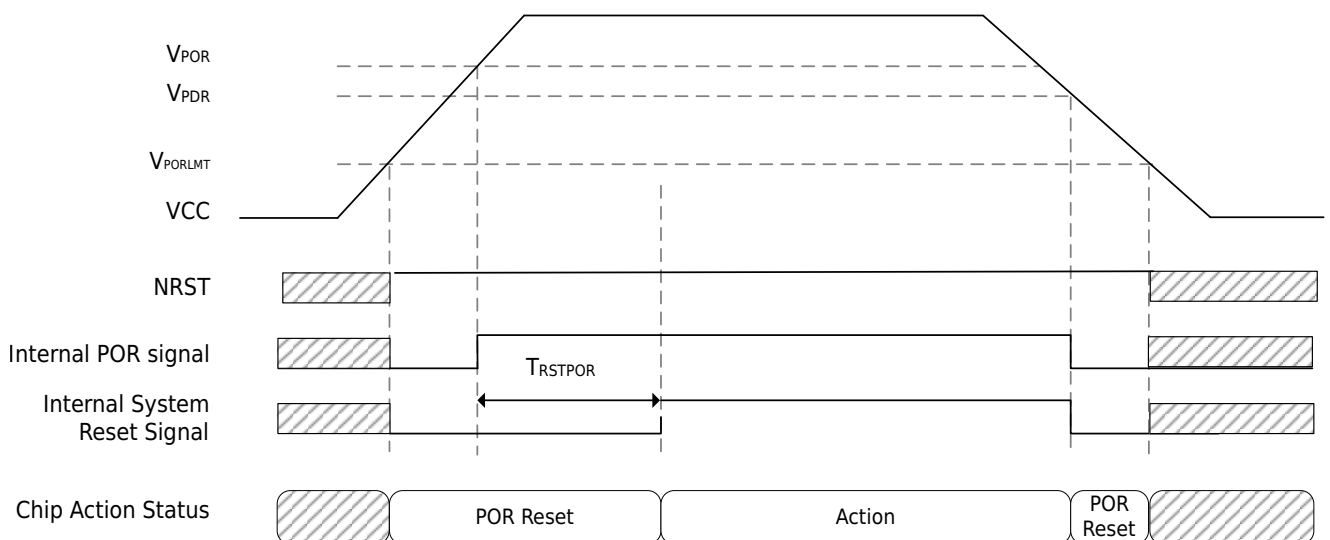


Figure 4-1 Power-on Reset Timing Diagram

#### 4.2.2.2 NRST Pin Reset

Pin reset is caused by the NRST pin being driven to a low level. The reset timing is shown in the figure below. After the NRST pin maintains a low level above the  $T_{NRST}$  width, after a certain internal reset time ( $T_{INRST}$ ), the internal reset is released.

When a NRST pin reset is generated, the pin reset flag `RMU_RSTF0.PINRF` is set.

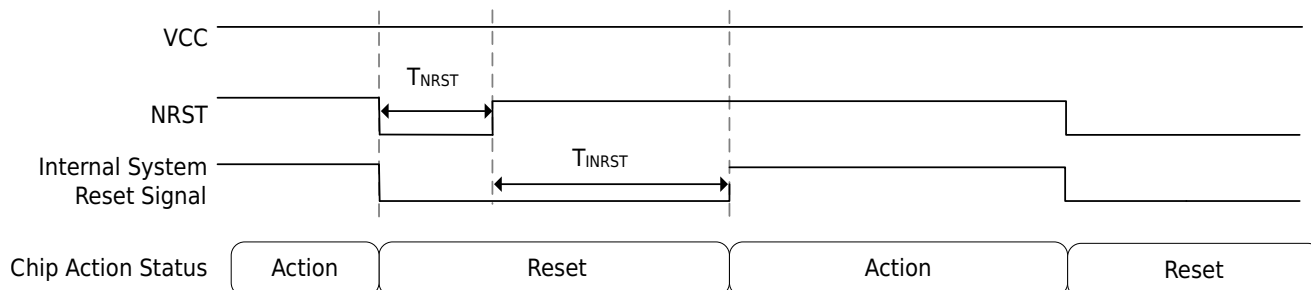


Figure 4-2 NRST Reset Timing Diagram

#### 4.2.2.3 Brown Out Reset

Brown-out reset is an internal reset caused by the voltage monitoring circuit, and the timing is shown in the figure below. After the undervoltage is set as reset enable through the ICG register, if the VCC voltage is lower than the monitoring voltage  $V_{BOR}$ , `RMU_RSTF0.BORF` will be set. When the VCC voltage is higher than the monitoring voltage  $V_{BOR}$ , the reset will be released after the  $V_{BOR}$  reset time ( $T_{RSTBOR}$ ).

For the reset setting of brownout, please refer to the "[Brown-out Reset \(BOR\) Description](#)" in the "Power Control (PWC)" chapter.

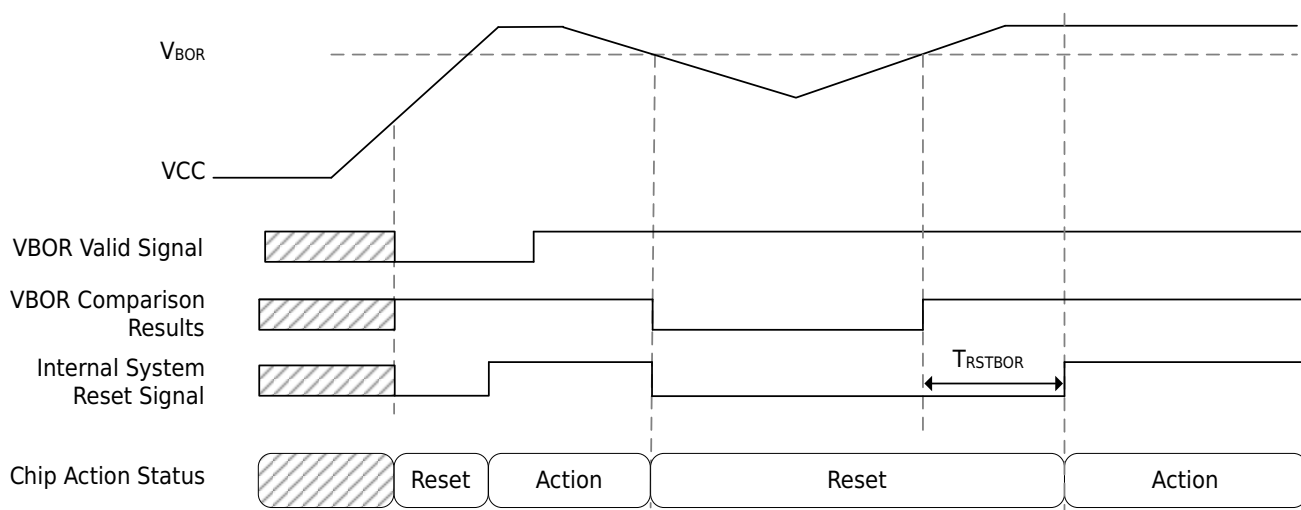


Figure 4-3 Brown-out Reset Timing Diagram

#### 4.2.2.4 Programmable Voltage Detection 1 Reset, Programmable Voltage Detection 2 Reset

Reset caused by the voltage monitor circuit when Programmable Voltage Detection 1 and Programmable Voltage Detection 2 are reset.

When the VCC voltage is lower than the monitoring voltage of the programmable voltage detection 1, a programmable voltage detection 1 reset may be generated according to the setting, and `RMU_RSTF0.PVD1RF` is set. When the VCC voltage is higher than the monitoring voltage of programmable voltage detection 1, the reset will be released after the reset time of PVD1 ( $T_{IPVD1}$ ).

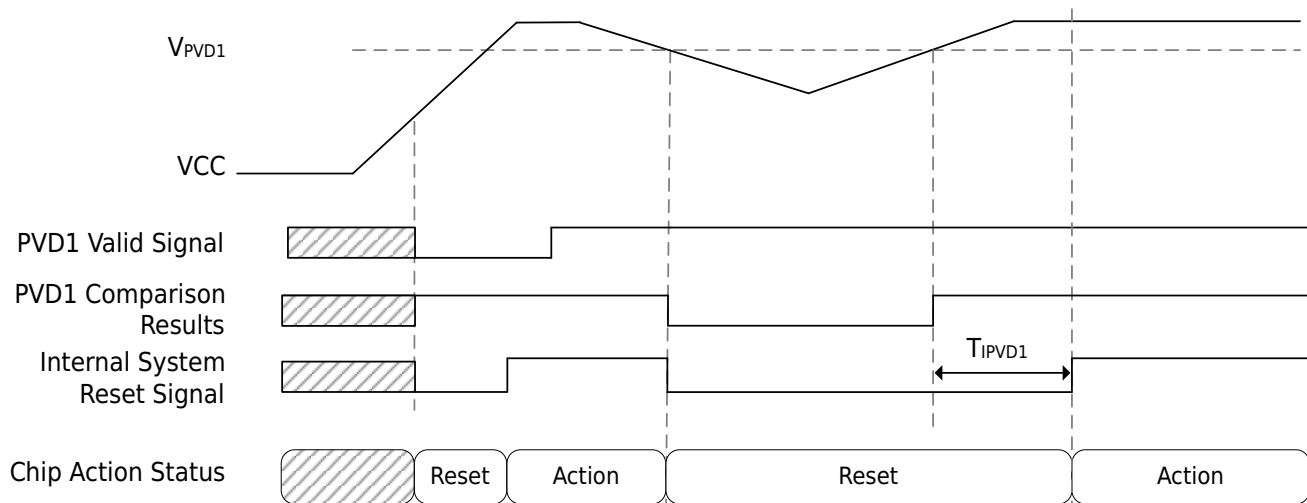


Figure 4-4 Programmable Voltage Detection 1 Reset Timing Diagram

When the VCC voltage is lower than the monitoring voltage of the programmable voltage detection 2, according to the setting, a programmable voltage detection 2 reset may be generated and RMU\_RSTF0.PVD2RF is set. When the VCC voltage is higher than the monitoring voltage of programmable voltage detection 2, the reset will be released after the reset time of PVD2 ( $T_{IPVD2}$ ).

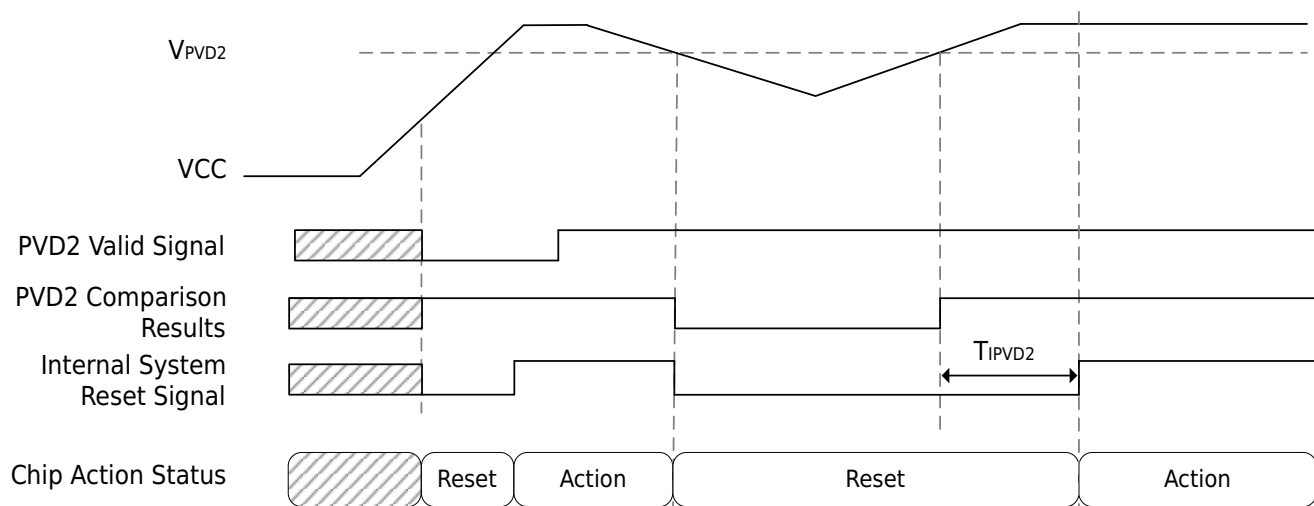


Figure 4-5 Programmable Voltage Detection 2 Reset Timing Diagram

For the reset settings of programmable voltage detection 1 and programmable voltage detection 2, see "[Programmable Voltage Detection 1 \(PVD1\)](#), [Programmable Voltage Detection 2 \(PVD2\)](#)" in the "Power Control (PWC)" chapter.

#### 4.2.2.5 Watchdog Reset, Special Watchdog Reset

The watchdog reset is an internal reset caused by the watchdog timer, and the special watchdog reset is an internal reset caused by the special watchdog timer. The reset timing is as follows.

After the watchdog reset is set to be active, the watchdog reset will be generated when the watchdog timer underflows or the write operation is not performed during the refresh period. A watchdog reset sets RMU\_RSTF0.WDRF. After the watchdog reset is generated, the chip is released from reset after the internal reset time  $T_{RIPT}$ .

After the special watchdog reset is set to be active, the watchdog reset will be generated when the special watchdog timer underflows or the write operation is not performed during the refresh period. Special

watchdog reset RMU\_RSTF0.SWDRF setup. After generating special watchdog reset, after internal reset time  $T_{RIPT}$ , the chip is released.

For details on watchdog reset and special watchdog reset, see the relevant description in the "Watchdog Timer (WDT/SWDT) " chapter.

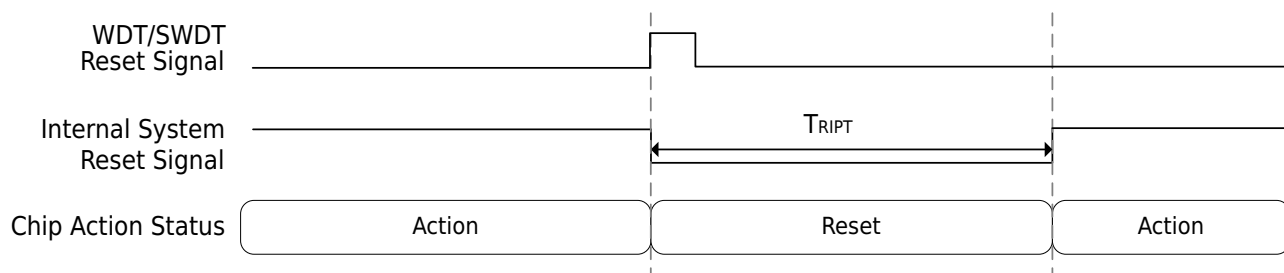


Figure 4-6 Watchdog and Special Watchdog Reset Timing Diagram

#### 4.2.2.6 Power-down Wake-up Reset

Power-down wake-up reset is an internal reset generated when the chip executes the WFI command when PWC\_PWRC0.PWDN is set to 1, enters Power-down mode, and releases Power-down mode through a Power-down mode wake-up event. The timing is as follows. After the Power-down mode is released and the return time  $T_{IPDx}$  ( $x=1, 2, 3, 4$ ) has elapsed, the Power-down wake-up reset is released. The return time varies according to the specific Power-down mode set, the minimum is in Power-down mode 1, and the maximum is in Power-down mode 3.

For details about Power-down wake-up reset, see "Power-down Mode" in the "Power Control (PWC) " chapter.

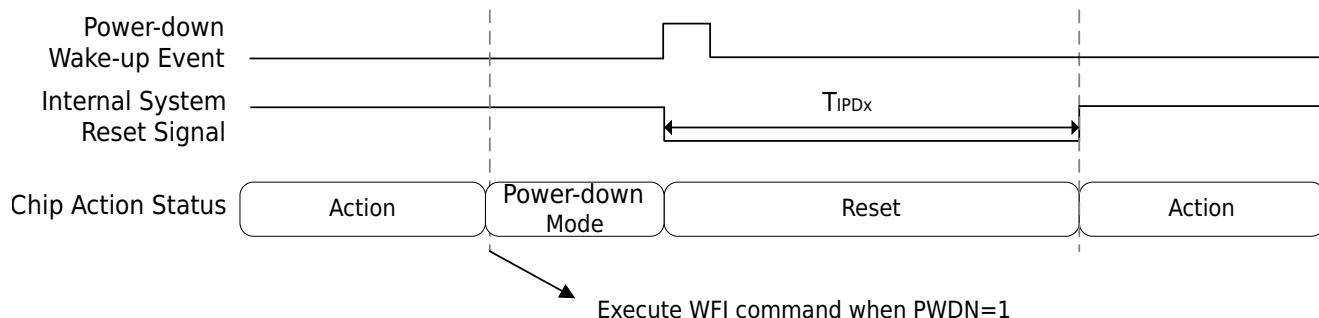


Figure 4-7 Power-off Wake-up Reset Timing Diagram

#### 4.2.2.7 Software Reset

A software reset is generated by writing the AIRCR.SYSRESETREQ bit of ARM register. When generating software reset, RMU\_RSTF0.SWRF bit is set. After the internal reset time  $T_{RIPT}$ , the chip is de-reset.

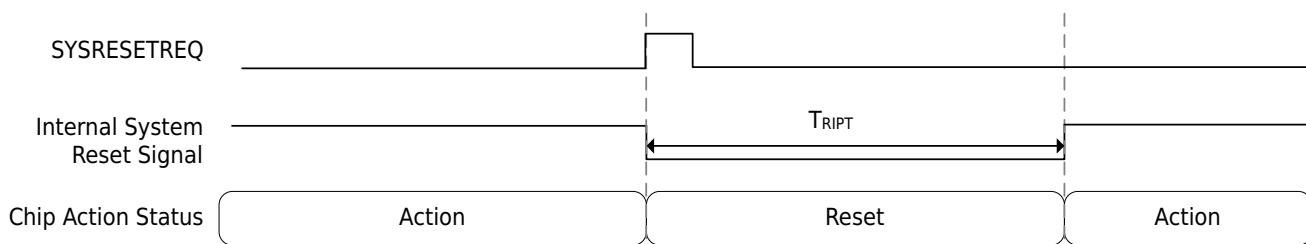


Figure 4-8 Software Reset Timing Diagram

#### 4.2.2.8 MPU Error Reset

The MPU error reset sets `RMU_RSTF0.MPUERF`, and the timing is shown below. After the internal reset time  $T_{RIPT}$ , the chip is de-reset.

For the settings of MPU error reset, see the relevant description in the "Memory Protection Unit (MPU) " chapter.

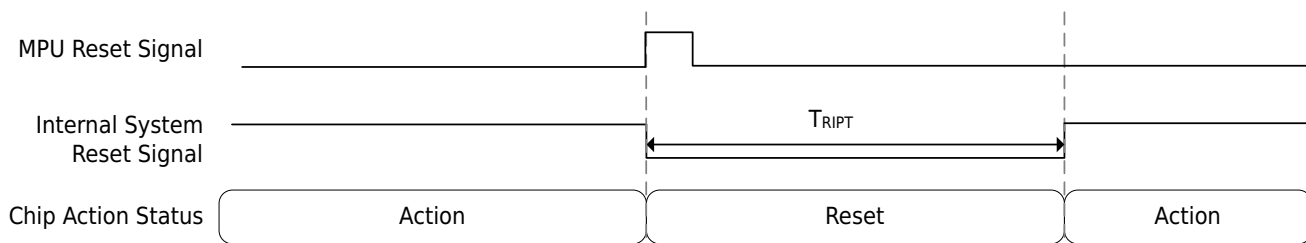


Figure 4-9 MPU Error Reset Timing Diagram

#### 4.2.2.9 RAM ECC Reset

When a RAM ECC check error occurs, a RAM ECC reset may occur according to the settings. The reset timing is as follows. A RAM ECC reset sets `RMU_RSTF0.RAECRF`. After the internal reset time  $T_{RIPT}$ , the chip is de-reset.

For the RAM ECC reset settings, please refer to the relevant description in the "Built-in SRAM (SRAM) " chapter.

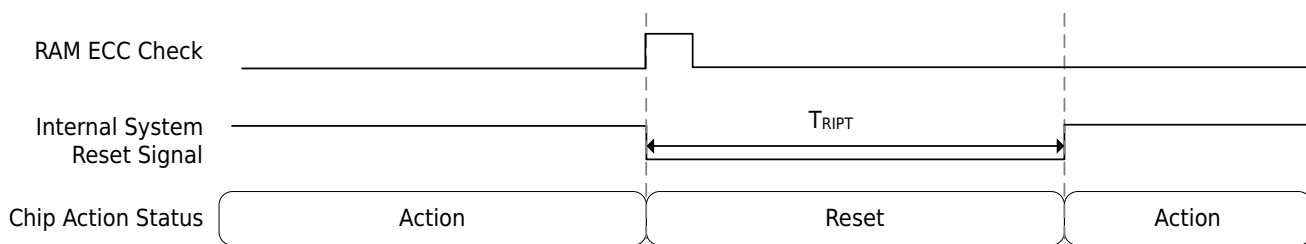


Figure 4-10 RAM ECC Reset Timing Diagram

#### 4.2.2.10 FLASH ECC Reset

When an error occurs in the FLASH ECC check, a FLASH ECC reset occurs, and the timing is as follows. A FLASH ECC reset sets `RMU_RSTF0.FLECRF`. After the internal reset time  $T_{RIPT}$ , the chip is de-reset.

For the setting of FLASH ECC reset, please refer to the relevant description in the "Embedded Flash Memory (EFM) " chapter.

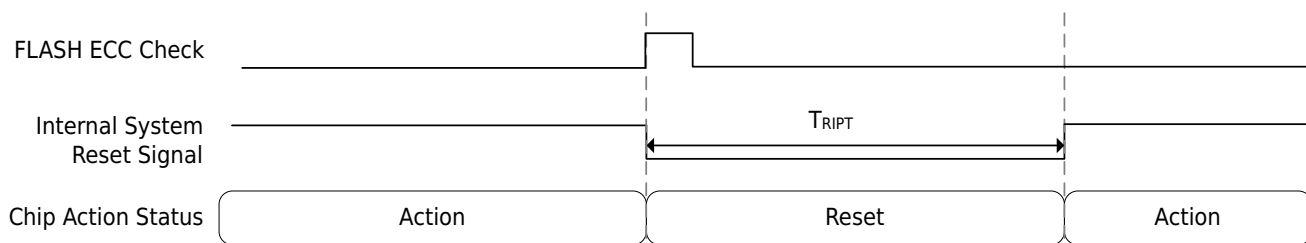


Figure 4-11 FLASH ECC Reset Timing Diagram

#### 4.2.2.11 ERMU Reset

When ERMU is reset, ERMU reset is generated, and the timing is as follows. An ERMU reset sets `RMU_RSTF0.ERMURF`. After the internal reset time  $T_{RIPT}$ , the chip is de-reset.

For the ERMU reset settings, please refer to the relevant description in the "Error Management Unit (ERMU)" chapter.

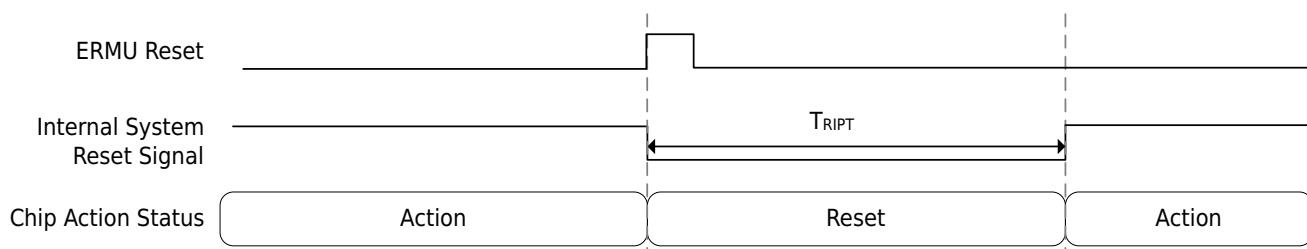


Figure 4-12 ERMU Reset Timing Diagram

#### 4.2.2.12 Clock Frequency Error Reset

When the FCM module built into the chip detects an abnormal clock frequency, it can generate a clock frequency error reset according to the settings. The reset timing is as follows. Clock frequency error reset sets `RMU_RSTF0.CKFERF`. After the internal reset time  $T_{RIPT}$ , the chip is de-reset.

For the setting of clock frequency error reset, refer to "[Clock Frequency Measurement](#)" in the "Clock Module Unit (CMU)" chapter.

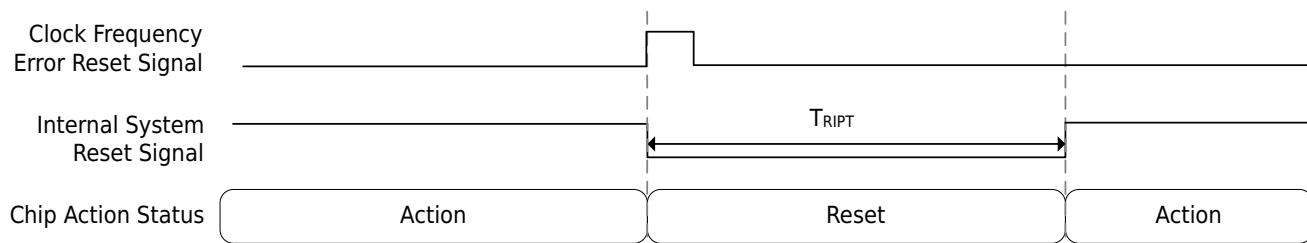


Figure 4-13 Clock Frequency Error Reset Timing Diagram

#### 4.2.2.13 XTAL Error Reset

When the chip's oscillation stop detection is enabled and reset is also enabled, if an abnormal stop of the external high-speed oscillator occurs, the XTAL error reset will be generated, and the `RMU_RSTF0.XTALERF` flag will be set. After the internal reset time  $T_{RIPT}$ , the chip is de-reset.

For the setting of XTAL error reset, refer to "[External High-speed Oscillator Fault Detection](#)" in the "Clock Module Unit (CMU)" chapter.

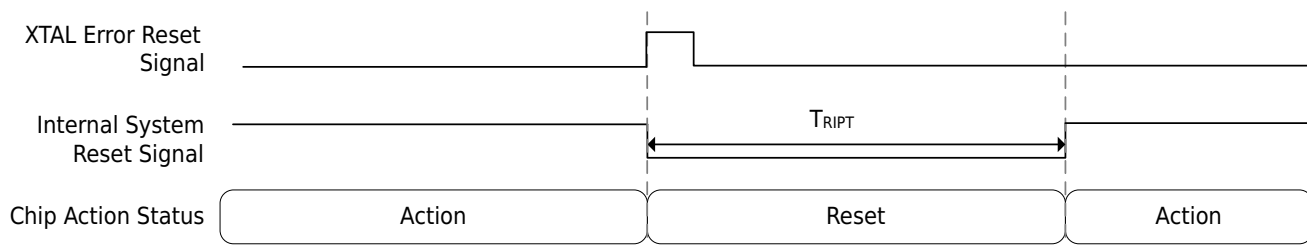


Figure 4-14 XTAL Error Reset Timing Diagram

#### 4.2.2.14 Cortex-M4 Lockup Reset

When Cortex-M4 encounters a serious exception, it will stop its PC pointer at the current address, lock it, and reset the entire chip after a delay of several clock periods. The timing is shown below. Cortex-M4 Lockup reset sets `RMU_RSTF0.LKUPRF`. After the internal reset time  $T_{RIPT}$ , the chip is de-reset.

Cortex-M4 Lockup reset can only be used after `RMU_PRSTCR0.LKUPREN` is set.

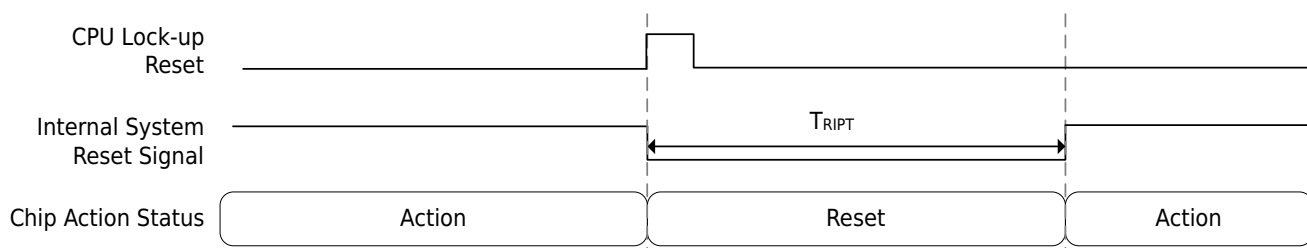


Figure 4-15 Cortex-M4 Lockup Reset Timing Diagram

### 4.2.3 Reset Mode Judgment

The reset mode can be judged according to the reset flag of RMU\_RSTF0. Multiple reset flags may occur when two or more resets occur simultaneously. Multiple resets occur when the RMU\_RSTF0.MULTIRF bit is 1. After reading RMU\_RSTF0, all reset flags can be cleared to 0 by setting the CLRFB bit. After setting and clearing RMU\_RSTF0, wait at least 6 CPU clock periods before reading the RMU\_RSTF0 register again.

### 4.2.4 Module Function Reset

In addition to the above-mentioned chip reset methods, the product also configures functional reset registers RMU\_FRST0-3 for each peripheral module. Writing 0 to the corresponding bit of RMU\_FRST can reset the function of the corresponding module. After RMU\_FRST is written to 0, it will be automatically set to 1 after 2 rising edge of peripheral module clock.

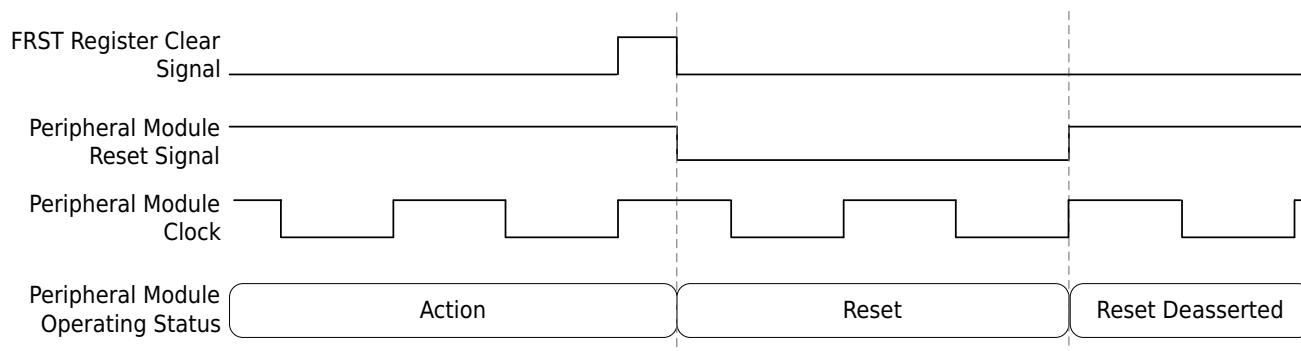


Figure 4-16 Module Function Reset

### 4.2.5 Reset Condition

Table 4-3 Reset Conditions for Each Module

| Modules | Register                                             | Reset Source                                        |
|---------|------------------------------------------------------|-----------------------------------------------------|
| DBG     | MCUSTPCTL<br>MCUSTPCTL2<br>MCUTRACECTL<br>MCUDBGSTAT | 1. Power-on reset<br>2. Power-down wake-up reset    |
| RTC     | RTC internal register                                | Module software reset control bit:<br>RTC_CR0.RESET |
| ERMU    | ERMU_ESSj                                            | 1. Power-on reset<br>2. Power-down wake-up reset    |

| Modules    | Register                                                                                                                                                                                         | Reset Source                                                                                                                                                                                                             |
|------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ERMU       | ERMU_EOyC<br>ERMU_EOyS<br>ERMU_EOyCTCNT<br>ERMU_EOyCTCMP<br>ERMU_EOyOM0<br>ERMU_EOyOM1                                                                                                           | All reset sources except ERMU functional reset (RMU_FRST0.ERMU)                                                                                                                                                          |
|            | Other registers than the above                                                                                                                                                                   | All reset sources                                                                                                                                                                                                        |
| MCAN       | MCANRAM_CKSR, MCANRAM_ECCERRADDR                                                                                                                                                                 | 1. Power-on reset<br>2. Power-down wake-up reset                                                                                                                                                                         |
|            | Other registers than the above                                                                                                                                                                   | All reset sources                                                                                                                                                                                                        |
| CAN1/CAN2  | CAN1_RAM_CKSR<br>CAN1_RAM_ECCERRADDR<br>CAN2_RAM_CKSR<br>CAN2_RAM_ECCERRADDR                                                                                                                     | 1. Power-on reset<br>2. Power-down wake-up reset                                                                                                                                                                         |
|            | Other registers than the above                                                                                                                                                                   | All reset sources                                                                                                                                                                                                        |
| PKE        | PKE_RAM0_CKSR<br>PKE_RAM0_ECCERRADDR<br>PKE_RAM1_CKSR<br>PKE_RAM1_ECCERRADDR                                                                                                                     | 1. Power-on reset<br>2. Power-down wake-up reset                                                                                                                                                                         |
|            | Other registers than the above                                                                                                                                                                   | All reset sources                                                                                                                                                                                                        |
| EFM        | EFM_CKSR, EFM_F0ERRn (n=0-1),<br>EFM_F1ERRn (n=0-1)                                                                                                                                              | 1. Power-on reset<br>2. Power-down wake-up reset                                                                                                                                                                         |
|            | Other registers than the above                                                                                                                                                                   | All reset sources                                                                                                                                                                                                        |
| SRAM       | SRAMC_CKSR<br>SRAMC_SRAM1_ECCERRADDR<br>SRAMC_SRAM2_ECCERRADDR<br>SRAMC_SRAM3_ECCERRADDR<br>SRAMC_SRAM4_ECCERRADDR<br>SRAMC_SRAMH_ECCERRADDR<br>SRAMC_SRAMB_ECCERRADDR<br>SRAMC_CACHE_ECCERRADDR | 1. Power-on reset<br>2. Power-down wake-up reset                                                                                                                                                                         |
|            | Other registers than the above                                                                                                                                                                   | All reset sources                                                                                                                                                                                                        |
| PWC<br>CMU | PWC_PWRC0<br>PWC_PWRC1<br>PWC_PDWKE0<br>PWC_PDWKE1<br>PWC_PDWKE2<br>PWC_PDWKES<br>PWC_PWRC1.PDTS<br>CMU_XTALCFGR                                                                                 | All reset sources other than the following:<br><ul style="list-style-type: none"> <li>● Power-down mode 1 wake-up reset</li> <li>● Power-down mode 2 wake-up reset</li> <li>● Power-down mode 4 wake-up reset</li> </ul> |
|            | PWC_PDWKF0<br>PWC_PDWKF1                                                                                                                                                                         | All reset sources except Power-down wake-up reset                                                                                                                                                                        |



| Modules                                | Register                                                                                                              | Reset Source                                                                                                                                    |
|----------------------------------------|-----------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------|
| PWC<br>CMU                             | PWC_PVDLCR<br>PWC_PVDCR1<br>PWC_PVDPCR<br>PWC_PVDCR0<br>Other bits than PWC_PWRC1.PDTS<br>PWR_PWRC4                   | 1. Power-on reset<br>2. Pin reset<br>3. Brown-out reset<br>4. Watchdog reset<br>5. Special watchdog reset<br>6. Power-down mode 3 wake-up reset |
|                                        | PWC_PVDICR[2:0]<br>PWC_PVDDSR.PVD1DETFLG<br>PWC_PVDICR[6:4]<br>PWC_PVDDSR.PVD2DETFLG                                  | 1. Power-on reset<br>2. Pin reset<br>3. Brown-out reset<br>4. Watchdog reset<br>5. Special watchdog reset<br>6. Power-down wake-up reset        |
|                                        | PWC_VBATCR<br>CMU_XTAL32CFGR<br>CMU_XTAL32NFR<br>CMU_XTAL32CR<br>CMU_RTCLRCTRM<br>PWC_WKTC0<br>PWC_WKTC1<br>PWC_WKTC2 | VBAT domain reset, method: write 0xA5 to PWC_VBATRSTR                                                                                           |
|                                        | Other registers than the above                                                                                        | All reset sources                                                                                                                               |
| Registers other than the above modules |                                                                                                                       | All reset sources                                                                                                                               |

## 4.3 Register Description

### 4.3.1 Register List

Base address: 0x4004CC00

Table 4-4 Reset Module Unit (RMU) Register Offset

| Offset | Register    | Description                       | Width |
|--------|-------------|-----------------------------------|-------|
| 0xC0   | RMU_FRST0   | Function Reset Control Register 0 | 32    |
| 0xC4   | RMU_FRST1   | Function Reset Control Register 1 | 32    |
| 0xC8   | RMU_FRST2   | Function Reset Control Register 2 | 32    |
| 0xCC   | RMU_FRST3   | Function Reset Control Register 3 | 32    |
| 0xF8   | RMU_PRSTCR0 | Reset Control Register 0          | 8     |
| 0xFC   | RMU_RSTF0   | Reset Flag Register 0             | 32    |

## 4.3.2 Function Reset Control Register 0 (RMU\_FRST0)

| Offset | Bit Position |      |      |      |      |      |      |      |     |      |      |     |     |     |     |     |      |      |         |     |          |    |   |      |          |   |   |   |   |   |   |   |  |
|--------|--------------|------|------|------|------|------|------|------|-----|------|------|-----|-----|-----|-----|-----|------|------|---------|-----|----------|----|---|------|----------|---|---|---|---|---|---|---|--|
| 0xC0   | 31           | 30   | 29   | 28   | 27   | 26   | 25   | 24   | 23  | 22   | 21   | 20  | 19  | 18  | 17  | 16  | 15   | 14   | 13      | 12  | 11       | 10 | 9 | 8    | 7        | 6 | 5 | 4 | 3 | 2 | 1 | 0 |  |
| Reset  | 0xFFFFFFFF   |      |      |      |      |      |      |      |     |      |      |     |     |     |     |     |      |      |         |     |          |    |   |      |          |   |   |   |   |   |   |   |  |
| Name   | DCU8         | DCU7 | DCU6 | DCU5 | DCU4 | DCU3 | DCU2 | DCU1 | CRC | TRNG | HASH | SKE | MAU | CTC | AOS | FCM | DMA2 | DMA1 | KEYSCAN | PKE | Reserved |    |   | ERMU | Reserved |   |   |   |   |   |   |   |  |
| Access | RW           | RW   | RW   | RW   | RW   | RW   | RW   | RW   | RW  | RW   | RW   | RW  | RW  | RW  | RW  | RW  | RW   | RW   | RW      | RW  |          |    |   | RW   |          |   |   |   |   |   |   |   |  |

| Bit | Name | Function           | Description                                                                                                         | Access |
|-----|------|--------------------|---------------------------------------------------------------------------------------------------------------------|--------|
| 31  | DCU8 | DCU8 reset control | <ul style="list-style-type: none"> <li>0b0: Function reset enabled</li> <li>0b1: Function reset released</li> </ul> | RW     |
| 30  | DCU7 | DCU7 reset control | <ul style="list-style-type: none"> <li>0b0: Function reset enabled</li> <li>0b1: Function reset released</li> </ul> | RW     |
| 29  | DCU6 | DCU6 reset control | <ul style="list-style-type: none"> <li>0b0: Function reset enabled</li> <li>0b1: Function reset released</li> </ul> | RW     |
| 28  | DCU5 | DCU5 reset control | <ul style="list-style-type: none"> <li>0b0: Function reset enabled</li> <li>0b1: Function reset released</li> </ul> | RW     |
| 27  | DCU4 | DCU4 reset control | <ul style="list-style-type: none"> <li>0b0: Function reset enabled</li> <li>0b1: Function reset released</li> </ul> | RW     |
| 26  | DCU3 | DCU3 reset control | <ul style="list-style-type: none"> <li>0b0: Function reset enabled</li> <li>0b1: Function reset released</li> </ul> | RW     |
| 25  | DCU2 | DCU2 reset control | <ul style="list-style-type: none"> <li>0b0: Function reset enabled</li> <li>0b1: Function reset released</li> </ul> | RW     |
| 24  | DCU1 | DCU1 reset control | <ul style="list-style-type: none"> <li>0b0: Function reset enabled</li> <li>0b1: Function reset released</li> </ul> | RW     |
| 23  | CRC  | CRC reset control  | <ul style="list-style-type: none"> <li>0b0: Function reset enabled</li> <li>0b1: Function reset released</li> </ul> | RW     |
| 22  | TRNG | TRNG reset control | <ul style="list-style-type: none"> <li>0b0: Function reset enabled</li> <li>0b1: Function reset released</li> </ul> | RW     |
| 21  | HASH | HASH reset control | <ul style="list-style-type: none"> <li>0b0: Function reset enabled</li> <li>0b1: Function reset released</li> </ul> | RW     |
| 20  | SKE  | SKE reset control  | <ul style="list-style-type: none"> <li>0b0: Function reset enabled</li> <li>0b1: Function reset released</li> </ul> | RW     |
| 19  | MAU  | MAU reset control  | <ul style="list-style-type: none"> <li>0b0: Function reset enabled</li> <li>0b1: Function reset released</li> </ul> | RW     |
| 18  | CTC  | CTC reset control  | <ul style="list-style-type: none"> <li>0b0: Function reset enabled</li> <li>0b1: Function reset released</li> </ul> | RW     |

| Bit  | Name            | Function              | Description                                                                                                             | Access |
|------|-----------------|-----------------------|-------------------------------------------------------------------------------------------------------------------------|--------|
| 17   | AOS             | AOS reset control     | <ul style="list-style-type: none"> <li>● 0b0: Function reset enabled</li> <li>● 0b1: Function reset released</li> </ul> | RW     |
| 16   | FCM             | FCM reset control     | <ul style="list-style-type: none"> <li>● 0b0: Function reset enabled</li> <li>● 0b1: Function reset released</li> </ul> | RW     |
| 15   | DMA2            | DMA2 reset control    | <ul style="list-style-type: none"> <li>● 0b0: Function reset enabled</li> <li>● 0b1: Function reset released</li> </ul> | RW     |
| 14   | DMA1            | DMA1 reset control    | <ul style="list-style-type: none"> <li>● 0b0: Function reset enabled</li> <li>● 0b1: Function reset released</li> </ul> | RW     |
| 13   | KEYSCAN         | KEYSCAN reset control | <ul style="list-style-type: none"> <li>● 0b0: Function reset enabled</li> <li>● 0b1: Function reset released</li> </ul> | RW     |
| 12   | PKE             | PKE reset control     | <ul style="list-style-type: none"> <li>● 0b0: Function reset enabled</li> <li>● 0b1: Function reset released</li> </ul> | RW     |
| 11:9 | <i>Reserved</i> | <i>Reserved</i>       | <i>All bits read as 1, Write as 1</i>                                                                                   | -      |
| 8    | ERMU            | ERMU reset control    | <ul style="list-style-type: none"> <li>● 0b0: Function reset enabled</li> <li>● 0b1: Function reset released</li> </ul> | RW     |
| 7:0  | <i>Reserved</i> | <i>Reserved</i>       | <i>All bits read as 1, Write as 1</i>                                                                                   | -      |

## 4.3.3 Function Reset Control Register 1 (RMU\_FRST1)

| Offset | Bit Position |       |       |       |       |       |       |       |       |      |      |      |      |      |      |      |      |      |      |        |        |      |      |      |      |      |      |      |        |      |      |    |
|--------|--------------|-------|-------|-------|-------|-------|-------|-------|-------|------|------|------|------|------|------|------|------|------|------|--------|--------|------|------|------|------|------|------|------|--------|------|------|----|
| 0xC4   | 31           | 30    | 29    | 28    | 27    | 26    | 25    | 24    | 23    | 22   | 21   | 20   | 19   | 18   | 17   | 16   | 15   | 14   | 13   | 12     | 11     | 10   | 9    | 8    | 7    | 6    | 5    | 4    | 3      | 2    | 1    | 0  |
| Reset  | 0xFFFFFFFF   |       |       |       |       |       |       |       |       |      |      |      |      |      |      |      |      |      |      |        |        |      |      |      |      |      |      |      |        |      |      |    |
| Name   | Reserved     | MCAN2 | MCAN1 | FMAC4 | FMAC3 | FMAC2 | FMAC1 | USBHS | USBFS | SPI6 | SPI5 | SPI4 | SPI3 | SPI2 | SPI1 | I2S4 | I2S3 | I2S2 | I2S1 | SDIOC2 | SDIOC1 | I2C6 | I2C5 | I2C4 | I2C3 | I2C2 | I2C1 | QSPI | ETHMAC | CAN2 | CAN1 |    |
| Access |              | RW    | RW    | RW    | RW    | RW    | RW    | RW    | RW    | RW   | RW   | RW   | RW   | RW   | RW   | RW   | RW   | RW   | RW   | RW     | RW     | RW   | RW   | RW   | RW   | RW   | RW   | RW   | RW     | RW   | RW   | RW |

| Bit   | Name     | Function            | Description                                                                                                         | Access |
|-------|----------|---------------------|---------------------------------------------------------------------------------------------------------------------|--------|
| 31:30 | Reserved | Reserved            | All bits read as 1, Write as 1                                                                                      | -      |
| 29    | MCAN2    | MCAN2 reset control | <ul style="list-style-type: none"> <li>0b0: Function reset enabled</li> <li>0b1: Function reset released</li> </ul> | RW     |
| 28    | MCAN1    | MCAN1 reset control | <ul style="list-style-type: none"> <li>0b0: Function reset enabled</li> <li>0b1: Function reset released</li> </ul> | RW     |
| 27    | FMAC4    | FMAC4 reset control | <ul style="list-style-type: none"> <li>0b0: Function reset enabled</li> <li>0b1: Function reset released</li> </ul> | RW     |
| 26    | FMAC3    | FMAC3 reset control | <ul style="list-style-type: none"> <li>0b0: Function reset enabled</li> <li>0b1: Function reset released</li> </ul> | RW     |
| 25    | FMAC2    | FMAC2 reset control | <ul style="list-style-type: none"> <li>0b0: Function reset enabled</li> <li>0b1: Function reset released</li> </ul> | RW     |
| 24    | FMAC1    | FMAC1 reset control | <ul style="list-style-type: none"> <li>0b0: Function reset enabled</li> <li>0b1: Function reset released</li> </ul> | RW     |
| 23    | USBHS    | USBHS reset control | <ul style="list-style-type: none"> <li>0b0: Function reset enabled</li> <li>0b1: Function reset released</li> </ul> | RW     |
| 22    | USBFS    | USBFS reset control | <ul style="list-style-type: none"> <li>0b0: Function reset enabled</li> <li>0b1: Function reset released</li> </ul> | RW     |
| 21    | SPI6     | SPI6 reset control  | <ul style="list-style-type: none"> <li>0b0: Function reset enabled</li> <li>0b1: Function reset released</li> </ul> | RW     |
| 20    | SPI5     | SPI5 reset control  | <ul style="list-style-type: none"> <li>0b0: Function reset enabled</li> <li>0b1: Function reset released</li> </ul> | RW     |
| 19    | SPI4     | SPI4 reset control  | <ul style="list-style-type: none"> <li>0b0: Function reset enabled</li> <li>0b1: Function reset released</li> </ul> | RW     |
| 18    | SPI3     | SPI3 reset control  | <ul style="list-style-type: none"> <li>0b0: Function reset enabled</li> <li>0b1: Function reset released</li> </ul> | RW     |
| 17    | SPI2     | SPI2 reset control  | <ul style="list-style-type: none"> <li>0b0: Function reset enabled</li> <li>0b1: Function reset released</li> </ul> | RW     |

| Bit | Name   | Function             | Description                                                                                                             | Access |
|-----|--------|----------------------|-------------------------------------------------------------------------------------------------------------------------|--------|
| 16  | SPI1   | SPI1 reset control   | <ul style="list-style-type: none"> <li>● 0b0: Function reset enabled</li> <li>● 0b1: Function reset released</li> </ul> | RW     |
| 15  | I2S4   | I2S4 reset control   | <ul style="list-style-type: none"> <li>● 0b0: Function reset enabled</li> <li>● 0b1: Function reset released</li> </ul> | RW     |
| 14  | I2S3   | I2S3 reset control   | <ul style="list-style-type: none"> <li>● 0b0: Function reset enabled</li> <li>● 0b1: Function reset released</li> </ul> | RW     |
| 13  | I2S2   | I2S2 reset control   | <ul style="list-style-type: none"> <li>● 0b0: Function reset enabled</li> <li>● 0b1: Function reset released</li> </ul> | RW     |
| 12  | I2S1   | I2S1 reset control   | <ul style="list-style-type: none"> <li>● 0b0: Function reset enabled</li> <li>● 0b1: Function reset released</li> </ul> | RW     |
| 11  | SDIOC2 | SDIOC2 reset control | <ul style="list-style-type: none"> <li>● 0b0: Function reset enabled</li> <li>● 0b1: Function reset released</li> </ul> | RW     |
| 10  | SDIOC1 | SDIOC1 reset control | <ul style="list-style-type: none"> <li>● 0b0: Function reset enabled</li> <li>● 0b1: Function reset released</li> </ul> | RW     |
| 9   | I2C6   | I2C6 reset control   | <ul style="list-style-type: none"> <li>● 0b0: Function reset enabled</li> <li>● 0b1: Function reset released</li> </ul> | RW     |
| 8   | I2C5   | I2C5 reset control   | <ul style="list-style-type: none"> <li>● 0b0: Function reset enabled</li> <li>● 0b1: Function reset released</li> </ul> | RW     |
| 7   | I2C4   | I2C4 reset control   | <ul style="list-style-type: none"> <li>● 0b0: Function reset enabled</li> <li>● 0b1: Function reset released</li> </ul> | RW     |
| 6   | I2C3   | I2C3 reset control   | <ul style="list-style-type: none"> <li>● 0b0: Function reset enabled</li> <li>● 0b1: Function reset released</li> </ul> | RW     |
| 5   | I2C2   | I2C2 reset control   | <ul style="list-style-type: none"> <li>● 0b0: Function reset enabled</li> <li>● 0b1: Function reset released</li> </ul> | RW     |
| 4   | I2C1   | I2C1 reset control   | <ul style="list-style-type: none"> <li>● 0b0: Function reset enabled</li> <li>● 0b1: Function reset released</li> </ul> | RW     |
| 3   | QSPI   | QSPI reset control   | <ul style="list-style-type: none"> <li>● 0b0: Function reset enabled</li> <li>● 0b1: Function reset released</li> </ul> | RW     |
| 2   | ETHMAC | ETHMAC reset control | <ul style="list-style-type: none"> <li>● 0b0: Function reset enabled</li> <li>● 0b1: Function reset released</li> </ul> | RW     |
| 1   | CAN2   | CAN2 reset control   | <ul style="list-style-type: none"> <li>● 0b0: Function reset enabled</li> <li>● 0b1: Function reset released</li> </ul> | RW     |
| 0   | CAN1   | CAN1 reset control   | <ul style="list-style-type: none"> <li>● 0b0: Function reset enabled</li> <li>● 0b1: Function reset released</li> </ul> | RW     |

## 4.3.4 Function Reset Control Register 2 (RMU\_FRST2)

| Offset | Bit Position |         |         |        |        |        |        |        |        |        |        |        |        |        |        |        |     |        |        |        |          |        |        |        |          |   |   |   |      |   |   |   |
|--------|--------------|---------|---------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|-----|--------|--------|--------|----------|--------|--------|--------|----------|---|---|---|------|---|---|---|
| 0xC8   | 31           | 30      | 29      | 28     | 27     | 26     | 25     | 24     | 23     | 22     | 21     | 20     | 19     | 18     | 17     | 16     | 15  | 14     | 13     | 12     | 11       | 10     | 9      | 8      | 7        | 6 | 5 | 4 | 3    | 2 | 1 | 0 |
| Reset  | 0xFFFFFFFF   |         |         |        |        |        |        |        |        |        |        |        |        |        |        |        |     |        |        |        |          |        |        |        |          |   |   |   |      |   |   |   |
| Name   | TMRA_12      | TMRA_11 | TMRA_10 | TMRA_9 | TMRA_8 | TMRA_7 | TMRA_6 | TMRA_5 | TMRA_4 | TMRA_3 | TMRA_2 | TMRA_1 | TMR2_4 | TMR2_3 | TMR2_2 | TMR2_1 | EMB | TMR0_3 | TMR0_2 | TMR0_1 | Reserved | TMR4_3 | TMR4_2 | TMR4_1 | Reserved |   |   |   | TMR6 |   |   |   |
| Access | RW           | RW      | RW      | RW     | RW     | RW     | RW     | RW     | RW     | RW     | RW     | RW     | RW     | RW     | RW     | RW     | RW  | RW     | RW     | RW     |          | RW     | RW     | RW     |          |   |   |   | RW   |   |   |   |

| Bit | Name    | Function                      | Description                                                                                                         | Access |
|-----|---------|-------------------------------|---------------------------------------------------------------------------------------------------------------------|--------|
| 31  | TMRA_12 | TMRA_12 reset control         | <ul style="list-style-type: none"> <li>0b0: Function reset enabled</li> <li>0b1: Function reset released</li> </ul> | RW     |
| 30  | TMRA_11 | TMRA_11 reset control         | <ul style="list-style-type: none"> <li>0b0: Function reset enabled</li> <li>0b1: Function reset released</li> </ul> | RW     |
| 29  | TMRA_10 | TMRA_10 reset control         | <ul style="list-style-type: none"> <li>0b0: Function reset enabled</li> <li>0b1: Function reset released</li> </ul> | RW     |
| 28  | TMRA_9  | TMRA_9 reset control          | <ul style="list-style-type: none"> <li>0b0: Function reset enabled</li> <li>0b1: Function reset released</li> </ul> | RW     |
| 27  | TMRA_8  | TMRA_8 reset control          | <ul style="list-style-type: none"> <li>0b0: Function reset enabled</li> <li>0b1: Function reset released</li> </ul> | RW     |
| 26  | TMRA_7  | TMRA_7 reset control          | <ul style="list-style-type: none"> <li>0b0: Function reset enabled</li> <li>0b1: Function reset released</li> </ul> | RW     |
| 25  | TMRA_6  | TMRA_6 reset control          | <ul style="list-style-type: none"> <li>0b0: Function reset enabled</li> <li>0b1: Function reset released</li> </ul> | RW     |
| 24  | TMRA_5  | TMRA_5 reset control          | <ul style="list-style-type: none"> <li>0b0: Function reset enabled</li> <li>0b1: Function reset released</li> </ul> | RW     |
| 23  | TMRA_4  | TMRA_4 reset control          | <ul style="list-style-type: none"> <li>0b0: Function reset enabled</li> <li>0b1: Function reset released</li> </ul> | RW     |
| 22  | TMRA_3  | TMRA_3 reset control          | <ul style="list-style-type: none"> <li>0b0: Function reset enabled</li> <li>0b1: Function reset released</li> </ul> | RW     |
| 21  | TMRA_2  | TMRA_2 reset control          | <ul style="list-style-type: none"> <li>0b0: Function reset enabled</li> <li>0b1: Function reset released</li> </ul> | RW     |
| 20  | TMRA_1  | TMRA_1 function reset control | <ul style="list-style-type: none"> <li>0b0: Function reset enabled</li> <li>0b1: Function reset released</li> </ul> | RW     |
| 19  | TMR2_4  | TMR2_4 reset control          | <ul style="list-style-type: none"> <li>0b0: Function reset enabled</li> <li>0b1: Function reset released</li> </ul> | RW     |
| 18  | TMR2_3  | TMR2_3 reset control          | <ul style="list-style-type: none"> <li>0b0: Function reset enabled</li> <li>0b1: Function reset released</li> </ul> | RW     |

| Bit | Name            | Function             | Description                                                                                                             | Access |
|-----|-----------------|----------------------|-------------------------------------------------------------------------------------------------------------------------|--------|
| 17  | TMR2_2          | TMR2_2 reset control | <ul style="list-style-type: none"> <li>● 0b0: Function reset enabled</li> <li>● 0b1: Function reset released</li> </ul> | RW     |
| 16  | TMR2_1          | TMR2_1 reset control | <ul style="list-style-type: none"> <li>● 0b0: Function reset enabled</li> <li>● 0b1: Function reset released</li> </ul> | RW     |
| 15  | EMB             | EMB reset control    | <ul style="list-style-type: none"> <li>● 0b0: Function reset enabled</li> <li>● 0b1: Function reset released</li> </ul> | RW     |
| 14  | TMR0_3          | TMR0_3 reset control | <ul style="list-style-type: none"> <li>● 0b0: Function reset enabled</li> <li>● 0b1: Function reset released</li> </ul> | RW     |
| 13  | TMR0_2          | TMR0_2 reset control | <ul style="list-style-type: none"> <li>● 0b0: Function reset enabled</li> <li>● 0b1: Function reset released</li> </ul> | RW     |
| 12  | TMR0_1          | TMR0_1 reset control | <ul style="list-style-type: none"> <li>● 0b0: Function reset enabled</li> <li>● 0b1: Function reset released</li> </ul> | RW     |
| 11  | <i>Reserved</i> | <i>Reserved</i>      | <i>Reads as 1, Write as 1</i>                                                                                           | -      |
| 10  | TMR4_3          | TMR4_3 reset control | <ul style="list-style-type: none"> <li>● 0b0: Function reset enabled</li> <li>● 0b1: Function reset released</li> </ul> | RW     |
| 9   | TMR4_2          | TMR4_2 reset control | <ul style="list-style-type: none"> <li>● 0b0: Function reset enabled</li> <li>● 0b1: Function reset released</li> </ul> | RW     |
| 8   | TMR4_1          | TMR4_1 reset control | <ul style="list-style-type: none"> <li>● 0b0: Function reset enabled</li> <li>● 0b1: Function reset released</li> </ul> | RW     |
| 7:1 | <i>Reserved</i> | <i>Reserved</i>      | <i>All bits read as 1, Write as 1</i>                                                                                   | -      |
| 0   | TMR6            | TMR6 reset control   | <ul style="list-style-type: none"> <li>● 0b0: Function reset enabled</li> <li>● 0b1: Function reset released</li> </ul> | RW     |



## 4.3.5 Function Reset Control Register 3 (RMU\_FRST3)

| Offset | Bit Position |         |        |        |        |        |        |        |        |        |        |          |     |     |     |     |          |     |        |        |       |       |          |      |      |          |      |      |      |    |    |   |
|--------|--------------|---------|--------|--------|--------|--------|--------|--------|--------|--------|--------|----------|-----|-----|-----|-----|----------|-----|--------|--------|-------|-------|----------|------|------|----------|------|------|------|----|----|---|
| 0xCC   | 31           | 30      | 29     | 28     | 27     | 26     | 25     | 24     | 23     | 22     | 21     | 20       | 19  | 18  | 17  | 16  | 15       | 14  | 13     | 12     | 11    | 10    | 9        | 8    | 7    | 6        | 5    | 4    | 3    | 2  | 1  | 0 |
| Reset  | 0xFFFFFFFF   |         |        |        |        |        |        |        |        |        |        |          |     |     |     |     |          |     |        |        |       |       |          |      |      |          |      |      |      |    |    |   |
| Name   | Reserved     | USART10 | USART9 | USART8 | USART7 | USART6 | USART5 | USART4 | USART3 | USART2 | USART1 | Reserved | NFC | DMC | SMC | DVP | Reserved | OTS | TMR0_5 | TMR0_4 | CMP34 | CMP12 | Reserved | DAC2 | DAC1 | Reserved | ADC3 | ADC2 | ADC1 |    |    |   |
| Access |              | RW      | RW     | RW     | RW     | RW     | RW     | RW     | RW     | RW     | RW     |          | RW  | RW  | RW  | RW  |          | RW  | RW     | RW     | RW    | RW    |          | RW   | RW   |          | RW   | RW   | RW   | RW | RW |   |

| Bit   | Name     | Function              | Description                                                                                                         | Access |
|-------|----------|-----------------------|---------------------------------------------------------------------------------------------------------------------|--------|
| 31:30 | Reserved | Reserved              | All bits read as 1, Write as 1                                                                                      | -      |
| 29    | USART10  | USART10 reset control | <ul style="list-style-type: none"> <li>0b0: Function reset enabled</li> <li>0b1: Function reset released</li> </ul> | RW     |
| 28    | USART9   | USART9 reset control  | <ul style="list-style-type: none"> <li>0b0: Function reset enabled</li> <li>0b1: Function reset released</li> </ul> | RW     |
| 27    | USART8   | USART8 reset control  | <ul style="list-style-type: none"> <li>0b0: Function reset enabled</li> <li>0b1: Function reset released</li> </ul> | RW     |
| 26    | USART7   | USART7 reset control  | <ul style="list-style-type: none"> <li>0b0: Function reset enabled</li> <li>0b1: Function reset released</li> </ul> | RW     |
| 25    | USART6   | USART6 reset control  | <ul style="list-style-type: none"> <li>0b0: Function reset enabled</li> <li>0b1: Function reset released</li> </ul> | RW     |
| 23    | USART4   | USART4 reset control  | <ul style="list-style-type: none"> <li>0b0: Function reset enabled</li> <li>0b1: Function reset released</li> </ul> | RW     |
| 22    | USART3   | USART3 reset control  | <ul style="list-style-type: none"> <li>0b0: Function reset enabled</li> <li>0b1: Function reset released</li> </ul> | RW     |
| 21    | USART2   | USART2 reset control  | <ul style="list-style-type: none"> <li>0b0: Function reset enabled</li> <li>0b1: Function reset released</li> </ul> | RW     |
| 20    | USART1   | USART1 reset control  | <ul style="list-style-type: none"> <li>0b0: Function reset enabled</li> <li>0b1: Function reset released</li> </ul> | RW     |
| 19    | Reserved | Reserved              | Reads as 1, Write as 1                                                                                              | -      |
| 18    | NFC      | NFC reset control     | <ul style="list-style-type: none"> <li>0b0: Function reset enabled</li> <li>0b1: Function reset released</li> </ul> | RW     |
| 17    | DMC      | DMC reset control     | <ul style="list-style-type: none"> <li>0b0: Function reset enabled</li> <li>0b1: Function reset released</li> </ul> | RW     |
| 16    | SMC      | SMC reset control     | <ul style="list-style-type: none"> <li>0b0: Function reset enabled</li> <li>0b1: Function reset released</li> </ul> | RW     |
| 15    | DVP      | DVP reset control     | <ul style="list-style-type: none"> <li>0b0: Function reset enabled</li> <li>0b1: Function reset released</li> </ul> | RW     |

| Bit   | Name            | Function             | Description                                                                                                             | Access |
|-------|-----------------|----------------------|-------------------------------------------------------------------------------------------------------------------------|--------|
| 14:13 | <i>Reserved</i> | <i>Reserved</i>      | <i>All bits read as 1, Write as 1</i>                                                                                   | -      |
| 12    | OTS             | OTS reset control    | <ul style="list-style-type: none"> <li>● 0b0: Function reset enabled</li> <li>● 0b1: Function reset released</li> </ul> | RW     |
| 11    | TMR0_5          | TMR0_5 reset control | <ul style="list-style-type: none"> <li>● 0b0: Function reset enabled</li> <li>● 0b1: Function reset released</li> </ul> | RW     |
| 10    | TMR0_4          | TMR0_4 reset control | <ul style="list-style-type: none"> <li>● 0b0: Function reset enabled</li> <li>● 0b1: Function reset released</li> </ul> | RW     |
| 9     | CMP34           | CMP34 reset control  | <ul style="list-style-type: none"> <li>● 0b0: Function reset enabled</li> <li>● 0b1: Function reset released</li> </ul> | RW     |
| 8     | CMP12           | CMP12 reset control  | <ul style="list-style-type: none"> <li>● 0b0: Function reset enabled</li> <li>● 0b1: Function reset released</li> </ul> | RW     |
| 7:6   | <i>Reserved</i> | <i>Reserved</i>      | <i>All bits read as 1, Write as 1</i>                                                                                   | -      |
| 5     | DAC2            | DAC2 reset control   | <ul style="list-style-type: none"> <li>● 0b0: Function reset enabled</li> <li>● 0b1: Function reset released</li> </ul> | RW     |
| 4     | DAC1            | DAC1 reset control   | <ul style="list-style-type: none"> <li>● 0b0: Function reset enabled</li> <li>● 0b1: Function reset released</li> </ul> | RW     |
| 3     | <i>Reserved</i> | <i>Reserved</i>      | <i>Reads as 1, Write as 1</i>                                                                                           | -      |
| 2     | ADC3            | ADC3 reset control   | <ul style="list-style-type: none"> <li>● 0b0: Function reset enabled</li> <li>● 0b1: Function reset released</li> </ul> | RW     |
| 1     | ADC2            | ADC2 reset control   | <ul style="list-style-type: none"> <li>● 0b0: Function reset enabled</li> <li>● 0b1: Function reset released</li> </ul> | RW     |
| 0     | ADC1            | ADC1 reset control   | <ul style="list-style-type: none"> <li>● 0b0: Function reset enabled</li> <li>● 0b1: Function reset released</li> </ul> | RW     |

**Note**

The reset of the following registers is not controlled by the functional reset control register RMU\_FRST0/1/2/3: USB\_SYCTLREG, SDIOC\_SYCTRLREG, NFC\_STCR, NFC\_STSR, EXMC\_ENAR, USART1\_NFC, TMR\_SYNNR1, TMR\_SYNNR2, CAN\_SYCTRLREG, ETH\_MAC\_IFCONFR

## 4.3.6 Reset Control Register 0 (RMU\_PRSTCR0)

| Offset | Bit Position |   |         |          |   |   |   |   |
|--------|--------------|---|---------|----------|---|---|---|---|
| 0xF8   | 7            | 6 | 5       | 4        | 3 | 2 | 1 | 0 |
| Reset  | 0x40         |   |         |          |   |   |   |   |
| Name   | Reserved     |   | LKUPREN | Reserved |   |   |   |   |
| Access |              |   | RW      |          |   |   |   |   |

| Bit | Name     | Function                      | Description                                                                                                                       | Access |
|-----|----------|-------------------------------|-----------------------------------------------------------------------------------------------------------------------------------|--------|
| 7   | Reserved | Reserved                      | -                                                                                                                                 | -      |
| 6   | Reserved | Reserved                      | Reads as 1, Write as 1                                                                                                            | -      |
| 5   | LKUPREN  | Cortex-M4 Lockup reset enable | <ul style="list-style-type: none"> <li>0b0: Cortex-M4 Lockup reset disable</li> <li>0b1: Cortex-M4 Lockup reset enable</li> </ul> | RW     |
| 4:0 | Reserved | Reserved                      | -                                                                                                                                 | -      |

## 4.3.7 Reset Flag Register 0 (RMU\_RSTF0)

| Offset | Bit Position                                                      |         |          |    |    |    |    |    |    |    |    |    |    |    |    |    |        |        |        |         |        |        |          |        |      |      |       |      |        |        |      |       |      |    |    |    |    |
|--------|-------------------------------------------------------------------|---------|----------|----|----|----|----|----|----|----|----|----|----|----|----|----|--------|--------|--------|---------|--------|--------|----------|--------|------|------|-------|------|--------|--------|------|-------|------|----|----|----|----|
| 0xFC   | 31                                                                | 30      | 29       | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15     | 14     | 13     | 12      | 11     | 10     | 9        | 8      | 7    | 6    | 5     | 4    | 3      | 2      | 1    | 0     |      |    |    |    |    |
| Reset  | 0xFFFFFFFF (the reset value varies depending on the reset method) |         |          |    |    |    |    |    |    |    |    |    |    |    |    |    |        |        |        |         |        |        |          |        |      |      |       |      |        |        |      |       |      |    |    |    |    |
| Name   | CLRf                                                              | MULTIRf | Reserved |    |    |    |    |    |    |    |    |    |    |    |    |    | ERMURf | FLECRf | LKUPRf | XTALERf | CKFERf | RAECRf | Reserved | MPUERf | SWRf | PDRf | SWDRf | WDRf | PVD2RF | PVDIRf | BORf | PINRf | PORf |    |    |    |    |
| Access | RW                                                                | RW      |          |    |    |    |    |    |    |    |    |    |    |    |    |    | RW     | RW     | RW     | RW      | RW     | RW     |          | RW     | RW   | RW   | RW    | RW   | RW     | RW     | RW   | RW    | RW   | RW | RW | RW | RW |

| Bit | Name            | Function                         | Description                                                                                                                                                                                                | Access |
|-----|-----------------|----------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 13  | XTALERF         | XTAL error reset flag            | Set by hardware when a XTAL error reset occurs.<br>Clear by setting CLRF.<br>Read:<br><ul style="list-style-type: none"> <li>● 0b0: No reset occurred</li> <li>● 0b1: Reset occurred</li> </ul>            | RW     |
| 12  | CKFERF          | Clock frequency error reset flag | Set by hardware when a clock frequency error reset occurs.<br>Clear by setting CLRF.<br>Read:<br><ul style="list-style-type: none"> <li>● 0b0: No reset occurred</li> <li>● 0b1: Reset occurred</li> </ul> | RW     |
| 11  | RAECRF          | RAM ECC reset flag               | Set by hardware when a RAM ECC reset occurs.<br>Clear by setting CLRF.<br>Read:<br><ul style="list-style-type: none"> <li>● 0b0: No reset occurred</li> <li>● 0b1: Reset occurred</li> </ul>               | RW     |
| 10  | <i>Reserved</i> | <i>Reserved</i>                  | -                                                                                                                                                                                                          | -      |
| 9   | MPUERF          | MPU error reset flag             | Set by hardware when an MPU error reset occurs.<br>Clear by setting CLRF.<br>Read:<br><ul style="list-style-type: none"> <li>● 0b0: No reset occurred</li> <li>● 0b1: Reset occurred</li> </ul>            | RW     |
| 8   | SWRF            | Software reset flag              | Set by hardware when a software reset occurs.<br>Clear by setting CLRF.<br>Read:<br><ul style="list-style-type: none"> <li>● 0b0: No reset occurred</li> <li>● 0b1: Reset occurred</li> </ul>              | RW     |
| 7   | PDRF            | Power-down mode reset            | Set by hardware when a power down reset occurs.<br>Clear by setting CLRF.<br>Read:<br><ul style="list-style-type: none"> <li>● 0b0: No reset occurred</li> <li>● 0b1: Reset occurred</li> </ul>            | RW     |
| 6   | SWDRF           | Special watchdog reset flag      | Set by hardware when a special watchdog reset occurs.<br>Clear by setting CLRF.<br>Read:<br><ul style="list-style-type: none"> <li>● 0b0: No reset occurred</li> <li>● 0b1: Reset occurred</li> </ul>      | RW     |

| Bit | Name   | Function                                    | Description                                                                                                                                                                                                           | Access |
|-----|--------|---------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 5   | WDRF   | Watchdog reset flag                         | Set by hardware when a watchdog reset occurs.<br>Clear by setting CLRF.<br>Read:<br><ul style="list-style-type: none"> <li>● 0b0: No reset occurred</li> <li>● 0b1: Reset occurred</li> </ul>                         | RW     |
| 4   | PVD2RF | Programmable voltage detection 2 reset flag | Set by hardware when a programmable voltage detection 2 reset occurs.<br>Clear by setting CLRF.<br>Read:<br><ul style="list-style-type: none"> <li>● 0b0: No reset occurred</li> <li>● 0b1: Reset occurred</li> </ul> | RW     |
| 3   | PVD1RF | Programmable voltage detection 1 reset flag | Set by hardware when a programmable voltage detection 1 reset occurs.<br>Clear by setting CLRF.<br>Read:<br><ul style="list-style-type: none"> <li>● 0b0: No reset occurred</li> <li>● 0b1: Reset occurred</li> </ul> | RW     |
| 2   | BORF   | Brown-out reset flag                        | Set by hardware when a brown-out reset occurs.<br>Clear by setting CLRF.<br>Read:<br><ul style="list-style-type: none"> <li>● 0b0: No reset occurred</li> <li>● 0b1: Reset occurred</li> </ul>                        | RW     |
| 1   | PINRF  | NRST pin reset flag                         | Set by hardware when a pin reset occurs.<br>Clear by setting CLRF.<br>Read:<br><ul style="list-style-type: none"> <li>● 0b0: No reset occurred</li> <li>● 0b1: Reset occurred</li> </ul>                              | RW     |
| 0   | PORF   | Power-on reset flag                         | Set by hardware when a power-on reset occurs.<br>Clear by setting CLRF.<br>Read:<br><ul style="list-style-type: none"> <li>● 0b0: No reset occurred</li> <li>● 0b1: Reset occurred</li> </ul>                         | RW     |

## 5 Clock Module Unit (CMU)

### 5.1 Introduction

The clock control unit provides a series of clock functions with different frequencies, including: an External High-speed Oscillator, an External Low-speed Oscillator, two PLL clocks, an Internal High-speed Oscillator, an Internal Medium-speed Oscillator, an Internal Low-speed Oscillator, a RTC Dedicated Internal Low-speed Oscillator, a dedicated Internal Low-speed Oscillator for SWDT, a Clock Prescaler, Clock Multiplexing, and Clock Gating Circuit.

The clock control unit also provides a clock frequency measurement function. The clock frequency measurement circuit (FCM) monitors and measures the measurement target clock using the measurement reference clock. An interrupt or reset occurs when the setting range is exceeded.

AHB, APB, and Cortex-M4 clocks are derived from the System Clock. The maximum operating clock frequency of the System Clock can reach 240MHz, with 6 selectable clock sources:

1. External High-speed Oscillator (XTAL)
2. External Low-speed Oscillator (XTAL32)
3. PLLH Clock (PLLH)
4. Internal High-speed Oscillator (HRC)
5. Internal Medium-speed Oscillator (MRC)
6. Internal Low-speed Oscillator (LRC)

For each clock source, it can be individually turned on and off when not in use to reduce power consumption.

SWDT has an independent clock source: the Dedicated Internal Low-speed Oscillator for SWDT (SWDTLRC). The Real Time Clock (RTC) uses the External Low-speed Oscillator, the RTC Dedicated Internal Low-speed Oscillator, or the XTAL fractional frequency division clock as the clock source. The 48MHz clock for USBFS can select the System Clock, PLLH, or PLLA as the clock source.

### 5.2 Features

Table 5-1 Clock Source Main Features

| Clock Source                           | Specifications                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|----------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| External High-speed Oscillator (XTAL)  | Crystal frequency range: 4-25MHz<br>External clock input: Up to 25MHz<br>Oscillator fault detection function                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| External Low-speed Oscillator (XTAL32) | Crystal frequency: 32.768kHz                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| PLLH Clock                             | Input Clock: External high-speed oscillator or internal high-speed oscillator<br>PLLH input clock frequency division: 1-4 arbitrary frequency division optional<br>PFD input clock frequency=input clock/PLLH input clock divided by frequency, frequency range 8MHz-25MHz<br>PLLH frequency multiplication coefficient: 25-150 times<br>VCO oscillation frequency: 600MHz-1200MHz<br>PLLHP output division ratio: arbitrary division from 2-16<br>PLLHQ output division ratio: arbitrary division from 2-16<br>PLLHR output division ratio: arbitrary division from 2-16 |

| Clock Source                                             | Specifications                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|----------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| PLLH Clock                                               | $\text{PLLHP output frequency} = (\text{input clock} / \text{PLLH input clock frequency division}) * \text{PLLH frequency multiplication factor} / \text{PLLHP output frequency division ratio}$<br>$\text{PLLHQ output frequency} = (\text{input clock} / \text{PLLH input clock frequency division}) * \text{PLLH frequency multiplication factor} / \text{PLLHQ output frequency division ratio}$<br>$\text{PLLHR output frequency} = (\text{input clock} / \text{PLLH input clock frequency division}) * \text{PLLH frequency multiplication factor} / \text{PLLHR output frequency division ratio}$                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| PLLA Clock                                               | Input clock: External high-speed oscillator or internal high-speed oscillator<br>PLLA input clock frequency division: 1-25 arbitrary frequency division optional<br>PFD Input clock frequency = input clock / PLLA Input clock division, with a frequency range of 1MHz-25MHz<br>PLLA Frequency multiplication coefficient: 20-480 times<br>VCO Oscillation frequency: 240MHz-480MHz<br>PLLAP Output division ratio: arbitrary division from 2-16<br>PLLAQ Output division ratio: arbitrary division from 2-16<br>PLLAR Output division Ratio: arbitrary division from 2-16<br>$\text{PLLAP output frequency} = (\text{input clock} / \text{PLLA input clock frequency division}) * \text{PLLA frequency multiplication factor} / \text{PLLAP output frequency division ratio}$<br>$\text{PLLAQ output frequency} = (\text{input clock} / \text{PLLA input clock frequency division}) * \text{PLLA frequency multiplication factor} / \text{PLLAQ output frequency division ratio}$<br>$\text{PLLAR output frequency} = (\text{input clock} / \text{PLLA input clock frequency division}) * \text{PLLA frequency multiplication factor} / \text{PLLAR output frequency division ratio}$ |
| Internal High-speed Oscillator (HRC)                     | Frequency: 16MHz or 20MHz<br>User-writable registers for frequency trimming                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| Internal Medium-speed Oscillator (MRC)                   | Frequency: 8MHz<br>User-writable registers for frequency trimming                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| Internal Low-speed Oscillator (LRC)                      | Frequency: 32.768kHz<br>User-writable registers for frequency trimming                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| Dedicated Internal Low Speed Oscillator for RTC (RTCLRC) | Frequency: 32.768kHz<br>User-writable registers for frequency trimming                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| SWDT Dedicated Internal Low-speed Oscillator (SWDTLRC)   | Frequency: 10kHz                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |



## 5.3 Functional Description

### 5.3.1 System Block Diagram

#### 5.3.1.1 Clock System Block Diagram

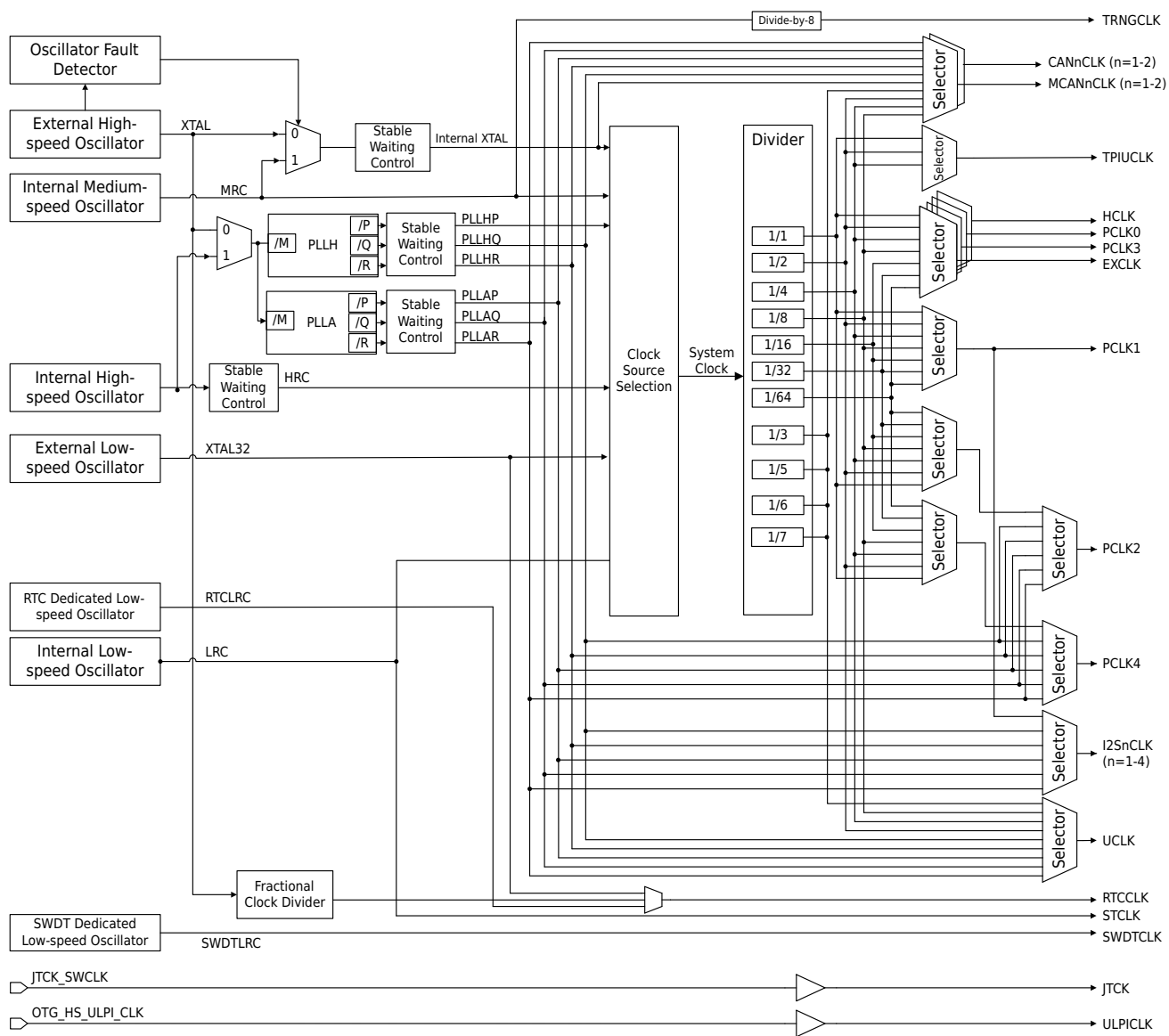


Figure 5-1 Clock System Block Diagram

### 5.3.1.2 Clock Frequency Measurement Block Diagram

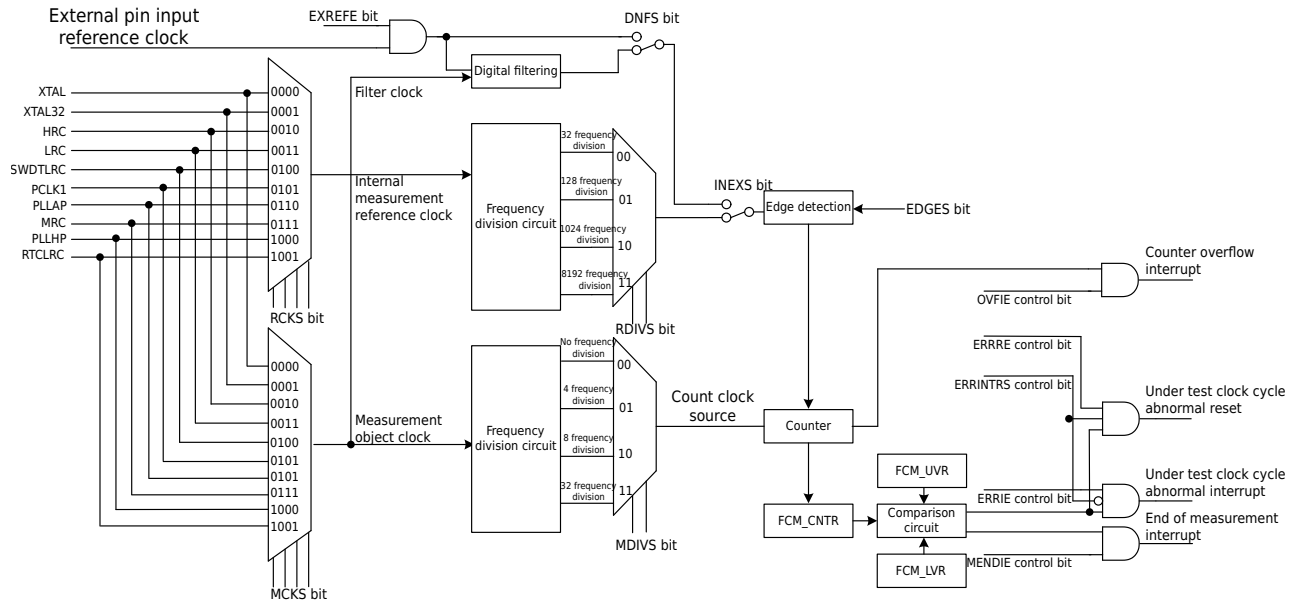


Figure 5-2 Clock Frequency Measurement Block Diagram

### 5.3.2 Working Clock Specifications

Table 5-2 Specifications for Each Internal Clock

| Clock | Scope of Action                                                                                                                                                                                                                                                          | Specifications                                                                                                                                                                                                                                    |
|-------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| HCLK  | CPU, DMan (n=1-2), EFM, SRAMn (n=1-4), SRAMH, Ret SRAM, MPU, GPIO, DCUn (n=1-8), INTC, QSPI, DVP, KEYSKAN, FMACn (n=1-4), MAU                                                                                                                                            | Maximum frequency 240MHz.<br>Configured by the CMU_SCFGR.HCLKS bit.<br>Selectable system clock division: 1, 2, 4, 8, 16, 32, 64.                                                                                                                  |
| PCLK0 | Timer6_n (n=1-8) counters, Timer4_n (n=1-3), TimerA_n (n=1-4)                                                                                                                                                                                                            | Maximum frequency 240MHz.<br>Configured by the CMU_SCFGR.PCLK0S bit.<br>Selectable system clock division: 1, 2, 4, 8, 16, 32, 64.                                                                                                                 |
| PCLK1 | USBHS/USBFS control logic, USARTn (n=1-10), SPIn (n=1-6), Timer0_n (n=1-5), Timer2_n (n=1-4), TimerA_n (n=5-12), Timer6_n (n=1-8) control logic, EMB, CRC, HASH, I2Sn (n=1-4), SDIOcn (n=1-2), ETHMAC, CANn (n=1-2) control logic, MCANn (n=1-2) control logic, SKE, PKE | Maximum frequency 120MHz .<br>Configured by the CMU_SCFGR.PCLK1S bit.<br>Selectable system clock division: 1, 2, 4, 8, 16, 32, 64.                                                                                                                |
| PCLK2 | ADC Conversion Clock                                                                                                                                                                                                                                                     | Maximum frequency 60 MHz.<br>Configured by the CMU_SCFGR.PCLK2S bit and the CMU_PERICKSEL.PERICKSEL bit.<br>Selectable system clock division: 1, 2, 4, 8, 16, 32, 64.<br>Selectable independent clock sources: PLLAP, PLLAQ, PLLAR, PLLHQ, PLLHR. |

| Clock    | Scope of Action                                                                                                            | Specifications                                                                                                                                                                                                                                             |
|----------|----------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| PCLK3    | RTC (control logic) , I2Cn (n=1-6), CMPn (n=1-2), WDT, SWDT control logic, WKTM, OTS, FCM, VBAT backup register, CTC, ERMU | Maximum frequency: 60MHz when the power supply voltage is 2.0V or higher.<br>Maximum Frequency: 30MHz when the power supply voltage is below 2.0V.<br>Configured by the CMU_SCFGR.PCLK3S bit.<br>Selectable system clock division: 1, 2, 4, 8, 16, 32, 64. |
| PCLK4    | ADCn (n=1-3) (control logic), DACn (n=1-2) (control logic), TRNG                                                           | Maximum frequency 120MHz.<br>Configured by the CMU_SCFGR.PCLK4S bit and the CMU_PERICKSEL.PERICKSEL bit.<br>Selectable system clock division: 1, 2, 4, 8, 16, 32, 64.<br>Selectable independent clock sources: PLLAP, PLLAQ, PLLAR, PLLHQ, PLLHR.          |
| EXCLK    | NFC, DMC, SMC                                                                                                              | Maximum frequency 120MHz.<br>Configured by the CMU_SCFGR.EXCKS bit.<br>Selectable system clock division: 1, 2, 4, 8, 16, 32, 64.                                                                                                                           |
| UCLK     | USBFS/USBHS FullSpeed Communication Clock                                                                                  | Frequency 48MHz<br>Configured by the CMU_USBCKCFGR.USBCKS bit.<br>Selectable system clock division: 2, 3, 4, 5, 6, 7, 8.<br>Selectable independent clock sources: PLLAP, PLLAQ, PLLAR, PLLHQ, PLLHR.                                                       |
| CANnCLK  | CANn (n=1-2) Communication Clock                                                                                           | Maximum frequency 80 MHz.<br>Configured by the CMU_CANCKCFGR.CANnCKS (n=1-2) bit.<br>Selectable system clock division: 2, 3, 4, 5, 6, 7, 8.<br>Selectable independent clock sources: PLLAP, PLLAQ, PLLAR, PLLHQ, PLLHR, XTAL.                              |
| MCANnCLK | MCANn (n=1-2) Communication Clock                                                                                          | Maximum frequency 80 MHz.<br>Configured by the CMU_CANCKCFGR.MCANnCKS (n=1-2) bit.<br>Selectable system clock division: 2, 3, 4, 5, 6, 7, 8.<br>Selectable independent clock sources: PLLAP, PLLAQ, PLLAR, PLLHQ, PLLHR, XTAL.                             |
| STCLK    | SYSTICK Timer External Reference Clock                                                                                     | Internal low speed oscillator clock.<br>Configured by the CLKSOURCE bit in the SYSTICK Control and Status Register of the CPU.                                                                                                                             |
| SWDTCLK  | SWDT Counter Clock                                                                                                         | Frequency 10kHz                                                                                                                                                                                                                                            |
| TCK      | JTAG Clock                                                                                                                 | Maximum frequency 25MHz.                                                                                                                                                                                                                                   |
| TPIUCLK  | Cortex-M4 Debug Tracker Clock                                                                                              | Maximum frequency 60MHz.<br>Configured by the CMU_TPIUCKCFGR.TPIUCKS bit.<br>Selectable system clock division: 1, 2, 4.                                                                                                                                    |
| ULPICKL  | USBHS HighSpeed Communication Clock                                                                                        | Frequency 60 MHz                                                                                                                                                                                                                                           |

| Clock   | Scope of Action   | Specifications                                                                                                                        |
|---------|-------------------|---------------------------------------------------------------------------------------------------------------------------------------|
| I2SnCLK | I2Sn (n=1-4)      | Maximum frequency 240MHz.<br>Configured by the CMU_I2SCKSEL register.<br>Selectable clocks: PLLAP, PLLAQ, PLLAR, PLLHQ, PLLHR, PCLK1. |
| TRNGCLK | TRNG Action Clock | The frequency is 1MHz.<br>Fixed at 8x division of MRC. Disable stopping the MRC clock when using the TRNG function.                   |

The following rules must be followed between clocks:

1. HCLK frequency  $\geq$  PCLK2, PCLK3, PCLK4 frequency
2. ETHMAC:
  - ETHMAC MII mode:
    - ▶ PCLK1 frequency > 2 times MII\_TX\_CLK/MII\_RX\_CLK frequency
    - ▶ HCLK frequency > PCLK1 frequency
  - ETHMAC RMII mode:
    - ▶ PCLK1 frequency > RMII\_REF\_CLK frequency
    - ▶ HCLK frequency > PCLK1 frequency
  - When ETHMAC is not in use:
    - ▶ HCLK frequency  $\geq$  PCLK1 frequency
3. NFC:
  - When NFC is in use: HCLK frequency > PCLK1 frequency
  - When NFC is not in use: HCLK frequency  $\geq$  PCLK1 frequency
4. PCLK0 frequency  $\geq$  PCLK1 frequency, PCLK0 frequency  $\geq$  PCLK3 frequency
5. HCLK frequency: PCLK0 frequency = N: 1, 1: N
6. PCLK2 frequency: PCLK4 frequency = 1: 8, 1: 4, 1: 2, 1: 1, 2: 1, 4: 1, 8: 1

### 5.3.3 Crystal Oscillator Circuit

#### 5.3.3.1 External High-speed Oscillator

##### 5.3.3.1.1 Oscillator Mode

External high-speed oscillator provides a more accurate timer for the system clock.

XTAL is turned on and off through the CMU\_XTALCR.XTALSTP bit.

The CMU\_OSCSTBSR.XTALSTBF bit indicates whether the external high-speed oscillator is stable or not. The stability time is configured through register CMU\_XTALSTBCR. CMU\_XTALSTBCR must be set at a stable time greater than or equal to that required by the crystal oscillator manufacturer.

The circuit constant of the crystal oscillator varies with the parasitic capacitance of the crystal oscillator and the installed circuit, so it must be discussed carefully with the crystal oscillator manufacturer.

Oscillator features are closely associated with the user's circuit board design, and crystal oscillator and load capacitance must be as close to the oscillator pin as possible to minimize output distortion and vibration stabilization time. The load capacitance value must be adjusted appropriately depending on the oscillator selected. It is not possible to pass the signal line near the oscillating circuit, otherwise it may not be able to oscillate normally due to inductance.

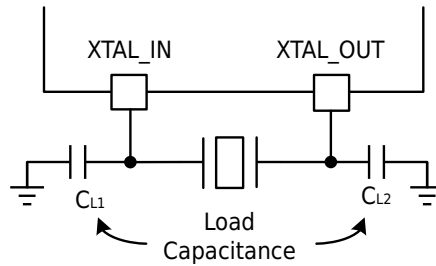


Figure 5-3 External High Speed Oscillator Connection Example

#### 5.3.3.1.2 Clock Input Mode

An external clock source must be provided in clock input mode. This mode is selected by setting the CMU\_XTALCFGR.XTALMS to 1 and the CMU\_XTALCR.XTALSTP to 0. An external clock signal with a duty period of approximately 50% must be used to drive the XTAL\_OUT/XTAL\_EXT pins. At this time, the XTAL\_IN pin can be configured as GPIO according to register settings.

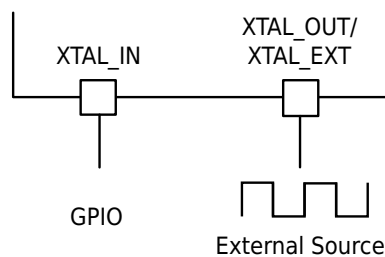


Figure 5-4 External Clock Input Connection Example

#### 5.3.3.2 External High-speed Oscillator Fault Detection

Oscillator fault detection is used to detect whether the oscillation of the external high-speed oscillator (XTAL) is normal.

Open or close by register CMU\_XTALSTDCR.XTALSTDE bit.

After reset release, the external high-speed oscillator stops oscillating, and the external high-speed oscillator fault detection function is disabled. To enable the external high-speed oscillator fault detection function, the external high-speed oscillator must be oscillating, and after waiting for the external high-speed oscillator to stabilize (CMU\_OSCSTBSR.XTALSTBF is 1), it can be turned on through the CMU\_XTALSTDCR.XTALSTDE register.

PLLH and PLLA select XTAL clock as the input source, they can only select the XTAL oscillation fault to generate reset function.

Since oscillator fault detection detects abnormal oscillation of the oscillator caused by external factors, the oscillator oscillation fault detection function should be disabled when stopping the external high-speed oscillator through software or transitioning to Stop mode and and Power-down mode.

If the external high-speed oscillator fails, the action waveform is shown in the following figure. For the operation procedure, please refer to "[XTAL Fault Detection Action Detected](#)".

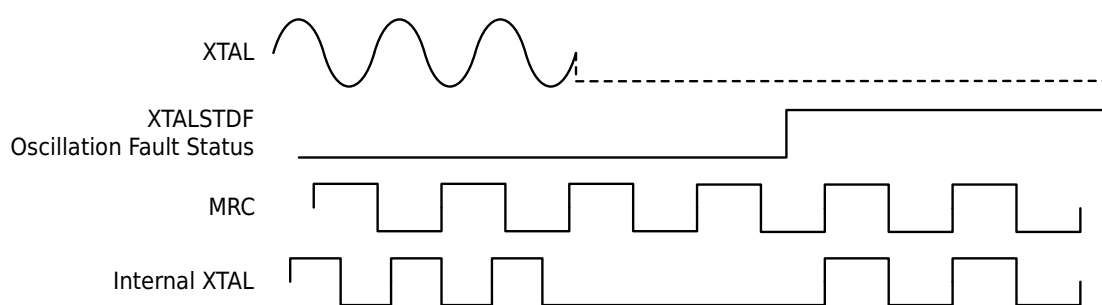


Figure 5-5 External High-speed Oscillator Fault Detection Example

#### 5.3.3.2.1 Action upon Detection of XTAL Fault

When an external high-speed oscillator oscillation fault is detected, if the system clock selects the external high-speed oscillator as the system clock, the system clock will automatically switch to MRC.

When an external high-speed oscillator oscillation fault is detected, it can trigger EMB to set the PWM output of Timer6/Timer4 to Hi-Z output. For more information, please refer to the relevant description in the "Emergency Brake Module (EMB) " chapter.

When the system clock is selected as XTAL and an XTAL oscillation fault is detected, the action example is as follows:

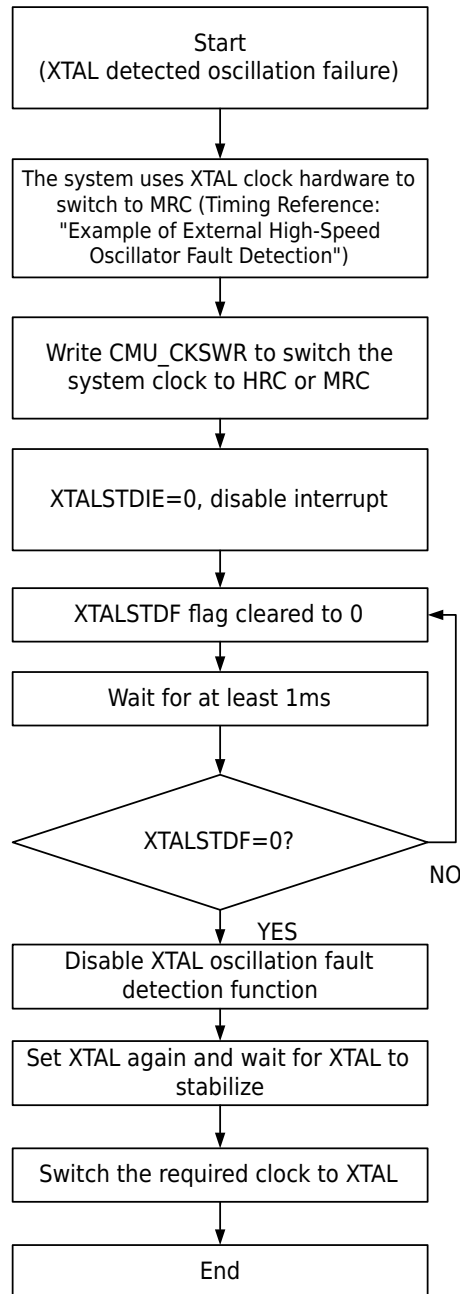


Figure 5-6 System Clock Selected as XTAL, Example of Detecting XTAL Oscillation Fault

### 5.3.3.2.2 Interrupt Reset Generated upon Detection of XTAL Oscillation Fault

The XTAL oscillation fault interrupt can be configured as a maskable interrupt or a non-maskable interrupt. For more information, please refer to the "Interrupt Controller (INTC) " chapter.

When the XTAL oscillation fault is configured as a reset, if an XTAL oscillation fault is detected, the chip will generate a reset. Please refer to the "Reset Module Unit (RMU) " chapter for reset actions.

### 5.3.3.3 External High-speed Oscillator Fractional Division

By setting the CMU\_XTALDIVR register, the external high-speed oscillator clock can be divided down to 32.768kHz for use by the RTC.

Example setting for dividing the external high-speed oscillator from 10MHz to 32.768kHz:

$$\text{Division number} = \frac{10000000\text{Hz}}{32768\text{Hz}} = \frac{78125(\text{Decimal})}{256(\text{Decimal})} = \frac{0x1312D(\text{Hexadecimal})}{0x100(\text{Hexadecimal})}$$

Set the NUMER[16:0] register to 0x1312D and the DENOM[10:0] register to 0x100. In other words, set the CMU\_XTALDIVR register to 0x1312D100, and then initiate the process via the CMU\_XTALDIVCR register.

Example setting for dividing the external high-speed oscillator from 8MHz to 32.768kHz:

$$\text{Division number} = \frac{8000000\text{Hz}}{32768\text{Hz}} = \frac{31250(\text{Decimal})}{128(\text{Decimal})} = \frac{0x7A12(\text{Hexadecimal})}{0x80(\text{Hexadecimal})}$$

Set the NUMER[16:0] register to 0x07A12 and the DENOM[10:0] register to 0x080. In other words, set the CMU\_XTALDIVR register to 0x07A12080, and then initiate the process via the CMU\_XTALDIVCR register.

### 5.3.3.4 External Low-speed Oscillator

The 32.768kHz external low-speed oscillator can provide a more precise clock source for the system clock and real time clock (RTC) circuit. It has the advantages of low power consumption and high precision.

XTAL32 is turned on and off by the CMU\_XTAL32CR.XTAL32STP bit.

The circuit constant of the crystal oscillator varies with the parasitic capacitance of the crystal oscillator and the installed circuit, so it must be discussed carefully with the crystal oscillator manufacturer. Oscillator features are closely associated with the user's circuit board design, and crystal oscillator and load capacitance must be as close to the oscillator pin as possible to minimize output distortion and vibration stabilization time. The load capacitance value must be properly adjusted according to the selected drive capability. It is not possible to pass the signal line near the oscillating circuit, otherwise it may not be able to oscillate normally due to inductance.

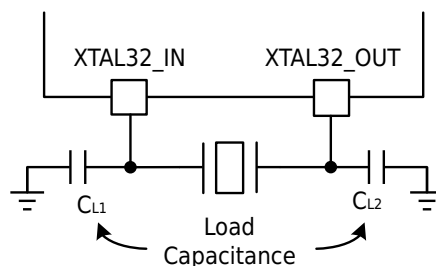


Figure 5-7 External Low-speed Oscillator Connection Example

The initialization procedure for XTAL32 upon initial power-on is as follows:

1. Write 1 to the CMU\_XTAL32CR.XTAL32STP bit to stop XTAL32;
2. Set the matching XTAL32 drive capability through CMU\_XTAL32FGR;
3. Set the filter function through CMU\_XTAL32NFR;
4. Write 0 to the CMU\_XTAL32CR.XTAL32STP bit to oscillate XTAL32;
5. Software waits for XTAL32 to stabilize. The stabilization time can be found in the "Electrical Specifications" chapter in the Datasheet.

If the external low-speed oscillator is not used, set the CMU\_XTAL32CR.XTAL32STP bit to 1 to turn off the external low-speed oscillator.

## 5.3.4 Internal RC Clock

### 5.3.4.1 HRC Clock

The HRC clock signal is generated by an internal high-speed oscillator and can be used directly as a system clock, or as an PLLH, PLLA input. The frequency of HRC can be configured to 16MHz or 20MHz by ICG1.HRCFREQSEL.



The advantage of the HRC oscillator is its low cost (no external components are required). In addition, its startup speed is faster than that of the XTAL crystal oscillator. However, even after calibration, its accuracy is not as good as that of an external crystal oscillator.

### Frequency Calibration

Due to differences in production processes, the RC oscillator frequency of different chips varies. Therefore, each device is factory-calibrated to ensure accuracy. Please refer to the "Electrical Specifications-Operating Conditions-Internal Clock Source Characteristics-HRC" chapter of the Datasheet.

If the application is affected by temperature changes, this may also affect the speed of the RC oscillator. Users can fine-tune the HRC frequency through registers.

CMU\_OSCSTBSR.HRCSTBF indicates whether HRC is stable. At startup, hardware sets this bit to 1 before HRC can be used.

HRC can be turned on or off through CMU\_HRCCR.HRCSTP bit.

#### 5.3.4.2 MRC Clock

The MRC clock signal is generated by an internal 8MHz medium-speed oscillator and can be directly used as the system clock.

The advantage of the MRC oscillator is its fast startup speed.

### Frequency Calibration

Due to differences in production processes, the RC oscillator frequency of different chips varies. Therefore, each device is factory-calibrated to ensure accuracy. Please refer to the "Electrical Specifications-Operating Conditions-Internal Clock Source Characteristics-MRC" chapter in the Datasheet.

If the application is affected by temperature changes, this may also affect the speed of the RC oscillator. Users can fine-tune the MRC frequency through registers.

MRC can be turned on or off through the CMU\_MRCCR.MRCSTP bit.

The MRC clock can also be used as a backup clock source in case of XTAL crystal oscillator failure. Please refer to "[Action upon Detection of XTAL Fault](#)".



#### Notice

Disable stopping the MRC clock when using the TRNG module.

#### 5.3.4.3 LRC Clock

The LRC clock signal is generated by the 32.768kHz internal low-speed oscillator and can be used directly as the system clock. LRC can be used as a low-power clock source to keep running in Power-down mode and Stop mode for Timer0, Timer2 and KEYSKAN.

The LRC oscillator features a fast start-up speed.

### Frequency Calibration

Due to differences in production processes, the RC oscillator frequency of different chips varies. Therefore, each device is factory-calibrated to ensure accuracy. Please refer to the "Electrical Specifications-Operating Conditions-Internal Clock Source Characteristics-LRC" chapter in the Datasheet.

If the application is affected by voltage or temperature changes, this may also affect the speed of the RC oscillator. Users can fine-tune the LRC frequency through registers.

LRC can be turned on or off through the CMU\_LRCCR.LRCSTP bit.

#### 5.3.4.4 SWDTLRC Clock

The SWDTLRC clock signal is generated by the internal 10kHz low-speed oscillator dedicated to SWDT. If SWDT is already started through ICG settings, the dedicated internal low-speed oscillator for SWDT will be forcibly turned on and cannot be disabled.

Due to differences in production processes, the RC oscillator frequency of different chips varies. Therefore, each device is factory-calibrated to ensure accuracy. Please refer to the "Electrical Specifications-Operating Conditions-Internal Clock Source Characteristics-SWDTLRC" chapter in the Datasheet.

#### 5.3.4.5 RTCLRC Clock

The RTCLRC clock signal is generated by the internal 32.768kHz low-speed oscillator. RTCLRC can be used as a low-power clock source to keep running in Power-down mode and Stop mode for RTC and WKTM.

The startup speed of the RTCLRC oscillator is fast.

#### Frequency Calibration

Due to differences in production processes, the oscillator frequency of different chips varies. Therefore, each device is factory-calibrated to ensure accuracy. Please refer to the "Electrical Specifications-Operating Conditions-Internal Clock Source Characteristics-RTCLRC" chapter in the Datasheet.

If the application is affected by voltage or temperature changes, this may also affect the speed of the RC oscillator. Users can fine-tune the RTCLRC frequency through registers.

RTCLRC can be turned on or off through the LRCEN bit in the RTC\_CR3 control register.

#### 5.3.5 PLL Clock

This product is equipped with two PLLs (PLLH and PLLA) :

- Both PLLH and PLLA can select XTAL or HRC oscillator as the input clock, configured uniformly by the CMU\_PLLHCFGR.PLLSRC bit;
- PLLH has three output clocks: PLLHP, PLLHQ, PLLHR, with independently configurable division factors;
- PLLA has three output clocks: PLLAP, PLLAQ, PLLAR, with independently configurable division factors;
- The PLLHP clock can be used as the system clock;
- The PLLHQ, PLLHR, PLLAP, PLLAQ, PLLAR clocks can be used to generate USB, ADC, DAC, I2S, CAN, MCAN clocks.



#### Notice

1. Configure the PLL after the HRC or XTAL oscillator is stable.
2. Since PLL configuration parameters cannot be changed after the PLL is enabled, it is recommended to configure the PLL first and then enable it.
3. When entering Power-down mode and Stop mode, the PLL will be disabled by hardware.

#### 5.3.6 Clock Switching Steps

After system reset, the default system clock is MRC. Switch the clock source by setting the CMU\_CKSW register. Please refer to "[Clock Source Switch](#)" for switching steps. Clock sources can only be switched from one to another when the target clock source is stable.

The waiting period of Flash/SRAM should be configured correctly to prevent the system clock frequency from being larger than the maximum operating frequency of Flash/SRAM. For detailed configuration,

please refer to the "CPU Clock and Flash Read Time Relationship" section of the "Embedded Flash Memory (EFM) " and the "Built-in SRAM (SRAM) " chapter.

### Clock Source Switching

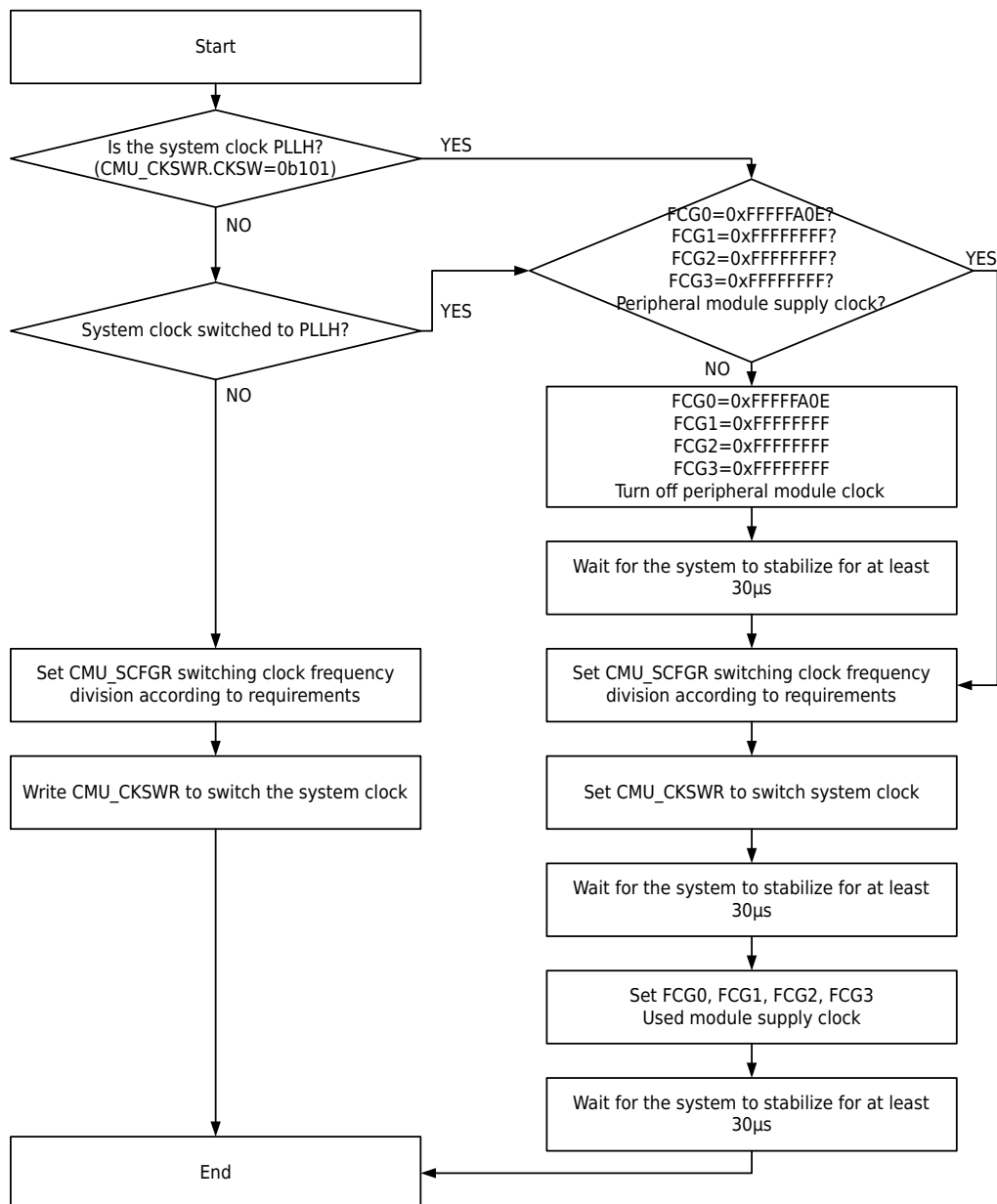


Figure 5-8 Clock Source Switching Flowchart

## Clock Division Switching

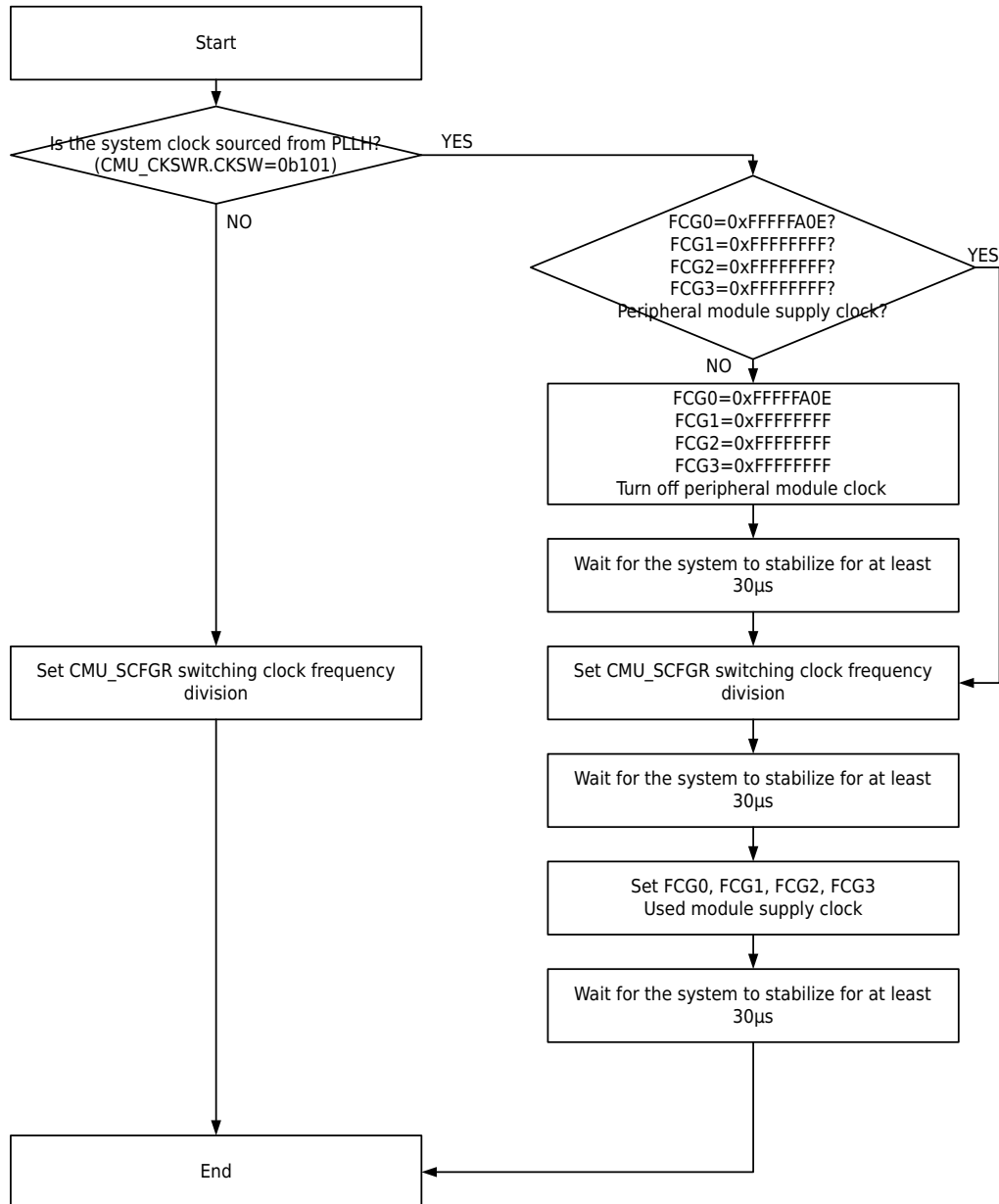


Figure 5-9 Clock Division Switching Flowchart

### 5.3.7 Clock Output Function

There are two clock outputs:

- MCO\_1

Users can output different clock sources to the MCO\_1 pin through a configurable pre-divider (1, 2, 4, 8,..., 128 division) :

- ▶ HRC Clock
- ▶ MRC Clock
- ▶ LRC Clock
- ▶ XTAL Clock
- ▶ XTAL32 Clock
- ▶ PLLHP/PLLHQ Clock

- ▶ PLLAP/PLLAQ/PLLAR Clock
- ▶ System Clock

The desired clock source is selected by the CMU\_MCO1CFGR.MCO1SEL bit.

#### ● MCO\_2

Users can output different clock sources to the MCO\_2 pin through a configurable pre-divider (1, 2, 4, 8,..., 128 division) :

- ▶ HRC Clock
- ▶ MRC Clock
- ▶ LRC Clock
- ▶ XTAL Clock
- ▶ XTAL32 Clock
- ▶ PLLHP/PLLHQ Clock
- ▶ PLLAP/PLLAQ/PLLAR Clock
- ▶ System Clock

The desired clock source is selected by the CMU\_MCO2CFGR.MCO2SEL bit.

The output clock of MCO\_1/MCO\_2 should not exceed 100MHz (maximum I/O speed).

### 5.3.8 Clock Frequency Measurement

#### 5.3.8.1 Clock Frequency Measurement Function

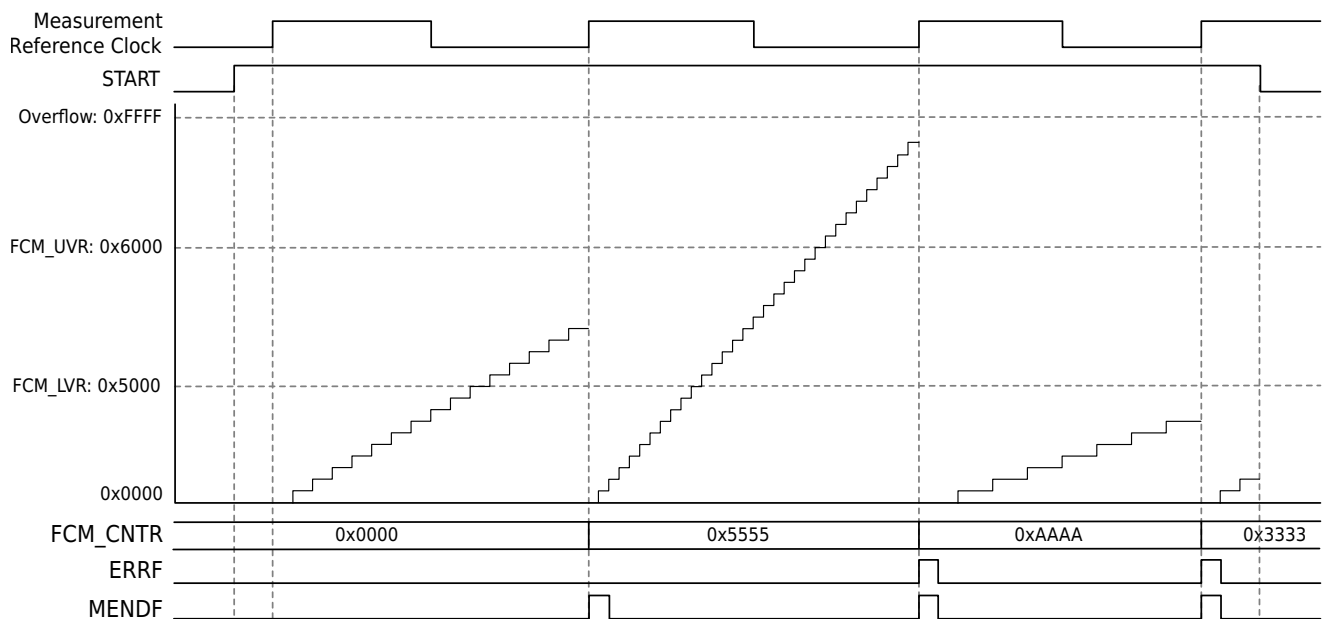


Figure 5-10 Clock Frequency Measurement Timing Diagram

- Select the reference clock and the clock to be measured, the clock division, and the active edge of the reference clock using FCM\_MCCR/FCM\_RCCR.
- After writing 1 to the FCM\_STR.START bit, the active edge selected by the EDGES bit is detected, and the counter starts counting up.
- When the active edge selected by the next EDGES bit of the reference clock is detected, the value of the counter is saved to the FCM\_CNTR register and compared with the set value of FCM\_LVR/FCM\_UVR. When  $\text{FCM\_LVR} \leq \text{FCM\_CNTR} \leq \text{FCM\_UVR}$ , the clock frequency measurement is normal.

When  $FCM\_LVR > FCM\_CNTR$  or  $FCM\_CNTR > FCM\_UVR$ , the clock frequency is abnormal, and an interrupt or reset can occur based on the `ERRINTRS/ERRRE/ERRIE` settings.

- After the `START` bit of `FCM_STR` is written to 0, the counter counting is stopped and cleared.
- The counter is divided into 4 groups, and each counter is independent of the others.
- When  $FCM\_LVR > FCM\_CNTR$  or  $FCM\_CNTR > FCM\_UVR$ , the clock frequency is abnormal, and the abnormal source is `FCM_FERR`.

#### 5.3.8.2 Digital Filter Function

External pin input reference clock `FCMREF` has digital filtering function. Digital filtering function performs 3 samples based on the sampling clock selected by the `DNFS` bit. If the levels of the 3 samples are the same, this level is sent internally.

The digital filtering function can set the digital filtering function to be active or inactive and the sampling clock.

#### 5.3.8.3 Interrupt/Reset Function

The clock frequency measurement circuit supports three interrupt requests:

- Frequency abnormal interrupt
- Frequency measurement end interrupt
- Counter overflow interrupt

The clock frequency measurement circuit supports one reset request:

- Frequency exceptionally reset

## 5.4 Register Description

### 5.4.1 Register List

FCM Base Address: 0x40048400

Table 5-3 Clock Frequency Measurement (FCM) Register Offset

| Offset          | Register  | Description                                  | Width |
|-----------------|-----------|----------------------------------------------|-------|
| 0x00+0x40*(n-1) | FCMn_LVR  | FCM Lower Limit Control Register (n=1-4)     | 32    |
| 0x04+0x40*(n-1) | FCMn_UVR  | FCM Upper Limit Control Register (n=1-4)     | 32    |
| 0x08+0x40*(n-1) | FCMn_CNTR | FCM Counter Register (n=1-4)                 | 32    |
| 0x0C            | FCM_STR   | FCM Start Control Register                   | 32    |
| 0x10+0x40*(n-1) | FCMn_MCCR | FCM Measurement Control Register (n=1-4)     | 32    |
| 0x14+0x40*(n-1) | FCMn_RCCR | FCM Reference Control Register (n=1-4)       | 32    |
| 0x18+0x40*(n-1) | FCMn_RIER | FCM Reset Interrupt Control Register (n=1-4) | 32    |
| 0x1C            | FCM_SR    | FCM Flag Register                            | 32    |
| 0x20+0x40*(n-1) | FCMn_CLR  | FCM Flag Clear Register (n=1-4)              | 32    |

CMU Base Address: 0x40048000

Table 5-4 Clock Module Unit (CMU) Register Offset

| Offset | Register       | Description                                             | Width |
|--------|----------------|---------------------------------------------------------|-------|
| 0x4400 | CMU_XTAL32CR   | CMU XTAL32 Control Register                             | 8     |
| 0x4404 | CMU_XTAL32CFGR | CMU XTAL32 Config Register                              | 8     |
| 0x4414 | CMU_XTAL32NFR  | CMU XTAL32 Filter Control Register                      | 8     |
| 0x441C | CMU_LRCCR      | CMU LRC Control Register                                | 8     |
| 0x4424 | CMU_LRCTRM     | CMU LRC Trimming Register                               | 8     |
| 0x442C | CMU_RTCLRCTRM  | CMU RTCLRC Trimming Register                            | 8     |
| 0x4C78 | CMU_XTALCFGR   | CMU XTAL Config Register                                | 8     |
| 0x0018 | CMU_XTALDIVR   | CMU XTAL Fractional Frequency Division Config Register  | 32    |
| 0x001C | CMU_XTALDIVCR  | CMU XTAL Fractional Frequency Division Control Register | 32    |
| 0xC0A2 | CMU_XTALSTBCR  | CMU XTAL Stable Control Register                        | 8     |
| 0xC032 | CMU_XTALCR     | CMU XTAL Control Register                               | 8     |
| 0xC040 | CMU_XTALSTDCR  | CMU XTAL Clock Stop Detect Control Register             | 8     |

| Offset | Register       | Description                                | Width |
|--------|----------------|--------------------------------------------|-------|
| 0xC041 | CMU_XTALSTDSR  | CMU XTAL Clock Stop Detect Status Register | 8     |
| 0xC036 | CMU_HRCCR      | CMU HRC Control Register                   | 8     |
| 0xC062 | CMU_HRCTRM     | CMU HRC Trimming Register                  | 8     |
| 0xC038 | CMU_MRCCR      | CMU MRC Control Register                   | 8     |
| 0xC061 | CMU_MRCTRM     | CMU MRC Trimming Register                  | 8     |
| 0xC100 | CMU_PLLHCFGR   | CMU PLLH Config Register                   | 32    |
| 0xC02A | CMU_PLLHCR     | CMU PLLH Control Register                  | 8     |
| 0xC104 | CMU_PLLACFGR   | CMU PLLA Config Register                   | 32    |
| 0xC02E | CMU_PLLACR     | CMU PLLA Control Register                  | 8     |
| 0xC03C | CMU_OSCSTBSR   | CMU Clock Source Stable Status Register    | 8     |
| 0xC026 | CMU_CKSWR      | CMU System Clock Source Switch Register    | 8     |
| 0xC020 | CMU_SCFGR      | CMU Clock Divider Config Register          | 32    |
| 0xC024 | CMU_USBCKCFGR  | CMU USB Clock Config Register              | 8     |
| 0xC018 | CMU_CANCKCFGR  | CMU CAN Clock Config Register              | 16    |
| 0xC010 | CMU_PERICKSEL  | CMU ADC/DAC Clock Config Register          | 16    |
| 0xC03F | CMU_TPIUCKCFGR | CMU Debug Clock Config Register            | 8     |
| 0xC012 | CMU_I2SCKCFGR  | CMU I2S Clock Config Register              | 16    |
| 0xC03D | CMU_MCO1CFGR   | CMU MCO1 Clock Output Config Register      | 8     |
| 0xC03E | CMU_MCO2CFGR   | CMU MCO2 Clock Output Config Register      | 8     |



## 5.4.2 CMU XTAL Config Register (CMU\_XTALCFGR)

| Offset | Bit Position |        |         |          |   |   |   |   |
|--------|--------------|--------|---------|----------|---|---|---|---|
| 0x4C78 | 7            | 6      | 5       | 4        | 3 | 2 | 1 | 0 |
| Reset  | 0x80         |        |         |          |   |   |   |   |
| Name   | Reserved     | XTALMS | XTALDRV | Reserved |   |   |   |   |
| Access |              | RW     | RW      |          |   |   |   |   |

| Bit | Name            | Function                        | Description                                                                                                                                                                                                                                                                                                      | Access |
|-----|-----------------|---------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 7   | <i>Reserved</i> | <i>Reserved</i>                 | <i>Reads as 1, Write as 1</i>                                                                                                                                                                                                                                                                                    | -      |
| 6   | XTALMS          | XTAL mode selection bit         | <ul style="list-style-type: none"> <li>0b0: Oscillator mode</li> <li>0b1: External clock input mode</li> </ul>                                                                                                                                                                                                   | RW     |
| 5:4 | XTALDRV         | XTAL drive capability selection | <ul style="list-style-type: none"> <li>0b00: High drive capability (20MHz &lt; Crystal ≤ 25MHz)</li> <li>0b01: Medium drive capability (16MHz &lt; Crystal ≤ 20MHz)</li> <li>0b10: Low drive capability (8MHz &lt; Crystal ≤ 16MHz)</li> <li>0b11: Ultra low drive capability (4MHz ≤ Crystal ≤ 8MHz)</li> </ul> | RW     |
| 3:0 | <i>Reserved</i> | <i>Reserved</i>                 | -                                                                                                                                                                                                                                                                                                                | -      |

## 5.4.3 CMU XTAL Stable Control Register (CMU\_XTALSTBCR)

| Offset | Bit Position |   |   |   |         |   |   |   |
|--------|--------------|---|---|---|---------|---|---|---|
| 0xC0A2 | 7            | 6 | 5 | 4 | 3       | 2 | 1 | 0 |
| Reset  | 0x05         |   |   |   |         |   |   |   |
| Name   | Reserved     |   |   |   | XTALSTB |   |   |   |
| Access |              |   |   |   | RW      |   |   |   |

| Bit | Name     | Function                          | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | Access |
|-----|----------|-----------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 7:4 | Reserved | Reserved                          | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | -      |
| 3:0 | XTALSTB  | XTAL stabilization time selection | <ul style="list-style-type: none"> <li>0b0001: Stabilization counter 35 periods</li> <li>0b0010: Stabilization counter 67 periods</li> <li>0b0011: Stabilization counter 131 periods</li> <li>0b0100: Stabilization counter 259 periods</li> <li>0b0101: Stabilization counter 547 periods</li> <li>0b0110: Stabilization counter 1059 periods</li> <li>0b0111: Stabilization counter 2147 periods</li> <li>0b1000: Stabilization counter 4291 periods</li> <li>0b1001: Stabilization counter 8163 periods</li> <li>Others: Setting prohibited</li> </ul> <div>  <b>Note</b><br/>           One period of the stabilization counter=LRC period/8<br/><br/>           This register can only be configured when CMU_XTALCR.XTALSTP=1 and CMU_OSCSTBSR.XTALSTBF=0.         </div> | RW     |

## 5.4.4 CMU XTAL Fractional Frequency Division Config Register (CMU\_XTALDIVR)

| Offset | Bit Position |    |       |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |          |    |       |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|-------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----------|----|-------|---|---|---|---|---|---|---|---|---|
| 0x0018 | 31           | 30 | 29    | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11       | 10 | 9     | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x03D09040   |    |       |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |          |    |       |   |   |   |   |   |   |   |   |   |
| Name   | Reserved     |    | NUMER |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | Reserved |    | DENOM |   |   |   |   |   |   |   |   |   |
| Access |              |    | RW    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |          |    | RW    |   |   |   |   |   |   |   |   |   |

| Bit   | Name     | Function                                          | Description                                                                                                                                                                                                                                                                                                                                                                                                     | Access |
|-------|----------|---------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:29 | Reserved | Reserved                                          | -                                                                                                                                                                                                                                                                                                                                                                                                               | -      |
| 28:12 | NUMER    | XTAL fractional divider numerator setting value   | XTAL fractional divider numerator setting value, refer to " <a href="#">External High-speed Oscillator Fractional Division</a> " for usage                                                                                                                                                                                                                                                                      | RW     |
| 11    | Reserved | Reserved                                          | -                                                                                                                                                                                                                                                                                                                                                                                                               | -      |
| 10:0  | DENOM    | XTAL fractional divider denominator setting value | XTAL fractional divider denominator setting value. For usage, please refer to " <a href="#">External High-speed Oscillator Fractional Division</a> "<br><br><div>  <b>Note</b><br/> Please set this value to 2<sup>N</sup>. For example, 0x100 represents a denominator of 256; 0x80 represents a denominator of 128. </div> | RW     |

## 5.4.5 CMU XTAL Fractional Frequency Division Control Register (CMU\_XTALDIVCR)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |          |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|----------|
| 0x001C | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0        |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |          |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   | FRADIVEN |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   | RW       |

| Bit  | Name     | Function                                   | Description                                                                                                                          | Access |
|------|----------|--------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:1 | Reserved | Reserved                                   | -                                                                                                                                    | -      |
| 0    | FRADIVEN | XTAL fractional divider start/stop control | <ul style="list-style-type: none"> <li>0b0: XTAL fractional divider stopped</li> <li>0b1: XTAL fractional divider started</li> </ul> | RW     |

### 5.4.6 CMU XTAL Control Register (CMU\_XTALCR)

| Offset | Bit Position |   |   |   |   |   |   |         |
|--------|--------------|---|---|---|---|---|---|---------|
| 0xC032 | 7            | 6 | 5 | 4 | 3 | 2 | 1 | 0       |
| Reset  | 0x01         |   |   |   |   |   |   |         |
| Name   | Reserved     |   |   |   |   |   |   | XTALSTP |
| Access |              |   |   |   |   |   |   | RW      |

| Bit | Name     | Function                           | Description                                                                                                              | Access |
|-----|----------|------------------------------------|--------------------------------------------------------------------------------------------------------------------------|--------|
| 7:1 | Reserved | Reserved                           | -                                                                                                                        | -      |
| 0   | XTALSTP  | XTAL Oscillator Start/Stop Control | <ul style="list-style-type: none"> <li>0b0: XTAL oscillator oscillating</li> <li>0b1: XTAL oscillator stopped</li> </ul> | RW     |



#### Notice

1. When XTAL is selected as system clock or PLLH/PLLA clock source, writing 1 to XTALSTP to stop XTAL oscillator is prohibited.
2. Software sets the XTAL oscillator to oscillate, and after confirming that the XTAL oscillator is stable through the XTALSTBF bit, the system can enter stop mode, Power-down mode, or software sets the XTAL oscillator to stop.
3. The XTAL oscillator is set to stop by software, and the XTAL oscillator can be entered into Stop mode, Power-down mode, or restarted after confirming that the XTAL oscillator is stopped through the XTALSTBF bit.

## 5.4.7 CMU XTAL Clock Stop Detect Control Register (CMU\_XTALSTDCR)

| Offset | Bit Position |          |   |   |   |            |           |           |
|--------|--------------|----------|---|---|---|------------|-----------|-----------|
| 0xC040 | 7            | 6        | 5 | 4 | 3 | 2          | 1         | 0         |
| Reset  | 0x00         |          |   |   |   |            |           |           |
| Name   | XTALSTDE     | Reserved |   |   |   | XTALSTDRIS | XTALSTDRE | XTALSTDIE |
| Access | RW           |          |   |   |   | RW         | RW        | RW        |

| Bit | Name       | Function                                         | Description                                                                                                                                                                                                                                                                                                                                                                                                                             | Access |
|-----|------------|--------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 7   | XTALSTDE   | XTAL Clock Stop Detect Function Enabled          | <ul style="list-style-type: none"> <li>0b0: XTAL clock stop detect disabled</li> <li>0b1: XTAL clock stop detect enabled</li> </ul> <div>  <b>Note</b><br/> Clock stop detect involves detecting abnormal oscillations caused by external factors. Before entering Stop mode or Power-down mode, please disable the clock stop detect function. </div> | RW     |
| 6:3 | Reserved   | Reserved                                         | -                                                                                                                                                                                                                                                                                                                                                                                                                                       | -      |
| 2   | XTALSTDRIS | XTAL Clock Stop Detect Reset Interrupt Selection | <ul style="list-style-type: none"> <li>0b0: XTAL clock stop detect generates an interrupt</li> <li>0b1: XTAL clock stop detect generates a reset</li> </ul> <div>  <b>Note</b><br/> When PLLH/PLLA select XTAL clock as the input source, they can only select the XTAL clock stop detect to generate reset function. </div>                         | RW     |
| 1   | XTALSTDRE  | XTAL Clock Stop Detect Reset Enabled             | <ul style="list-style-type: none"> <li>0b0: XTAL clock stop detect reset disabled</li> <li>0b1: XTAL clock stop detect reset enabled</li> </ul>                                                                                                                                                                                                                                                                                         | RW     |
| 0   | XTALSTDIE  | XTAL Clock Stop Detect Interrupt Enabled         | <ul style="list-style-type: none"> <li>0b0: XTAL clock stop detect interrupt disabled</li> <li>0b1: XTAL clock stop detect interrupt enabled</li> </ul> <div>  <b>Note</b><br/> Set Timer6/Timer4 PWM output to Hi-Z output via EMB, and set the XTALSTDIE bit to 1. </div>                                                                          | RW     |

**Notice**

When XTAL is selected as system clock or PLLH/PLLA clock source, XTALSTP is disabled to write "1" to stop XTAL oscillator.

## 5.4.8 CMU XTAL Clock Stop Detect Status Register (CMU\_XTALSTDSR)

| Offset | Bit Position |   |   |   |   |   |   |          |
|--------|--------------|---|---|---|---|---|---|----------|
| 0xC041 | 7            | 6 | 5 | 4 | 3 | 2 | 1 | 0        |
| Reset  | 0x00         |   |   |   |   |   |   |          |
| Name   | Reserved     |   |   |   |   |   |   | XTALSTDF |
| Access |              |   |   |   |   |   |   | RW       |

| Bit | Name     | Function                          | Description                                                                                                                                                                                                                                                                                                                                                                                  | Access |
|-----|----------|-----------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 7:1 | Reserved | Reserved                          | -                                                                                                                                                                                                                                                                                                                                                                                            | -      |
| 0   | XTALSTDF | XTAL Clock Stop Detect Status Bit | <ul style="list-style-type: none"> <li>0b0: XTAL clock stop not detected</li> <li>0b1: XTAL clock stop detected</li> </ul> Setting conditions: <ul style="list-style-type: none"> <li>XTAL clock stop detect under XTALSTDE=1</li> </ul> Clearing conditions: <ul style="list-style-type: none"> <li>When the system clock selects a clock other than XTAL, read as 1, write as 0</li> </ul> | RW     |

## 5.4.9 CMU XTAL32 Config Register (CMU\_XTAL32CFGR)

| Offset | Bit Position |   |   |   |   |           |   |   |
|--------|--------------|---|---|---|---|-----------|---|---|
| 0x4404 | 7            | 6 | 5 | 4 | 3 | 2         | 1 | 0 |
| Reset  | 0xXX         |   |   |   |   |           |   |   |
| Name   | Reserved     |   |   |   |   | XTAL32DRV |   |   |
| Access |              |   |   |   |   | RW        |   |   |

| Bit | Name      | Function                          | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | Access |
|-----|-----------|-----------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 7:3 | Reserved  | Reserved                          | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | -      |
| 2:0 | XTAL32DRV | XTAL32 Drive Capability Selection | <ul style="list-style-type: none"> <li>● 0b000: Medium drive capability</li> <li>● 0b001: High drive capability</li> <li>● 0b010: Low drive capability</li> <li>● Others: Setting prohibited</li> </ul> <div>  <b>Note</b><br/>           For usage, please refer to the "Electrical Specifications-Operating conditions-External Clock Source Characteristics-Crystal Oscillator/Ceramic Resonator Generated Low-speed External Clock" section of the Datasheet.         </div> | RW     |

## 5.4.10 CMU XTAL32 Filter Control Register (CMU\_XTAL32NFR)

| Offset | Bit Position |   |   |   |   |   |          |   |
|--------|--------------|---|---|---|---|---|----------|---|
| 0x4414 | 7            | 6 | 5 | 4 | 3 | 2 | 1        | 0 |
| Reset  | 0xXX         |   |   |   |   |   |          |   |
| Name   | Reserved     |   |   |   |   |   | XTAL32NF |   |
| Access |              |   |   |   |   |   | RW       |   |

| Bit | Name     | Function                           | Description                                                                                                                                                                                                                                                                                                                                | Access |
|-----|----------|------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 7:2 | Reserved | Reserved                           | -                                                                                                                                                                                                                                                                                                                                          | -      |
| 1:0 | XTAL32NF | XTAL32 Oscillator Filter Selection | <ul style="list-style-type: none"> <li>0b00: XTAL32 3μs filter enabled in RUN mode/ Stop mode/Power-down mode</li> <li>0b01: XTAL32 3μs filter enabled in RUN mode, disabled in stop mode or Power-down mode</li> <li>0b10: Setting prohibited</li> <li>0b11: XTAL32 3μs filter disabled in RUN mode/ stop mode/Power-down mode</li> </ul> | RW     |

## 5.4.11 CMU XTAL32 Control Register (CMU\_XTAL32CR)

| Offset | Bit Position |   |   |   |   |   |           |   |
|--------|--------------|---|---|---|---|---|-----------|---|
| 0x4400 | 7            | 6 | 5 | 4 | 3 | 2 | 1         | 0 |
| Reset  | 0x00         |   |   |   |   |   |           |   |
| Name   | Reserved     |   |   |   |   |   | XTAL32STP |   |
| Access |              |   |   |   |   |   | RW        |   |

| Bit | Name      | Function                         | Description                                                                                                                  | Access |
|-----|-----------|----------------------------------|------------------------------------------------------------------------------------------------------------------------------|--------|
| 7:1 | Reserved  | Reserved                         | -                                                                                                                            | -      |
| 0   | XTAL32STP | XTAL32 oscillator start/stop bit | <ul style="list-style-type: none"> <li>0b0: XTAL32 oscillator oscillating</li> <li>0b1: XTAL32 oscillator stopped</li> </ul> | RW     |

**Notice**

1. When XTAL32 is selected as the system clock source, stopping the XTAL32 oscillator by writing 1 to XTAL32STP is prohibited.
2. After software sets the XTAL32 oscillator to oscillate, wait for 5 XTAL32 periods before stopping XTAL32 again.
3. After software sets XTAL32 to stop, wait for 5 XTAL32 periods before starting XTAL32 again.



## 5.4.12 CMU HRC Trimming Register (CMU\_HRCTRM)

| Offset | Bit Position |   |   |   |   |   |   |   |
|--------|--------------|---|---|---|---|---|---|---|
| 0xC062 | 7            | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00         |   |   |   |   |   |   |   |
| Name   | HRCTRM       |   |   |   |   |   |   |   |
| Access | RW           |   |   |   |   |   |   |   |

| Bit | Name   | Function               | Description                                                                                                                                                                                                                                                                       | Access |
|-----|--------|------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 7:0 | HRCTRM | HRC Frequency Trimming | <ul style="list-style-type: none"> <li>● 0b10000000: -128</li> <li>● 0b10000001: -127</li> <li>● .....</li> <li>● 0b11111111: -1</li> <li>● 0b00000000: Center Code</li> <li>● 0b00000001: +1</li> <li>● .....</li> <li>● 0b01111110: +126</li> <li>● 0b01111111: +127</li> </ul> | RW     |

**Note**

Frequency calibration needs to be within the HRC frequency guarantee range.

## 5.4.13 CMU HRC Control Register (CMU\_HRCCR)

| Offset | Bit Position                         |   |   |   |   |   |   |        |
|--------|--------------------------------------|---|---|---|---|---|---|--------|
| 0xC036 | 7                                    | 6 | 5 | 4 | 3 | 2 | 1 | 0      |
| Reset  | Determined by the ICG1.HRCSTOP value |   |   |   |   |   |   |        |
| Name   | Reserved                             |   |   |   |   |   |   | HRCSTP |
| Access |                                      |   |   |   |   |   |   | RW     |

| Bit | Name     | Function                     | Description                                                                                                                                                                                                                                                                                                                            | Access |
|-----|----------|------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 7:1 | Reserved | Reserved                     | -                                                                                                                                                                                                                                                                                                                                      | -      |
| 0   | HRCSTP   | HRC Oscillator Open Stop Bit | <ul style="list-style-type: none"> <li>0b0: HRC oscillator oscillating</li> <li>0b1: HRC oscillator stopped</li> </ul> <div>  <b>Note</b><br/>           After reset, HRC oscillates or stops based on the ICG1.HRCSTOP configuration.         </div> | RW     |

**Notice**

1. When HRC is selected as the system clock source or PLLH/PLLA clock source, it is forbidden to stop the HRC clock by writing 1 to HRCSTP.
2. After software sets the HRC oscillator to oscillate, and confirms that HRC is stable through the HRCSTBF bit, the system can enter Stop mode, Power-down mode, or stop HRC.
3. After software sets HRC to stop, and confirms that HRC has stopped through the HRCSTBF bit, the system can enter Stop mode, Power-down mode, or restart HRC.

## 5.4.14 CMU MRC Trimming Register (CMU\_MRCTRM)

| Offset | Bit Position |   |   |   |   |   |   |   |
|--------|--------------|---|---|---|---|---|---|---|
| 0xC061 | 7            | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00         |   |   |   |   |   |   |   |
| Name   | MRCTRM       |   |   |   |   |   |   |   |
| Access | RW           |   |   |   |   |   |   |   |

| Bit | Name   | Function               | Description                                                                                                                                                                                                                                                                       | Access |
|-----|--------|------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 7:0 | MRCTRM | MRC Frequency Trimming | <ul style="list-style-type: none"> <li>● 0b10000000: -128</li> <li>● 0b10000001: -127</li> <li>● .....</li> <li>● 0b11111111: -1</li> <li>● 0b00000000: Center Code</li> <li>● 0b00000001: +1</li> <li>● .....</li> <li>● 0b01111110: +126</li> <li>● 0b01111111: +127</li> </ul> | RW     |

**Note**

Frequency calibration needs to be within the MRC frequency guarantee range.

## 5.4.15 CMU MRC Control Register (CMU\_MRCCR)

| Offset | Bit Position |   |   |   |   |   |   |        |
|--------|--------------|---|---|---|---|---|---|--------|
| 0xC038 | 7            | 6 | 5 | 4 | 3 | 2 | 1 | 0      |
| Reset  | 0x80         |   |   |   |   |   |   |        |
| Name   | Reserved     |   |   |   |   |   |   | MRCSTP |
| Access |              |   |   |   |   |   |   | RW     |

| Bit | Name     | Function                      | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | Access |
|-----|----------|-------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 7   | Reserved | Reserved                      | Reads as 1, Write as 1                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | -      |
| 6:1 | Reserved | Reserved                      | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | -      |
| 0   | MRCSTP   | MRC oscillator start/stop bit | <ul style="list-style-type: none"> <li>0b0: MRC oscillator oscillating</li> <li>0b1: MRC oscillator stopped</li> </ul> <div>  <b>Note</b> <ul style="list-style-type: none"> <li>When the XTAL oscillation fault function is enabled, this bit is cleared simultaneously, and MRC oscillates.</li> <li>The Stop mode wake-up action when PWC_STPMCR.CKSMRC bit is 1 is set when the MRC oscillator is oscillating.</li> </ul> </div> | RW     |

**Notice**

1. When MRC is selected as the system clock source, stopping the MRC clock by writing 1 to MRCSTP is prohibited.
2. After software sets the MRC oscillator to oscillate, wait for 5 MRC periods before entering Stop mode, Power-down mode, or stopping MRC.
3. The software sets the MRC to stop, and waits for 5 MRC periods before entering the Stop mode, Power-down mode, or restarting the MRC.
4. MRC is used as RTC calibration clock. When RTC is not initialized, there is a possibility of MRC oscillation. When the RTC calibration function is enabled, MRC oscillates regardless of the MRCSTP bit setting.

## 5.4.16 CMU LRC Trimming Register (CMU\_LRCTRM)

| Offset | Bit Position |   |   |   |   |   |   |   |
|--------|--------------|---|---|---|---|---|---|---|
| 0x4424 | 7            | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00         |   |   |   |   |   |   |   |
| Name   | LRCTRM       |   |   |   |   |   |   |   |
| Access | RW           |   |   |   |   |   |   |   |

| Bit | Name   | Function               | Description                                                                                                                                                                                                                                                                       | Access |
|-----|--------|------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 7:0 | LRCTRM | LRC Frequency Trimming | <ul style="list-style-type: none"> <li>● 0b10000000: -128</li> <li>● 0b10000001: -127</li> <li>● .....</li> <li>● 0b11111111: -1</li> <li>● 0b00000000: Center Code</li> <li>● 0b00000001: +1</li> <li>● .....</li> <li>● 0b01111110: +126</li> <li>● 0b01111111: +127</li> </ul> | RW     |

**Note**

Frequency calibration needs to be within the LRC frequency guarantee range.

## 5.4.17 CMU LRC Control Register (CMU\_LRCCR)

| Offset | Bit Position |   |   |   |   |   |   |        |
|--------|--------------|---|---|---|---|---|---|--------|
| 0x441C | 7            | 6 | 5 | 4 | 3 | 2 | 1 | 0      |
| Reset  | 0x00         |   |   |   |   |   |   |        |
| Name   | Reserved     |   |   |   |   |   |   | LRCSTP |
| Access |              |   |   |   |   |   |   | RW     |

| Bit | Name     | Function                      | Description                                                                                                            | Access |
|-----|----------|-------------------------------|------------------------------------------------------------------------------------------------------------------------|--------|
| 7:1 | Reserved | Reserved                      | -                                                                                                                      | -      |
| 0   | LRCSTP   | LRC oscillator start/stop bit | <ul style="list-style-type: none"> <li>0b0: LRC oscillator oscillating</li> <li>0b1: LRC oscillator stopped</li> </ul> | RW     |

**Notice**

- When LRC is selected as the system clock source, stopping the LRC clock by writing 1 to LRCSTP is prohibited.
- After software sets the LRC oscillator to oscillate, wait for 5 LRC periods before entering Stop mode, Power-down mode or stopping LRC.
- The software sets the LRC to stop, and waits for 5 LRC periods before entering the Stop mode, Power-down mode or starting the LRC again.
- When waiting for the XTAL oscillator, HRC, PLLH, PLLA clocks to stabilize, the LRCSTP bit setting is ignored, and LRC oscillates forcibly.

## 5.4.18 CMU RTCLRC Trimming Register (CMU\_RTCLRCTRM)

| Offset | Bit Position |   |   |   |   |   |   |   |
|--------|--------------|---|---|---|---|---|---|---|
| 0x442C | 7            | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00         |   |   |   |   |   |   |   |
| Name   | RTCLRCTRM    |   |   |   |   |   |   |   |
| Access | RW           |   |   |   |   |   |   |   |

| Bit | Name      | Function                       | Description                                                                                                                                                                                                                                                                       | Access |
|-----|-----------|--------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 7:0 | RTCLRCTRM | RTCLRC Fre-<br>quency Trimming | <ul style="list-style-type: none"> <li>● 0b10000000: -128</li> <li>● 0b10000001: -127</li> <li>● .....</li> <li>● 0b11111111: -1</li> <li>● 0b00000000: Center Code</li> <li>● 0b00000001: +1</li> <li>● .....</li> <li>● 0b01111110: +126</li> <li>● 0b01111111: +127</li> </ul> | RW     |

**Note**

Frequency calibration needs to be within the RTCLRC frequency guarantee range.

## 5.4.19 CMU PLLH Config Register (CMU\_PLLHCFGR)

| Offset | Bit Position |    |    |    |       |    |    |    |       |    |    |    |          |    |    |    |       |    |    |    |        |    |   |   |          |   |   |   |       |   |   |   |
|--------|--------------|----|----|----|-------|----|----|----|-------|----|----|----|----------|----|----|----|-------|----|----|----|--------|----|---|---|----------|---|---|---|-------|---|---|---|
| 0xC100 | 31           | 30 | 29 | 28 | 27    | 26 | 25 | 24 | 23    | 22 | 21 | 20 | 19       | 18 | 17 | 16 | 15    | 14 | 13 | 12 | 11     | 10 | 9 | 8 | 7        | 6 | 5 | 4 | 3     | 2 | 1 | 0 |
| Reset  | 0x11101300   |    |    |    |       |    |    |    |       |    |    |    |          |    |    |    |       |    |    |    |        |    |   |   |          |   |   |   |       |   |   |   |
| Name   | PLLHP        |    |    |    | PLLHQ |    |    |    | PLLHR |    |    |    | Reserved |    |    |    | PLLHN |    |    |    | PLLSRC |    |   |   | Reserved |   |   |   | PLLHM |   |   |   |
| Access | RW           |    |    |    | RW    |    |    |    | RW    |    |    |    |          |    |    |    | RW    |    |    |    | RW     |    |   |   |          |   |   |   | RW    |   |   |   |

| Bit   | Name  | Function                    | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | Access |
|-------|-------|-----------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:28 | PLLHP | PLLHP Clock Division Factor | PLLHP output clock frequency=PLLH VCO frequency/PLLHP<br><ul style="list-style-type: none"> <li>● 0b0000: Setting prohibited</li> <li>● 0b0001: Divide by 2</li> <li>● 0b0010: Divide by 3</li> <li>● .....</li> <li>● 0b1110: Divide by 15</li> <li>● 0b1111: Divide by 16</li> </ul> <div>  <b>Note</b><br/>           Write to PLLHP under the condition that PLLH is stopped to configure the PLLHP clock frequency.         </div>  | RW     |
| 27:24 | PLLHQ | PLLHQ Clock Division Factor | PLLHQ output clock frequency=PLLH VCO frequency/PLLHQ<br><ul style="list-style-type: none"> <li>● 0b0000: Setting prohibited</li> <li>● 0b0001: Divide by 2</li> <li>● 0b0010: Divide by 3</li> <li>● .....</li> <li>● 0b1110: Divide by 15</li> <li>● 0b1111: Divide by 16</li> </ul> <div>  <b>Note</b><br/>           Write to PLLHQ under the condition that PLLH is stopped to configure the PLLHQ clock frequency.         </div> | RW     |



| Bit   | Name     | Function                                  | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | Access |
|-------|----------|-------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 23:20 | PLLHR    | PLLHR Clock Division Factor               | <p>PLLHR output clock frequency=PLLH VCO Frequency/PLLHR</p> <ul style="list-style-type: none"> <li>● 0b0000: Setting prohibited</li> <li>● 0b0001: Divide by 2</li> <li>● 0b0010: Divide by 3</li> <li>● .....</li> <li>● 0b1110: Divide by 15</li> <li>● 0b1111: Divide by 16</li> </ul> <div>  <b>Note</b><br/>           Write to PLLHR under the condition that PLLH is stopped to configure the PLLHR clock frequency.         </div>                                                                                                   | RW     |
| 19:16 | Reserved | Reserved                                  | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | -      |
| 15:8  | PLLHN    | PLLH Frequency Multiplication Coefficient | <p>PLLH VCO frequency=PLLH VCO input frequency * PLLHN</p> <ul style="list-style-type: none"> <li>● 0b00011000: 25</li> <li>● 0b00011001: 26</li> <li>● 0b00011010: 27</li> <li>● .....</li> <li>● 0b10010100: 149</li> <li>● 0b10010101: 150</li> <li>● Others: Setting prohibited</li> </ul> <div>  <b>Note</b><br/>           Used to control the frequency multiplication coefficient of the PLLH VCO. Write PLLHN under the PLLH stop condition. Make sure the VCO frequency of the PLLH is between 600MHz and 1200MHz.         </div> | RW     |
| 7     | PLLSRC   | PLLH/PLLA Input Clock Source Selection    | <ul style="list-style-type: none"> <li>● 0b0: Select external high-speed oscillator</li> <li>● 0b1: Select internal high-speed oscillator</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | RW     |
| 6:2   | Reserved | Reserved                                  | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | -      |
| 1:0   | PLLHM    | PLLH Input Clock Division Factor          | <ul style="list-style-type: none"> <li>● 0b00: Divide by 1</li> <li>● 0b01: Divide by 2</li> <li>● 0b10: Divide by 3</li> <li>● 0b11: Divide by 4</li> </ul> <div>  <b>Note</b><br/>           Used to divide the PLLH input clock before the VCO of PLLH. Write PLLHM under the PLLH stop condition. Ensure that the PLLH PFD input clock frequency is between 8MHz and 25MHz.         </div>                                                                                                                                              | RW     |

## 5.4.20 CMU PLLH Control Register (CMU\_PLLHCR)

| Offset | Bit Position |   |   |   |   |   |   |         |
|--------|--------------|---|---|---|---|---|---|---------|
| 0xC02A | 7            | 6 | 5 | 4 | 3 | 2 | 1 | 0       |
| Reset  | 0x01         |   |   |   |   |   |   |         |
| Name   | Reserved     |   |   |   |   |   |   | PLLHOFF |
| Access |              |   |   |   |   |   |   | RW      |

| Bit | Name     | Function     | Description                                                                                                                                                                                                                                                                                                  | Access |
|-----|----------|--------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 7:1 | Reserved | Reserved     | -                                                                                                                                                                                                                                                                                                            | -      |
| 0   | PLLHOFF  | PLLH Enabled | <ul style="list-style-type: none"> <li>0b0: PLLH action starts</li> <li>0b1: PLLH stops</li> </ul> <div>  <b>Note</b><br/>           Do not set this bit to 1 if the PLLH clock is used as the system clock.         </div> | RW     |

**Notice**

- When PLLH is selected as the system clock source, it is prohibited to stop the PLLH clock by writing 1 to PLLHOFF.
- Software sets PLLH action to start, and after confirming that PLLH is stable through the PLLHSTBF bit, the system can enter Stop mode, Power-down mode, or software-configured stop of PLLH.
- Software sets PLLH to stop, and after confirming that PLLH is stopped through the PLLHSTBF bit, the system can enter Stop mode, Power-down mode, or restart PLLH.
- When PLLH selects the XTAL oscillator as the clock source, set PLLH action to start after confirming that the XTAL oscillator is stable through the XTALSTBF bit. When PLLH selects HRC as the clock source, the PLLH action can be set to start after confirming that HRC is stable through the HRCSTBF bit.

## 5.4.21 CMU PLLA Config Register (CMU\_PLLACFGR)

| Offset | Bit Position |    |    |    |       |    |    |    |       |    |    |    |          |    |    |    |       |    |    |    |    |    |   |   |          |   |   |   |       |   |   |   |
|--------|--------------|----|----|----|-------|----|----|----|-------|----|----|----|----------|----|----|----|-------|----|----|----|----|----|---|---|----------|---|---|---|-------|---|---|---|
| 0xC104 | 31           | 30 | 29 | 28 | 27    | 26 | 25 | 24 | 23    | 22 | 21 | 20 | 19       | 18 | 17 | 16 | 15    | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7        | 6 | 5 | 4 | 3     | 2 | 1 | 0 |
| Reset  | 0x11101300   |    |    |    |       |    |    |    |       |    |    |    |          |    |    |    |       |    |    |    |    |    |   |   |          |   |   |   |       |   |   |   |
| Name   | PLLAP        |    |    |    | PLLAQ |    |    |    | PLLAR |    |    |    | Reserved |    |    |    | PLLAN |    |    |    |    |    |   |   | Reserved |   |   |   | PLLAM |   |   |   |
| Access | RW           |    |    |    | RW    |    |    |    | RW    |    |    |    |          |    |    |    | RW    |    |    |    |    |    |   |   |          |   |   |   | RW    |   |   |   |

| Bit   | Name  | Function                    | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | Access |
|-------|-------|-----------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:28 | PLLAP | PLLAP Clock Division Factor | PLLAP output clock frequency=PLLA VCO frequency/PLLAP<br><ul style="list-style-type: none"> <li>● 0b0000: Setting prohibited</li> <li>● 0b0001: Divide by 2</li> <li>● 0b0010: Divide by 3</li> <li>● .....</li> <li>● 0b1110: Divide by 15</li> <li>● 0b1111: Divide by 16</li> </ul> <div>  <b>Note</b><br/>           Write PLLAP under the PLLA stop condition to set the frequency of the PLLAP clock.         </div>  | RW     |
| 27:24 | PLLAQ | PLLAQ Clock Division Factor | PLLAQ output clock frequency=PLLA VCO frequency/PLLAQ<br><ul style="list-style-type: none"> <li>● 0b0000: Setting prohibited</li> <li>● 0b0001: Divide by 2</li> <li>● 0b0010: Divide by 3</li> <li>● .....</li> <li>● 0b1110: Divide by 15</li> <li>● 0b1111: Divide by 16</li> </ul> <div>  <b>Note</b><br/>           Write PLLAQ under the PLLA stop condition to set the frequency of the PLLAQ clock.         </div> | RW     |

| Bit   | Name     | Function                                  | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | Access |
|-------|----------|-------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 23:20 | PLLAR    | PLLAR Clock Division Factor               | <p>PLLAR output clock frequency=PLLA VCO frequency/PLLAR</p> <ul style="list-style-type: none"> <li>● 0b0000: Setting prohibited</li> <li>● 0b0001: Divide by 2</li> <li>● 0b0010: Divide by 3</li> <li>● .....</li> <li>● 0b1110: Divide by 15</li> <li>● 0b1111: Divide by 16</li> </ul> <div>  <b>Note</b><br/>           Write PLLAR under the PLLA stop condition to set the frequency of the PLLAR clock.         </div>                                                                                                                         | RW     |
| 19:17 | Reserved | Reserved                                  | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | -      |
| 16:8  | PLLAN    | PLLA Frequency Multiplication Coefficient | <p>VCO frequency of PLLA=VCO input frequency of PLLA * PLLAN</p> <ul style="list-style-type: none"> <li>● 0b000010011: 20</li> <li>● 0b000010100: 21</li> <li>● 0b000010101: 22</li> <li>● .....</li> <li>● 0b111011110: 479</li> <li>● 0b111011111: 480</li> <li>● Others: Setting prohibited</li> </ul> <div>  <b>Note</b><br/>           Used to control the frequency multiplication coefficient of the PLLA VCO. Write PLLAN under the PLLA stop condition. Make sure the VCO frequency of the PLLA is between 240MHz and 480Hz.         </div> | RW     |
| 7:5   | Reserved | Reserved                                  | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | -      |
| 4:0   | PLLAM    | PLLA Input Clock Division Factor          | <ul style="list-style-type: none"> <li>● 0b00000: Divide by 1</li> <li>● 0b00001: Divide by 2</li> <li>● 0b00010: Divide by 3</li> <li>● .....</li> <li>● 0b11000: Divide by 25</li> <li>● Others: Setting prohibited</li> </ul> <div>  <b>Note</b><br/>           Used to divide the PLLA input clock before the VCO of PLLA. Write PLLAM under the PLLA stop condition. Ensure that the PLLA PFD input clock frequency is between 1MHz and 25MHz.         </div>                                                                                   | RW     |

## 5.4.22 CMU PLLA Control Register (CMU\_PLLACR)

| Offset | Bit Position |   |   |   |   |   |   |         |
|--------|--------------|---|---|---|---|---|---|---------|
| 0xC02E | 7            | 6 | 5 | 4 | 3 | 2 | 1 | 0       |
| Reset  | 0x01         |   |   |   |   |   |   |         |
| Name   | Reserved     |   |   |   |   |   |   | PLLAOFF |
| Access |              |   |   |   |   |   |   | RW      |

| Bit | Name     | Function     | Description                                                                                                                                                                                                                                                                                                  | Access |
|-----|----------|--------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 7:1 | Reserved | Reserved     | -                                                                                                                                                                                                                                                                                                            | -      |
| 0   | PLLAOFF  | PLLA Enabled | <ul style="list-style-type: none"> <li>0b0: PLLA action starts</li> <li>0b1: PLLA stops</li> </ul> <div>  <b>Note</b><br/>           Do not set this bit to 1 if the PLLA clock is used as the system clock.         </div> | RW     |

**Notice**

- When PLLA is selected as the clock source for I2S/ADC/DAC/USB/CAN/MCAN, it is prohibited to stop the PLLA clock by writing 1 to PLLAOFF.
- Software sets PLLA action to start, and after confirming that PLLA is stable through the PLLASTBF bit, the system can enter Stop mode, Power-down mode, or software-configured stop of PLLA.
- Software sets PLLA to stop, and after confirming that PLLA is stopped through the PLLASTBF bit, the system can enter Stop mode, Power-down mode, or restart PLLA.
- When PLLA selects the XTAL oscillator as the clock source, set PLLA action to start after confirming that the XTAL oscillator is stable through the XTALSTBF bit. When PLLA selects HRC as the clock source, the PLLA action can be set to start after confirming that HRC is stable through the HRCSTBF bit.

## 5.4.23 CMU Clock Source Stable Status Register (CMU\_OSCSTBSR)

| Offset | Bit Position |          |           |          |          |          |   |         |
|--------|--------------|----------|-----------|----------|----------|----------|---|---------|
| 0xC03C | 7            | 6        | 5         | 4        | 3        | 2        | 1 | 0       |
| Reset  | 0x00         |          |           |          |          |          |   |         |
| Name   | Reserved     | PLLASTBF | PL LHSTBF | Reserved | XTALSTBF | Reserved |   | HRCSTBF |
| Access |              | RO       | RO        |          | RO       |          |   | RO      |

| Bit | Name      | Function                 | Description                                                                                                     | Access |
|-----|-----------|--------------------------|-----------------------------------------------------------------------------------------------------------------|--------|
| 7   | Reserved  | Reserved                 | -                                                                                                               | -      |
| 6   | PLLASTBF  | PLLA Stability Flag Bit  | <ul style="list-style-type: none"> <li>0b0: PLLA is stopped or unstable</li> <li>0b1: PLLA is stable</li> </ul> | RO     |
| 5   | PL LHSTBF | PL LH Stability Flag Bit | <ul style="list-style-type: none"> <li>0b0: PLLH is stopped or unstable</li> <li>0b1: PLLH is stable</li> </ul> | RO     |
| 4   | Reserved  | Reserved                 | -                                                                                                               | -      |
| 3   | XTALSTBF  | XTAL Stability Flag Bit  | <ul style="list-style-type: none"> <li>0b0: XTAL is stopped or unstable</li> <li>0b1: XTAL is stable</li> </ul> | RO     |
| 2:1 | Reserved  | Reserved                 | -                                                                                                               | -      |
| 0   | HRCSTBF   | HRC Stability Flag Bit   | <ul style="list-style-type: none"> <li>0b0: HRC is stopped or unstable</li> <li>0b1: HRC is stable</li> </ul>   | RO     |

## 5.4.24 CMU System Clock Source Switch Register (CMU\_CKSWR)

| Offset | Bit Position |   |   |   |   |      |   |   |
|--------|--------------|---|---|---|---|------|---|---|
| 0xC026 | 7            | 6 | 5 | 4 | 3 | 2    | 1 | 0 |
| Reset  | 0x01         |   |   |   |   |      |   |   |
| Name   | Reserved     |   |   |   |   | CKSW |   |   |
| Access |              |   |   |   |   | RW   |   |   |

| Bit | Name            | Function                      | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | Access |
|-----|-----------------|-------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 7:3 | <i>Reserved</i> | <i>Reserved</i>               | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | -      |
| 2:0 | CKSW            | System Clock Source Switching | <ul style="list-style-type: none"> <li>0b000: Select HRC clock as the system clock</li> <li>0b001: Select MRC clock as the system clock</li> <li>0b010: Select LRC clock as the system clock</li> <li>0b011: Select XTAL clock as the system clock</li> <li>0b100: Select the XTAL32 clock as the system clock</li> <li>0b101: Select PLLH as the system clock</li> <li>Others: Setting prohibited</li> </ul> <div>  <b>Note</b> <ul style="list-style-type: none"> <li>Ensure that the target clock source is in a stable state before switching. Please refer to "<a href="#">Source Switching</a>" for the operating steps.</li> <li>When the PWC_STPMCR.CKSMRC bit is 1, after waking up from Stop mode, this register initializes, and the system clock source selects MRC clock.</li> </ul> </div> | RW     |

## 5.4.25 CMU Clock Divider Config Register (CMU\_SCFGR)

| Offset | Bit Position |    |    |    |       |    |    |    |          |       |    |    |    |          |        |    |    |    |          |        |    |    |   |          |        |   |   |   |          |        |   |   |  |          |        |  |  |  |
|--------|--------------|----|----|----|-------|----|----|----|----------|-------|----|----|----|----------|--------|----|----|----|----------|--------|----|----|---|----------|--------|---|---|---|----------|--------|---|---|--|----------|--------|--|--|--|
| 0xC020 | 31           | 30 | 29 | 28 | 27    | 26 | 25 | 24 | 23       | 22    | 21 | 20 | 19 | 18       | 17     | 16 | 15 | 14 | 13       | 12     | 11 | 10 | 9 | 8        | 7      | 6 | 5 | 4 | 3        | 2      | 1 | 0 |  |          |        |  |  |  |
| Reset  | 0x00000000   |    |    |    |       |    |    |    |          |       |    |    |    |          |        |    |    |    |          |        |    |    |   |          |        |   |   |   |          |        |   |   |  |          |        |  |  |  |
| Name   | Reserved     |    |    |    | HCLKS |    |    |    | Reserved | EXCKS |    |    |    | Reserved | PCLK4S |    |    |    | Reserved | PCLK3S |    |    |   | Reserved | PCLK2S |   |   |   | Reserved | PCLK1S |   |   |  | Reserved | PCLK0S |  |  |  |
| Access |              |    |    |    | RW    |    |    |    |          | RW    |    |    |    |          | RW     |    |    |    |          | RW     |    |    |   |          | RW     |   |   |   |          | RW     |   |   |  |          | RW     |  |  |  |

| Bit   | Name     | Function                           | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | Access |
|-------|----------|------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:27 | Reserved | Reserved                           | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | -      |
| 26:24 | HCLKS    | HCLK Clock Division Selection Bit  | <ul style="list-style-type: none"> <li>0b000: 1 division of the system clock</li> <li>0b001: 2 division of the system clock</li> <li>0b010: 4 division of the system clock</li> <li>0b011: 8 division of the system clock</li> <li>0b100: 16 division of the system clock</li> <li>0b101: 32 division of the system clock</li> <li>0b110: 64 division of the system clock</li> <li>0b111: Setting prohibited</li> </ul> <div>  <b>Note</b><br/>           When the PWC_STPMCR.CKSMRC bit is set to 1, this register is initialized after waking up from Stop Mode, and HCLK is set to the system clock divided by 1.         </div>  | RW     |
| 23    | Reserved | Reserved                           | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | -      |
| 22:20 | EXCKS    | EXMC Clock Division Selection Bit  | <ul style="list-style-type: none"> <li>0b000: 1 division of the system clock</li> <li>0b001: 2 division of the system clock</li> <li>0b010: 4 division of the system clock</li> <li>0b011: 8 division of the system clock</li> <li>0b100: 16 division of the system clock</li> <li>0b101: 32 division of the system clock</li> <li>0b110: 64 division of the system clock</li> <li>0b111: Setting prohibited</li> </ul> <div>  <b>Note</b><br/>           When the PWC_STPMCR.CKSMRC bit is set to 1, this register is initialized after waking up from Stop Mode, and EXCLK is set to the system clock divided by 1.         </div> | RW     |
| 19    | Reserved | Reserved                           | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | -      |
| 18:16 | PCLK4S   | PCLK4 Clock Division Selection Bit | <ul style="list-style-type: none"> <li>0b000: 1 division of the system clock</li> <li>0b001: 2 division of the system clock</li> <li>0b010: 4 division of the system clock</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | RW     |



| Bit   | Name            | Function                           | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | Access |
|-------|-----------------|------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 18:16 | PCLK4S          | PCLK4 Clock Division Selection Bit | <ul style="list-style-type: none"> <li>● 0b011: 8 division of the system clock</li> <li>● 0b100: 16 division of the system clock</li> <li>● 0b101: 32 division of the system clock</li> <li>● 0b110: 64 division of the system clock</li> <li>● 0b111: Setting prohibited</li> </ul> <div>  <b>Note</b><br/>           When the PWC_STPMCR.CKSMRC bit is 1, after waking up from Stop Mode, this register initializes, and PCLK4 is 1 division of the system clock.         </div>                                                                                                                                                      | RW     |
| 15    | <i>Reserved</i> | <i>Reserved</i>                    | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | -      |
| 14:12 | PCLK3S          | PCLK3 Clock Division Selection Bit | <ul style="list-style-type: none"> <li>● 0b000: 1 division of the system clock</li> <li>● 0b001: 2 division of the system clock</li> <li>● 0b010: 4 division of the system clock</li> <li>● 0b011: 8 division of the system clock</li> <li>● 0b100: 16 division of the system clock</li> <li>● 0b101: 32 division of the system clock</li> <li>● 0b110: 64 division of the system clock</li> <li>● 0b111: Setting prohibited</li> </ul> <div>  <b>Note</b><br/>           When the PWC_STPMCR.CKSMRC bit is 1, after waking up from Stop Mode, this register initializes, and PCLK3 is 1 division of the system clock.         </div>  | RW     |
| 11    | <i>Reserved</i> | <i>Reserved</i>                    | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | -      |
| 10:8  | PCLK2S          | PCLK2 Clock Division Selection Bit | <ul style="list-style-type: none"> <li>● 0b000: 1 division of the system clock</li> <li>● 0b001: 2 division of the system clock</li> <li>● 0b010: 4 division of the system clock</li> <li>● 0b011: 8 division of the system clock</li> <li>● 0b100: 16 division of the system clock</li> <li>● 0b101: 32 division of the system clock</li> <li>● 0b110: 64 division of the system clock</li> <li>● 0b111: Setting prohibited</li> </ul> <div>  <b>Note</b><br/>           When the PWC_STPMCR.CKSMRC bit is 1, after waking up from Stop Mode, this register initializes, and PCLK2 is 1 division of the system clock.         </div> | RW     |
| 7     | <i>Reserved</i> | <i>Reserved</i>                    | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | -      |

| Bit | Name            | Function                           | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | Access |
|-----|-----------------|------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 6:4 | PCLK1S          | PCLK1 Clock Division Selection Bit | <ul style="list-style-type: none"> <li>● 0b000: 1 division of the system clock</li> <li>● 0b001: 2 division of the system clock</li> <li>● 0b010: 4 division of the system clock</li> <li>● 0b011: 8 division of the system clock</li> <li>● 0b100: 16 division of the system clock</li> <li>● 0b101: 32 division of the system clock</li> <li>● 0b110: 64 division of the system clock</li> <li>● 0b111: Setting prohibited</li> </ul> <div>  <b>Note</b><br/>           When the PWC_STPMCR.CKSMRC bit is 1, after waking up from Stop Mode, this register initializes, and PCLK1 is 1 division of the system clock.         </div>   | RW     |
| 3   | <i>Reserved</i> | <i>Reserved</i>                    | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | -      |
| 2:0 | PCLK0S          | PCLK0 Clock Division Selection Bit | <ul style="list-style-type: none"> <li>● 0b000: 1 division of the system clock</li> <li>● 0b001: 2 division of the system clock</li> <li>● 0b010: 4 division of the system clock</li> <li>● 0b011: 8 division of the system clock</li> <li>● 0b100: 16 division of the system clock</li> <li>● 0b101: 32 division of the system clock</li> <li>● 0b110: 64 division of the system clock</li> <li>● 0b111: Setting prohibited</li> </ul> <div>  <b>Note</b><br/>           When the PWC_STPMCR.CKSMRC bit is 1, after waking up from Stop Mode, this register initializes, and PCLK0 is 1 division of the system clock.         </div> | RW     |

## 5.4.26 CMU USB Clock Config Register (CMU\_USBCKCFGR)

| Offset | Bit Position |   |   |   |          |   |   |   |
|--------|--------------|---|---|---|----------|---|---|---|
| 0xC024 | 7            | 6 | 5 | 4 | 3        | 2 | 1 | 0 |
| Reset  | 0x40         |   |   |   |          |   |   |   |
| Name   | USBCKS       |   |   |   | Reserved |   |   |   |
| Access | RW           |   |   |   |          |   |   |   |

| Bit | Name     | Function                                 | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | Access |
|-----|----------|------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 7:4 | USBCKS   | USBFS/USBHS 48MHz Clock Source Selection | <ul style="list-style-type: none"> <li>● 0b0001: 2 divisions of the system clock</li> <li>● 0b0010: 3 divisions of the system clock</li> <li>● 0b0011: 4 divisions of the system clock</li> <li>● 0b0100: 5 divisions of the system clock</li> <li>● 0b0101: 6 divisions of the system clock</li> <li>● 0b0110: 7 divisions of the system clock</li> <li>● 0b0111: 8 divisions of the system clock</li> <li>● 0b1000: PLLHQ</li> <li>● 0b1001: PLLHR</li> <li>● 0b1010: PLLAP</li> <li>● 0b1011: PLLAQ</li> <li>● 0b1100: PLLAR</li> <li>● Others: Setting prohibited</li> </ul> <div>  <b>Note</b> <ul style="list-style-type: none"> <li>● Ensure that the target clock source is in a stable state when switching to PLLH/PLLA.</li> <li>● When the PWC_STPMCR.CKSMRC bit is set to 1, this register is initialized after waking up from Stop mode, and USBCLK is set to the system clock divided by 5.</li> </ul> </div> | RW     |
| 3:0 | Reserved | Reserved                                 | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | -      |

## 5.4.27 CMU CAN Clock Config Register (CMU\_CANCKCFGR)

| Offset | Bit Position |    |    |    |          |    |   |   |         |   |   |   |         |   |   |   |
|--------|--------------|----|----|----|----------|----|---|---|---------|---|---|---|---------|---|---|---|
| 0xC018 | 15           | 14 | 13 | 12 | 11       | 10 | 9 | 8 | 7       | 6 | 5 | 4 | 3       | 2 | 1 | 0 |
| Reset  | 0xDDDD       |    |    |    |          |    |   |   |         |   |   |   |         |   |   |   |
| Name   | MCAN2CKS     |    |    |    | MCAN1CKS |    |   |   | CAN2CKS |   |   |   | CAN1CKS |   |   |   |
| Access | RW           |    |    |    | RW       |    |   |   | RW      |   |   |   | RW      |   |   |   |

| Bit   | Name     | Function                            | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | Access |
|-------|----------|-------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 15:12 | MCAN2CKS | MCAN2 Communication Clock Selection | <ul style="list-style-type: none"> <li>0b0001: 2 divisions of the system clock</li> <li>0b0010: 3 divisions of the system clock</li> <li>0b0011: 4 divisions of the system clock</li> <li>0b0100: 5 divisions of the system clock</li> <li>0b0101: 6 divisions of the system clock</li> <li>0b0110: 7 divisions of the system clock</li> <li>0b0111: 8 divisions of the system clock</li> <li>0b1000: PLLHQ</li> <li>0b1001: PLLHR</li> <li>0b1010: PLLAP</li> <li>0b1011: PLLAQ</li> <li>0b1100: PLLAR</li> <li>0b1101: XTAL</li> <li>Others: Setting prohibited</li> </ul> <div>  <b>Note</b> <ul style="list-style-type: none"> <li>When the target clock source to be switched to is PLLH/PLLA, it is necessary to ensure that the PLLH/PLLA clock is in a stable state.</li> <li>When the system clock selects PLLH, it is necessary to turn off the clocks of various peripheral blocks before writing to the CMU_SCFGR register to switch the clock division. After writing to the CMU_CANCKCFGR register, software waits for at least 30μs for system stabilization time.</li> <li>When the PWC_STPMCR.CKSMRC bit is 1, after waking up from Stop Mode, this register initializes, and MCAN2CLK is the XTAL clock source.</li> </ul> </div> | RW     |
| 11:8  | MCAN1CKS | MCAN1 Communication Clock Selection | <ul style="list-style-type: none"> <li>0b0001: 2 divisions of the system clock</li> <li>0b0010: 3 divisions of the system clock</li> <li>0b0011: 4 divisions of the system clock</li> <li>0b0100: 5 divisions of the system clock</li> <li>0b0101: 6 divisions of the system clock</li> <li>0b0110: 7 divisions of the system clock</li> <li>0b0111: 8 divisions of the system clock</li> <li>0b1000: PLLHQ</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | RW     |

| Bit  | Name     | Function                            | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | Access |
|------|----------|-------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 11:8 | MCAN1CKS | MCAN1 Communication Clock Selection | <ul style="list-style-type: none"> <li>● 0b1001: PLLHR</li> <li>● 0b1010: PLLAP</li> <li>● 0b1011: PLLAQ</li> <li>● 0b1100: PLLAR</li> <li>● 0b1101: XTAL</li> <li>● Others: Setting prohibited</li> </ul> <div>  <b>Note</b> <ul style="list-style-type: none"> <li>● When the target clock source to be switched to is PLLH/PLLA, it is necessary to ensure that the PLLH/PLLA clock is in a stable state.</li> <li>● When the system clock selects PLLH, it is necessary to turn off the clocks of various peripheral blocks before writing to the CMU_SCFGR register to switch the clock division. After writing to the CMU_CANCKCFGR register, software waits for at least 30μs for system stabilization time.</li> <li>● When the PWC_STPMCR.CKSMRC bit is 1, after waking up from Stop Mode, this register initializes, and MCAN1CLK is the XTAL clock source.</li> </ul> </div>                                                                                                                                                                                                                                                                                                                                                                                                  | RW     |
| 7:4  | CAN2CKS  | CAN2 Communication Clock Selection  | <ul style="list-style-type: none"> <li>● 0b0001: 2 divisions of the system clock</li> <li>● 0b0010: 3 divisions of the system clock</li> <li>● 0b0011: 4 divisions of the system clock</li> <li>● 0b0100: 5 divisions of the system clock</li> <li>● 0b0101: 6 divisions of the system clock</li> <li>● 0b0110: 7 divisions of the system clock</li> <li>● 0b0111: 8 divisions of the system clock</li> <li>● 0b1000: PLLHQ</li> <li>● 0b1001: PLLHR</li> <li>● 0b1010: PLLAP</li> <li>● 0b1011: PLLAQ</li> <li>● 0b1100: PLLAR</li> <li>● 0b1101: XTAL</li> <li>● Others: Setting prohibited</li> </ul> <div>  <b>Note</b> <ul style="list-style-type: none"> <li>● When the target clock source to be switched to is PLLH/PLLA, it is necessary to ensure that the PLLH/PLLA clock is in a stable state.</li> <li>● When the system clock selects PLLH, it is necessary to turn off the clocks of various peripheral blocks before writing to the CMU_SCFGR register to switch the clock division. After writing to the CMU_CANCKCFGR register, software waits for at least 30μs for system stabilization time.</li> <li>● When the PWC_STPMCR.CKSMRC bit is 1, after waking up from Stop mode, this register is initialized and CAN2CLK is the XTAL clock source.</li> </ul> </div> | RW     |

| Bit | Name    | Function                           | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | Access |
|-----|---------|------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 3:0 | CAN1CKS | CAN1 Communication Clock Selection | <ul style="list-style-type: none"> <li>● 0b0001: 2 divisions of the system clock</li> <li>● 0b0010: 3 divisions of the system clock</li> <li>● 0b0011: 4 divisions of the system clock</li> <li>● 0b0100: 5 divisions of the system clock</li> <li>● 0b0101: 6 divisions of the system clock</li> <li>● 0b0110: 7 divisions of the system clock</li> <li>● 0b0111: 8 divisions of the system clock</li> <li>● 0b1000: PLLHQ</li> <li>● 0b1001: PLLHR</li> <li>● 0b1010: PLLAP</li> <li>● 0b1011: PLLAQ</li> <li>● 0b1100: PLLAR</li> <li>● 0b1101: XTAL</li> <li>● Others: Setting prohibited</li> </ul> <div>  <b>Note</b> <ul style="list-style-type: none"> <li>● When the target clock source to be switched to is PLLH/PLLA, it is necessary to ensure that the PLLH/PLLA clock is in a stable state.</li> <li>● When the system clock selects PLLH, it is necessary to turn off the clocks of various peripheral blocks before writing to the CMU_SCFGR register to switch the clock division. After writing to the CMU_CANCKCFGR register, software waits for at least 30μs for system stabilization time.</li> <li>● When the PWC_STPMCR.CKSMRC bit is 1, after waking up from Stop mode, this register is initialized and CAN1CLK is the XTAL clock source.</li> </ul> </div> | RW     |

## 5.4.28 CMU ADC/DAC Clock Config Register (CMU\_PERICKSEL)

| Offset | Bit Position |    |    |    |    |    |   |   |   |   |   |   |           |   |   |   |
|--------|--------------|----|----|----|----|----|---|---|---|---|---|---|-----------|---|---|---|
| 0xC010 | 15           | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3         | 2 | 1 | 0 |
| Reset  | 0x0000       |    |    |    |    |    |   |   |   |   |   |   |           |   |   |   |
| Name   | Reserved     |    |    |    |    |    |   |   |   |   |   |   | PERICKSEL |   |   |   |
| Access |              |    |    |    |    |    |   |   |   |   |   |   | RW        |   |   |   |

| Bit  | Name      | Function                       | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | Access |
|------|-----------|--------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 15:4 | Reserved  | Reserved                       | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | -      |
| 3:0  | PERICKSEL | ADC/DAC Clock Source Selection | <ul style="list-style-type: none"> <li>0b0000: PCLK2/PCLK4 set by CMU_SCFGR</li> <li>0b1000: PCLK2/PCLK4 configured as PLLHQ</li> <li>0b1001: PCLK2/PCLK4 configured as PLLHR</li> <li>0b1010: PCLK2/PCLK4 configured as PLLAP</li> <li>0b1011: PCLK2/PCLK4 configured as PLLAQ</li> <li>0b1100: PCLK2/PCLK4 configured as PLLAR</li> <li>Others: Setting prohibited</li> </ul> <div>  <b>Note</b> <ul style="list-style-type: none"> <li>When the target clock source to be switched to is PLLH/PLLA, it is necessary to ensure that the PLLH/PLLA clock is in a stable state.</li> </ul> </div> | RW     |

## 5.4.29 CMU Debug Clock Config Register (CMU\_TPIUCKCFGR)

| Offset | Bit Position |          |   |   |   |   |         |   |
|--------|--------------|----------|---|---|---|---|---------|---|
| 0xC03F | 7            | 6        | 5 | 4 | 3 | 2 | 1       | 0 |
| Reset  | 0x00         |          |   |   |   |   |         |   |
| Name   | TPIUCKOE     | Reserved |   |   |   |   | TPIUCKS |   |
| Access | RW           |          |   |   |   |   | RW      |   |

| Bit | Name            | Function                          | Description                                                                                                                                                         | Access |
|-----|-----------------|-----------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 7   | TPIUCKOE        | TPIU Clock Supply Enable Bit      | <ul style="list-style-type: none"> <li>● 0b0: Disable</li> <li>● 0b1: Enable</li> </ul>                                                                             | RW     |
| 6:2 | <i>Reserved</i> | <i>Reserved</i>                   | -                                                                                                                                                                   | -      |
| 1:0 | TPIUCKS         | TPIU Clock Division Selection Bit | <ul style="list-style-type: none"> <li>● 0b00: Divide by 1</li> <li>● 0b01: Divide by 2</li> <li>● 0b10: Divide by 4</li> <li>● 0b11: Setting prohibited</li> </ul> | RW     |



## 5.4.30 CMU I2S Clock Config Register (CMU\_I2SCKCFGR)

| Offset | Bit Position |    |    |    |           |    |   |   |           |   |   |   |           |   |   |   |
|--------|--------------|----|----|----|-----------|----|---|---|-----------|---|---|---|-----------|---|---|---|
| 0xC012 | 15           | 14 | 13 | 12 | 11        | 10 | 9 | 8 | 7         | 6 | 5 | 4 | 3         | 2 | 1 | 0 |
| Reset  | 0xB BBBB     |    |    |    |           |    |   |   |           |   |   |   |           |   |   |   |
| Name   | I2S4CKSEK    |    |    |    | I2S3CKSEK |    |   |   | I2S2CKSEK |   |   |   | I2S1CKSEK |   |   |   |
| Access | RW           |    |    |    | RW        |    |   |   | RW        |   |   |   | RW        |   |   |   |

| Bit   | Name      | Function             | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | Access |
|-------|-----------|----------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 15:12 | I2S4CKSEK | I2S4 Clock Selection | <ul style="list-style-type: none"> <li>0b0000: PCLK1 set by CMU_SCFGR</li> <li>0b1000: PLLHQ</li> <li>0b1001: PLLHR</li> <li>0b1010: PLLAP</li> <li>0b1011: PLLAQ</li> <li>0b1100: PLLAR</li> <li>Others: Setting prohibited</li> </ul> <div>  <b>Note</b> <ul style="list-style-type: none"> <li>When the target clock source to be switched to is PLLH/PLLA, it is necessary to ensure that the PLLH/PLLA clock is in a stable state.</li> </ul> </div>  | RW     |
| 11:8  | I2S3CKSEK | I2S3 Clock Selection | <ul style="list-style-type: none"> <li>0b0000: PCLK1 set by CMU_SCFGR</li> <li>0b1000: PLLHQ</li> <li>0b1001: PLLHR</li> <li>0b1010: PLLAP</li> <li>0b1011: PLLAQ</li> <li>0b1100: PLLAR</li> <li>Others: Setting prohibited</li> </ul> <div>  <b>Note</b> <ul style="list-style-type: none"> <li>When the target clock source to be switched to is PLLH/PLLA, it is necessary to ensure that the PLLH/PLLA clock is in a stable state.</li> </ul> </div> | RW     |

| Bit | Name      | Function             | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | Access |
|-----|-----------|----------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 7:4 | I2S2CKSEK | I2S2 Clock Selection | <ul style="list-style-type: none"> <li>● 0b0000: PCLK1 set by CMU_SCFGR</li> <li>● 0b1000: PLLHQ</li> <li>● 0b1001: PLLHR</li> <li>● 0b1010: PLLAP</li> <li>● 0b1011: PLLAQ</li> <li>● 0b1100: PLLAR</li> <li>● Others: Setting prohibited</li> </ul> <div>  <b>Note</b> <ul style="list-style-type: none"> <li>● When the target clock source to be switched to is PLLH/PLLA, it is necessary to ensure that the PLLH/PLLA clock is in a stable state.</li> </ul> </div>  | RW     |
| 3:0 | I2S1CKSEK | I2S1 Clock Selection | <ul style="list-style-type: none"> <li>● 0b0000: PCLK1 set by CMU_SCFGR</li> <li>● 0b1000: PLLHQ</li> <li>● 0b1001: PLLHR</li> <li>● 0b1010: PLLAP</li> <li>● 0b1011: PLLAQ</li> <li>● 0b1100: PLLAR</li> <li>● Others: Setting prohibited</li> </ul> <div>  <b>Note</b> <ul style="list-style-type: none"> <li>● When the target clock source to be switched to is PLLH/PLLA, it is necessary to ensure that the PLLH/PLLA clock is in a stable state.</li> </ul> </div> | RW     |

## 5.4.31 CMU MCO1 Clock Output Config Register (CMU\_MCO1CFGR)

| Offset | Bit Position |         |   |   |         |   |   |   |
|--------|--------------|---------|---|---|---------|---|---|---|
| 0xC03D | 7            | 6       | 5 | 4 | 3       | 2 | 1 | 0 |
| Reset  | 0x00         |         |   |   |         |   |   |   |
| Name   | MCO1EN       | MCO1DIV |   |   | MCO1SEL |   |   |   |
| Access | RW           | RW      |   |   | RW      |   |   |   |

| Bit | Name    | Function                           | Description                                                                                                                                                                                                                                                                                                                                                              | Access |
|-----|---------|------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 7   | MCO1EN  | MCO_1 Output Permission            | <ul style="list-style-type: none"> <li>0b0: Disable</li> <li>0b1: Enable</li> </ul>                                                                                                                                                                                                                                                                                      | RW     |
| 6:4 | MCO1DIV | MCO_1 Frequency Division Selection | <ul style="list-style-type: none"> <li>0b000: Divide by 1</li> <li>0b001: Divide by 2</li> <li>0b010: Divide by 4</li> <li>0b011: Divide by 8</li> <li>0b100: Divide by 16</li> <li>0b101: Divide by 32</li> <li>0b110: Divide by 64</li> <li>0b111: Divide by 128</li> </ul>                                                                                            | RW     |
| 3:0 | MCO1SEL | MCO_1 Clock Source Selection       | <ul style="list-style-type: none"> <li>0b0000: HRC Clock</li> <li>0b0001: MRC Clock</li> <li>0b0010: LRC Clock</li> <li>0b0011: XTAL Clock</li> <li>0b0100: XTAL32 clock</li> <li>0b0110: PLLHP</li> <li>0b0111: PLLAP</li> <li>0b1000: PLLHQ</li> <li>0b1001: PLLAQ</li> <li>0b1010: PLLAR</li> <li>0b1011: System clock</li> <li>Others: Setting prohibited</li> </ul> | RW     |

## 5.4.32 CMU MCO2 Clock Output Config Register (CMU\_MCO2CFGR)

| Offset | Bit Position |         |   |   |         |   |   |   |
|--------|--------------|---------|---|---|---------|---|---|---|
| 0xC03E | 7            | 6       | 5 | 4 | 3       | 2 | 1 | 0 |
| Reset  | 0x00         |         |   |   |         |   |   |   |
| Name   | MCO2EN       | MCO2DIV |   |   | MCO2SEL |   |   |   |
| Access | RW           | RW      |   |   | RW      |   |   |   |

| Bit | Name    | Function                           | Description                                                                                                                                                                                                                                                                                                                                                              | Access |
|-----|---------|------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 7   | MCO2EN  | MCO_2 Output Permission            | <ul style="list-style-type: none"> <li>0b0: Disable</li> <li>0b1: Enable</li> </ul>                                                                                                                                                                                                                                                                                      | RW     |
| 6:4 | MCO2DIV | MCO_2 Frequency Division Selection | <ul style="list-style-type: none"> <li>0b000: Divide by 1</li> <li>0b001: Divide by 2</li> <li>0b010: Divide by 4</li> <li>0b011: Divide by 8</li> <li>0b100: Divide by 16</li> <li>0b101: Divide by 32</li> <li>0b110: Divide by 64</li> <li>0b111: Divide by 128</li> </ul>                                                                                            | RW     |
| 3:0 | MCO2SEL | MCO_2 Clock Source Selection       | <ul style="list-style-type: none"> <li>0b0000: HRC Clock</li> <li>0b0001: MRC Clock</li> <li>0b0010: LRC Clock</li> <li>0b0011: XTAL Clock</li> <li>0b0100: XTAL32 clock</li> <li>0b0110: PLLHP</li> <li>0b0111: PLLAP</li> <li>0b1000: PLLHQ</li> <li>0b1001: PLLAQ</li> <li>0b1010: PLLAR</li> <li>0b1011: System clock</li> <li>Others: Setting prohibited</li> </ul> | RW     |

### 5.4.33 FCM Lower Limit Control Register (FCMn\_LVR) (n=1-4)

Offset: 0x00+0x40\*(n-1)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |     |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| -      | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15  | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |     |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | LVR |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RW  |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit   | Name     | Function                  | Description                                      | Access |
|-------|----------|---------------------------|--------------------------------------------------|--------|
| 31:16 | Reserved | Reserved                  | -                                                | -      |
| 15:0  | LVR      | Lower Limit Compare Value | Configure this register when the START bit is 0. | RW     |

### 5.4.34 FCM Upper Limit Control Register (FCMn\_UVR) (n=1-4)

Offset: 0x04+0x40\*(n-1)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |     |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| -      | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15  | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |     |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | UVR |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |     |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit   | Name     | Function                  | Description                                      | Access |
|-------|----------|---------------------------|--------------------------------------------------|--------|
| 31:16 | Reserved | Reserved                  | -                                                | -      |
| 15:0  | UVR      | Upper Limit Compare Value | Configure this register when the START bit is 0. | RW     |

5.4.35 FCM Counter Register (FCM<sub>n</sub>\_CNTR) (n=1-4)

Offset: 0x08+0x40\*(n-1)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |      |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| -      | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15   | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |      |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | CNTR |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RO   |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit   | Name     | Function      | Description                                                                                                                                                         | Access |
|-------|----------|---------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:16 | Reserved | Reserved      | -                                                                                                                                                                   | -      |
| 15:0  | CNTR     | Counter Value | Save the counter value to this register when a active edge selected by the EDGES bit of the base clock is detected (except for the first active edge after START=1) | RO     |

## 5.4.36 FCM Start Control Register (FCM\_STR)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |        |        |        |        |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|--------|--------|--------|--------|
| 0x0C   | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3      | 2      | 1      | 0      |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |        |        |        |        |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   | START4 | START3 | START2 | START1 |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   | RW     | RW     | RW     | RW     |

| Bit  | Name                          | Function                                  | Description                                                                                                                 | Access |
|------|-------------------------------|-------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------|--------|
| 31:4 | Reserved                      | Reserved                                  | -                                                                                                                           | -      |
| 3:0  | START <sub>n</sub><br>(n=4-1) | Channel n Frequency Measurement Start Bit | <ul style="list-style-type: none"> <li>0b0: Frequency measurement stop</li> <li>0b1: Frequency measurement start</li> </ul> | RW     |

5.4.37 FCM Measurement Control Register (FCM<sub>n</sub>\_MCCR) (n=1-4)

Offset: 0x10+0x40\*(n-1)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |      |   |   |   |          |   |       |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|------|---|---|---|----------|---|-------|---|
| -      | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7    | 6 | 5 | 4 | 3        | 2 | 1     | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |      |   |   |   |          |   |       |   |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   | MCKS |   |   |   | Reserved |   | MDIVS |   |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   | RW   |   |   |   |          |   | RW    |   |

| Bit  | Name     | Function                               | Description                                                                                                                                                                                                                                                                                                     | Access |
|------|----------|----------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:8 | Reserved | Reserved                               | -                                                                                                                                                                                                                                                                                                               | -      |
| 7:4  | MCKS     | Measurement Object Clock Selection Bit | <ul style="list-style-type: none"> <li>0b0000: XTAL</li> <li>0b0001: XTAL32</li> <li>0b0010: HRC</li> <li>0b0011: LRC</li> <li>0b0100: SWDTLRC</li> <li>0b0101: PCLK1</li> <li>0b0110: PLLAP</li> <li>0b0111: MRC</li> <li>0b1000: PLLHP</li> <li>0b1001: RTCLRC</li> <li>Others: Setting prohibited</li> </ul> | RW     |
| 3:2  | Reserved | Reserved                               | -                                                                                                                                                                                                                                                                                                               | -      |
| 1:0  | MDIVS    | Measurement Object Division Selection  | <ul style="list-style-type: none"> <li>0b00: No division</li> <li>0b01: Divide by 4</li> <li>0b10: Divide by 8</li> <li>0b11: Divide by 32</li> </ul>                                                                                                                                                           | RW     |

5.4.38 FCM Reference Control Register (FCM<sub>n</sub>\_RCCR) (n=1-4)

Offset: 0x14+0x40\*(n-1)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |        |          |       |          |      |       |      |   |   |   |          |       |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|--------|----------|-------|----------|------|-------|------|---|---|---|----------|-------|---|---|---|---|
| -      | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15     | 14       | 13    | 12       | 11   | 10    | 9    | 8 | 7 | 6 | 5        | 4     | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |        |          |       |          |      |       |      |   |   |   |          |       |   |   |   |   |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | EXREFE | Reserved | EDGES | Reserved | DNFS | INEXS | RCKS |   |   |   | Reserved | RDIVS |   |   |   |   |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RW     |          | RW    |          | RW   | RW    | RW   |   |   |   |          | RW    |   |   |   |   |

| Bit   | Name     | Function                                                          | Description                                                                                                                                                                                                                                                                                     | Access |
|-------|----------|-------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:16 | Reserved | Reserved                                                          | -                                                                                                                                                                                                                                                                                               | -      |
| 15    | EXREFE   | External Pin Input Reference Clock FCMREF Enable Bit              | <ul style="list-style-type: none"> <li>0b0: Disable external pin input reference clock FCMREF</li> <li>0b1: Enable external pin input reference clock FCMREF</li> </ul>                                                                                                                         | RW     |
| 14    | Reserved | Reserved                                                          | -                                                                                                                                                                                                                                                                                               | -      |
| 13:12 | EDGES    | Measurement Reference Edge Selection Bit                          | <ul style="list-style-type: none"> <li>0b00: Rising edge</li> <li>0b01: Falling Edge</li> <li>0b10: Rising and falling edges</li> <li>0b11: Setting prohibited</li> </ul>                                                                                                                       | RW     |
| 11:10 | Reserved | Reserved                                                          | -                                                                                                                                                                                                                                                                                               | -      |
| 9:8   | DNFS     | Digital Filtering Function Selection Bit                          | <ul style="list-style-type: none"> <li>0b00: No filter function</li> <li>0b01: Clock selected by MCKS bit as filter clock</li> <li>0b10: 4 divisions of the clock selected by MCKS bit as filter clock</li> <li>0b11: 16 divisions of the clock selected by MCKS bit as filter clock</li> </ul> | RW     |
| 7     | INEXS    | Measurement Reference, Internal Clock, and Terminal Selection Bit | <ul style="list-style-type: none"> <li>0b0: External pin input reference clock FCMREF</li> <li>0b1: Clock selected by RCKS bit</li> </ul>                                                                                                                                                       | RW     |



| Bit | Name            | Function                                  | Description                                                                                                                                                                                                                                                                                                                           | Access |
|-----|-----------------|-------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 6:3 | RCKS            | Measurement Reference Clock Selection Bit | <ul style="list-style-type: none"> <li>● 0b0000: XTAL</li> <li>● 0b0001: XTAL32</li> <li>● 0b0010: HRC</li> <li>● 0b0011: LRC</li> <li>● 0b0100: SWDTLRC</li> <li>● 0b0101: PCLK1</li> <li>● 0b0110: PLLAP</li> <li>● 0b0111: MRC</li> <li>● 0b1000: PLLHP</li> <li>● 0b1001: RTCLRC</li> <li>● Others: Setting prohibited</li> </ul> | RW     |
| 2   | <i>Reserved</i> | <i>Reserved</i>                           | -                                                                                                                                                                                                                                                                                                                                     | -      |
| 1:0 | RDIVS           | Measurement Reference Division Selection  | <ul style="list-style-type: none"> <li>● 0b00: Divide by 32</li> <li>● 0b01: Divide by 128</li> <li>● 0b10: Divide by 1024</li> <li>● 0b11: Divide by 8192</li> </ul>                                                                                                                                                                 | RW     |

## 5.4.39 FCM Reset Interrupt Control Register (FCMn\_RIER) (n=1-4)

Offset: 0x18+0x40\*(n-1)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |       |   |          |   |          |   |          |   |       |  |        |  |       |  |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|-------|---|----------|---|----------|---|----------|---|-------|--|--------|--|-------|--|
| -      | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7     | 6 | 5        | 4 | 3        | 2 | 1        | 0 |       |  |        |  |       |  |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |       |   |          |   |          |   |          |   |       |  |        |  |       |  |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   | ERRRE |   | Reserved |   | ERRINTRS |   | Reserved |   | OVFIE |  | MENDIE |  | ERRIE |  |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   | RW    |   |          |   | RW       |   |          |   | RW    |  | RW     |  | RW    |  |

| Bit  | Name     | Function                                          | Description                                                                                                                                                             | Access |
|------|----------|---------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:8 | Reserved | Reserved                                          | -                                                                                                                                                                       | -      |
| 7    | ERRRE    | Frequency Exception Reset Enable Bit              | <ul style="list-style-type: none"> <li>0b0: Disable</li> <li>0b1: Enable</li> </ul>                                                                                     | RW     |
| 6:5  | Reserved | Reserved                                          | -                                                                                                                                                                       | -      |
| 4    | ERRINTRS | Frequency Exception Interrupt Reset Selection Bit | <ul style="list-style-type: none"> <li>0b0: Interrupt occurs when frequency exception occurs</li> <li>0b1: Reset occurs when frequency exception occurs</li> </ul>      | RW     |
| 3    | Reserved | Reserved                                          | -                                                                                                                                                                       | -      |
| 2    | OVFIE    | Counter Overflow Interrupt Enable Bit             | <ul style="list-style-type: none"> <li>0b0: Disable counter overflow interrupt</li> <li>0b1: Enable counter overflow interrupt</li> </ul>                               | RW     |
| 1    | MENDIE   | Measurement End Interrupt Enable Bit              | <ul style="list-style-type: none"> <li>0b0: Disable interrupt when measurement ends</li> <li>0b1: Enable interrupt when measurement ends</li> </ul>                     | RW     |
| 0    | ERRIE    | Frequency Exception Interrupt Enable Bit          | <ul style="list-style-type: none"> <li>0b0: Disable interrupt when frequency exception occurs</li> <li>0b1: Enable interrupt when frequency exception occurs</li> </ul> | RW     |

## 5.4.40 FCM Flag Register (FCM\_SR)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |      |    |        |    |       |    |      |    |        |    |
|--------|--------------|----|----|----|----|----|----|----|----|----|------|----|--------|----|-------|----|------|----|--------|----|
| 0x1C   | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21   | 20 | 19     | 18 | 17    | 16 | 15   | 14 | 13     | 12 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |      |    |        |    |       |    |      |    |        |    |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    | OVF4 |    | MENDF4 |    | ERRF4 |    | OVF3 |    | MENDF3 |    |
| Access |              |    |    |    |    |    |    |    |    |    | RO   | RO | RO     | RO | RO    | RO | RO   | RO | RO     | RO |

| Bit   | Name     | Function                               | Description                                                                                                                       | Access |
|-------|----------|----------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:12 | Reserved | Reserved                               | -                                                                                                                                 | -      |
| 11    | OVF4     | Channel 4 Counter Overflow Flag Bit    | <ul style="list-style-type: none"> <li>0b0: Counter has not overflowed</li> <li>0b1: Counter has overflowed</li> </ul>            | RO     |
| 10    | MENDF4   | Channel 4 Measurement End Flag Bit     | <ul style="list-style-type: none"> <li>0b0: In measurement</li> <li>0b1: Measurement end</li> </ul>                               | RO     |
| 9     | ERRF4    | Channel 4 Frequency Exception Flag Bit | <ul style="list-style-type: none"> <li>0b0: No frequency exception occurred</li> <li>0b1: Frequency exception occurred</li> </ul> | RO     |
| 8     | OVF3     | Channel 3 Counter Overflow Flag Bit    | <ul style="list-style-type: none"> <li>0b0: Counter has not overflowed</li> <li>0b1: Counter has overflowed</li> </ul>            | RO     |
| 7     | MENDF3   | Channel 3 Measurement End Flag Bit     | <ul style="list-style-type: none"> <li>0b0: In measurement</li> <li>0b1: Measurement end</li> </ul>                               | RO     |
| 6     | ERRF3    | Channel 3 Frequency Exception Flag Bit | <ul style="list-style-type: none"> <li>0b0: No frequency exception occurred</li> <li>0b1: Frequency exception occurred</li> </ul> | RO     |
| 5     | OVF2     | Channel 2 Counter Overflow Flag Bit    | <ul style="list-style-type: none"> <li>0b0: Counter has not overflowed</li> <li>0b1: Counter has overflowed</li> </ul>            | RO     |
| 4     | MENDF2   | Channel 2 Measurement End Flag Bit     | <ul style="list-style-type: none"> <li>0b0: In measurement</li> <li>0b1: Measurement end</li> </ul>                               | RO     |
| 3     | ERRF2    | Channel 2 Frequency Exception Flag Bit | <ul style="list-style-type: none"> <li>0b0: No frequency exception occurred</li> <li>0b1: Frequency exception occurred</li> </ul> | RO     |
| 2     | OVF1     | Channel 1 Counter Overflow Flag Bit    | <ul style="list-style-type: none"> <li>0b0: Counter has not overflowed</li> <li>0b1: Counter has overflowed</li> </ul>            | RO     |
| 1     | MENDF1   | Channel 1 Measurement End Flag Bit     | <ul style="list-style-type: none"> <li>0b0: In measurement</li> <li>0b1: Measurement end</li> </ul>                               | RO     |

| Bit | Name  | Function                               | Description                                                                                                                        | Access |
|-----|-------|----------------------------------------|------------------------------------------------------------------------------------------------------------------------------------|--------|
| 0   | ERRF1 | Channel 1 Frequency Exception Flag Bit | <ul style="list-style-type: none"><li>● 0b0: No frequency exception occurred</li><li>● 0b1: Frequency exception occurred</li></ul> | RO     |

5.4.41 FCM Flag Clear Register (FCM<sub>n</sub>\_CLR) (n=1-4)

Offset: 0x20+0x40\*(n-1)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |        |          |         |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|--------|----------|---------|
| '      | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |        |          |         |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |        |          |         |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   | OVFCLR | MENDFCLR | ERRFCLR |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   | WO     | WO       | WO      |

| Bit  | Name     | Function                       | Description                                       | Access |
|------|----------|--------------------------------|---------------------------------------------------|--------|
| 31:3 | Reserved | Reserved                       | -                                                 | -      |
| 2    | OVFCLR   | Counter Over-flow Flag Clear   | Write 1 to clear the flag, write 0 has no effect. | WO     |
| 1    | MENDFCLR | Measurement End Flag Clear     | Write 1 to clear the flag, write 0 has no effect. | WO     |
| 0    | ERRFCLR  | Frequency Exception Flag Clear | Write 1 to clear the flag, write 0 has no effect. | WO     |

## 6 Power Control (PWC)

### 6.1 Introduction

Power controller is used to control the supply, switching and detection of multiple power domains in multiple run modes and low power modes. The power controller consists of a power consumption control logic (PWCL), a power supply voltage detection unit (PVD), and a battery backup control module (BATBKUP) that automatically switches between VCC and battery power.

The operating voltage (VCC) of the chip is 1.8V to 3.63V. The voltage regulator (LDO) supplies power to the VDD and VDDR domains, and the VDDR voltage regulator (RLDO) supplies power to the VDDR domain in Power-down mode or when VCC is powered off. The chip provides two operating modes: high-speed and ultra-low-speed, and three low-power modes: sleep, stop and power-off through the power consumption control logic (PWCL).

The power voltage detection unit (PVD) provides functions such as power on reset (POR), power down reset (PDR), brown out reset (BOR), programmable voltage detection 1 (PVD1), programmable voltage detection 2 (PVD2), reference voltage measurement path, VBAT voltage detection, and VBAT voltage measurement. Among them, POR, PDR, and BOR control the chip reset action by detecting the VCC voltage. PVD1 detects the VCC voltage and resets or interrupts the chip according to the register settings. PVD2 detects VCC voltage or external input detection voltage, and generates reset or interrupt according to register selection. The reference voltage measurement path is a function for measuring the reference voltage using ADC. VBAT voltage detection is a function that reads the register to obtain the VBAT voltage is higher or lower than the VBAT detection voltage. The VBAT detection voltage can be selected as 1.8V or 2.1V using the register. The VBAT voltage measurement function refers to the function of using ADC to measure the 1/2 voltage division of VBAT to obtain the VBAT voltage value.

The battery backup domain maintains power through VBAT when VCC is powered off, ensuring that the real-time clock module (RTC) and wake-up timer (WKTm) can continue to operate and provide power for RLDO. When the chip enters Power-down mode or VCC loses power, the VDDR area can maintain power through RLDO to keep the data of 4KB Ret SRAM.

The analog blocks are equipped with dedicated supply pins for improved analog performance.

### 6.2 Functional Description

#### 6.2.1 Power Distribution

##### 6.2.1.1 Power Distribution Structure

The chip consists of a VCC domain, a VDD power domain, an AVCC power domain, a battery backup domain, and a VDDR domain.

- The VCC domain is powered by the VCC/VSS pins and consists of power control logic (PWCL), power voltage detection unit (PVD), IO level holding circuit, voltage regulator (LDO), external high-speed oscillator (XTAL), internal low-speed oscillator (LRC) and other circuits.
- The VDD power domain is composed of digital logic such as CPU, digital peripherals, RAM, FLASH, etc., and is powered by VDD generated by LDO. The RAM in the VDD domain is divided into 5 groups, consisting of 11 independent modules, and the power-off of each module can be independently controlled by registers.

- The AVCC power domain is the analog power domain, which is mainly composed of analog-to-digital converter (ADC), digital-to-analog converter (DAC), comparator (CMP), programmable gain amplifier (PGA), and input and output pins of the analog system, and is powered by the AVCC/AVSS pin. In order to provide high-precision analog performance, the analog area is equipped with an independent power supply. To ensure higher accuracy of the ADC, the reference voltage of the ADC uses dedicated pins (VREFH, VREFL).
- The BATBKUP power domain is a battery backup domain. The power supply in this area is automatically selected by the power switch to be supplied by VCC or VBAT. When VCC is powered off, the power supply of the backup domain automatically switches to VBAT. The battery backup domain consists of the real-time clock (RTC), the wake-up timer (WKTm), the external low-speed oscillator (XTAL32), and the internal low-speed oscillator for RTC (RTCLRC).
- The VDDR domain consists of 4KB Ret SRAM. Ret SRAM is powered by RLDO. In Power-down mode, Ret SRAM can retain data. When Ret SRAM is not needed, PWC\_VBATCR.VBTRSD can be set to put the chip into Power-down mode or cut off the VDDR domain power supply when VCC is powered off to further reduce power consumption.

The figure below is the power distribution diagram of the chip.

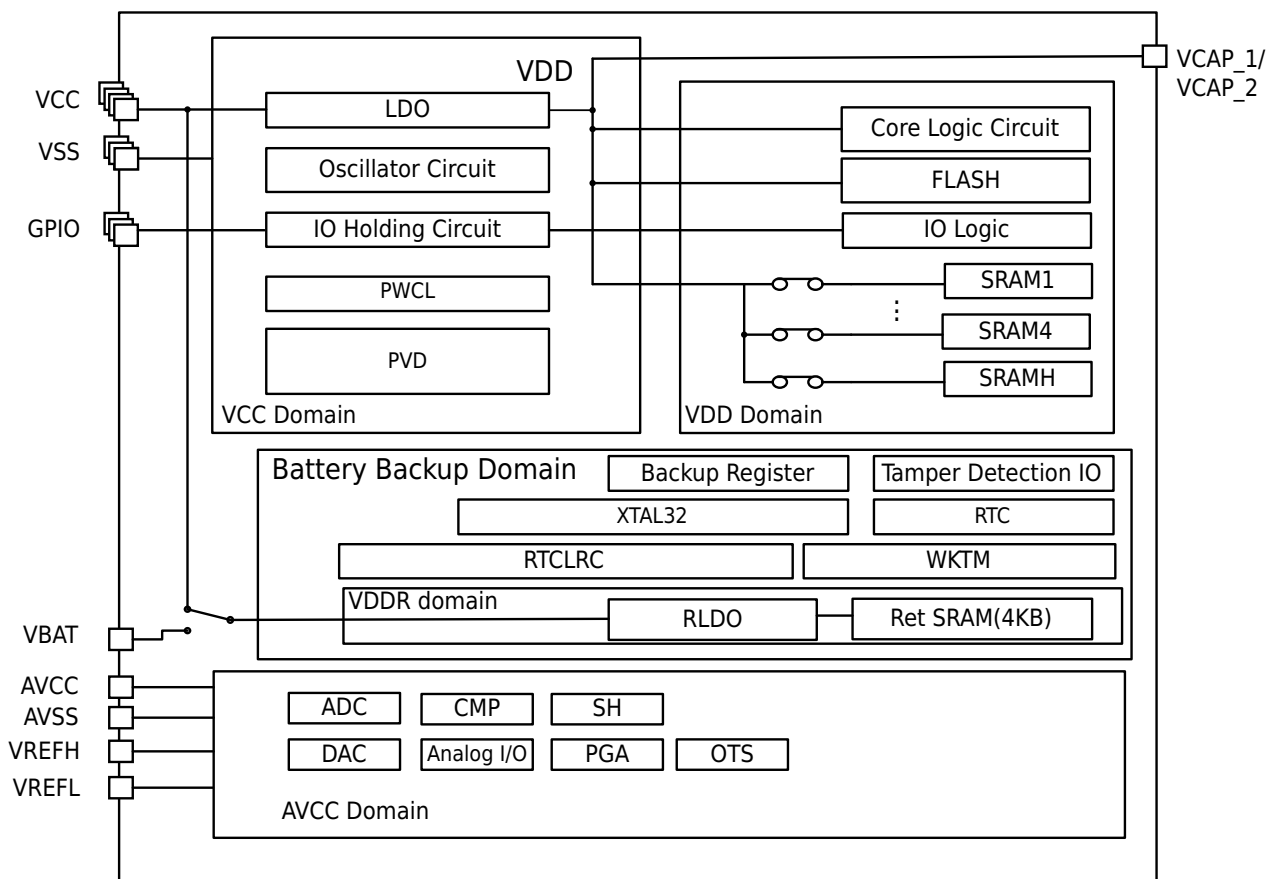


Figure 6-1 Power Supply Configuration Diagram

### 6.2.1.2 Battery Backup Power Domain

The battery backup domain uses an internal power switch to select VCC power supply or VBAT (battery) power supply. The backup domain includes RTC (real-time clock), XTAL32 (external low-speed oscillator), 128-byte backup registers, 4K-byte Ret SRAM, internal low-speed oscillator, and 2 pins RTCIC0-1. When the voltage of the VCC pin decreases, the real-time clock (RTC) and the internal low-speed oscillator of the RTC are powered by the dedicated battery backup power pin (VBAT pin). For applications without an

external battery, it is recommended to connect the VBAT pin to the VCC pin through an external 100nF ceramic decoupling capacitor.

The switching operation of the battery backup function is shown in the following figure. When the VCC voltage drops below  $V_{PDR}$ , the power backup domain switches to be powered by VBAT; when the VCC voltage rises above  $V_{POR}$ , the power backup domain switches to be powered by VCC. If the chip uses Power-down mode 3 or Power-down mode 4, the detection level of  $V_{PDR}$  cannot be determined. When the VCC voltage drops, it cannot be guaranteed that the chip can correctly switch to the VBAT power supply. Therefore, the chip cannot use Power-down mode 3 and battery backup function at the same time, nor can it use Power-down mode 4 and battery backup function at the same time.

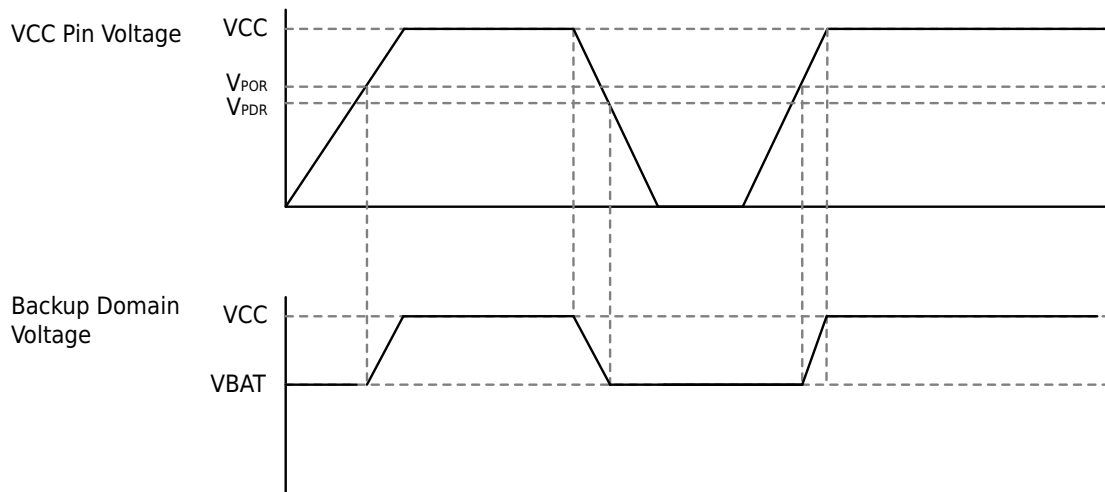


Figure 6-2 Battery Backup Domain Power Switching Diagram

The clock source of the RTC can be the internal low-speed oscillator (RTCLRC) for the RTC or an external low-speed oscillator (XTAL32). Before entering power saving mode through WFI/WFE instructions, Cortex-M4 needs to set the expected wake-up time through the RTC register and enable the wake-up function to implement the RTC timer wake-up event.

Ret SRAM can still retain data in Power-down mode or even after VCC power is lost, so if VBAT is continuously powered, Ret SRAM can be regarded as an internal EEPROM. When an intrusion event occurs, the software setting can be used to select whether the Ret SRAM should be powered off and whether the backup register needs to be initialized. Ret SRAM can be turned off by setting the VBTRSD bit in register PWC\_VBATCR. After the Ret SRAM is turned off, circuits such as a voltage regulator that supplies power to the Ret SRAM are also turned off, which can further reduce power consumption. After the VBTRSD bit is set, the SRAM cannot perform read and write operations until the RAMVALID flag bit becomes 1.

When using the battery backup domain, you need to write the PWC\_VBATRSTR register to initialize the VBAT domain. After PWC\_VBATRSTR writes 0xA5, the external low-speed oscillator XTAL32, the RLDO for Ret SRAM is started, and the wake-up timer WKTM is initialized.

### 6.2.2 Power Voltage Detection Unit (PVD) description

The power voltage detection unit (PVD) includes power on reset (POR), power down reset (PDR), brown out reset (BOR), programmable voltage detection 1 (PVD1), programmable voltage detection 2 (PVD2), reference voltage measurement path, VBAT voltage detection, VBAT voltage measurement and other functions.



### 6.2.2.1 Power-on/Power-off Reset Action Description

The chip is integrated with power-on reset circuit and power-off reset circuit. The power-on reset and power-off reset waveforms are as shown. When  $V_{CC}$  is higher than the specified threshold  $V_{POR}$ , after the  $T_{RSTPOR}$  time, the chip releases the power-on reset state, and the CPU starts to execute code. When  $V_{CC}$  is lower than  $V_{PDR}$ , the chip remains in reset state. When power-on reset is used, the reset pin NRST must be 1. If the reset pin is pulled down, the chip will be reset and started by pin reset.

For detailed information on parameters such as  $V_{POR}$ ,  $V_{PDR}$ , and  $T_{RSTPOR}$ , refer to the "Electrical Characteristics" chapter of the Datasheet.

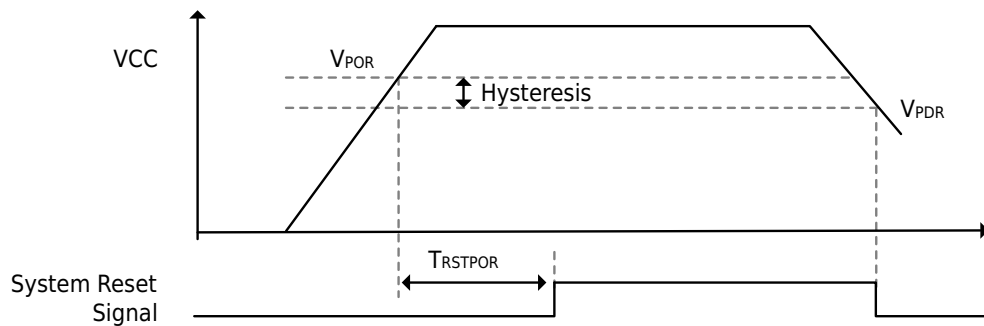


Figure 6-3 Power-on Reset, Power-off Reset Waveforms

### 6.2.2.2 Brown-out Reset (BOR) Description

During power-up, until  $V_{CC}$  is higher than  $V_{BOR}$ , a brown-out reset (BOR) will put the chip in reset.

The  $V_{BOR}$  threshold is configured through the BOR\_LEV, BORDIS of the initialization configuration bits (ICG). When BORDIS=0b0, the BOR detection voltage can be selected from 4 thresholds. When BORDIS is configured as 0b1, the chip performs reset control through power-on reset and power-off reset.

Table 6-1 BOR Configuration

| BORDIS | BOR_LEV | Description                                |
|--------|---------|--------------------------------------------|
| 0b1    | 0bXX    | BOR Inactive                               |
| 0b0    | 0b00    | BOR Active, Select BOR Threshold 0 (VBOR0) |
| 0b0    | 0b01    | BOR Active, Select BOR Threshold 1 (VBOR1) |
| 0b0    | 0b10    | BOR Active, Select BOR Threshold 2 (VBOR2) |
| 0b0    | 0b11    | BOR Active, Select BOR Threshold 3 (VBOR3) |

For the electrical characteristics of the BOR threshold, see the "Electrical Characteristics" chapter of the Datasheet.

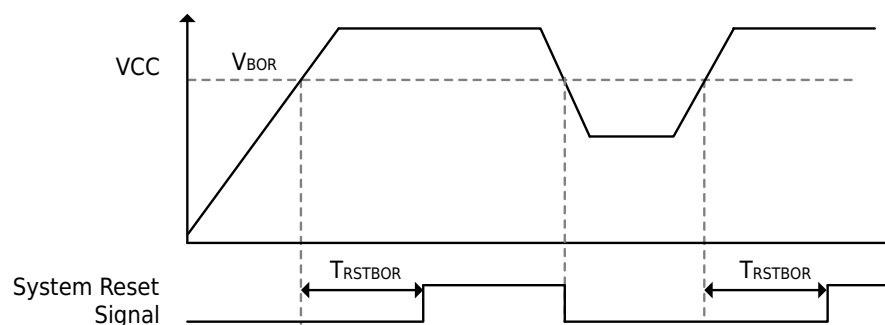


Figure 6-4 Brown-out Reset Waveform

### 6.2.2.3 Programmable Voltage Detection 1 (PVD1), Programmable Voltage Detection 2 (PVD2)

Programmable voltage detection 1 and programmable voltage detection 2 trigger corresponding reset or interrupt actions by detecting whether the VCC power supply voltage passes the detection threshold. Each detection circuit can be programmed and configured separately.

When the power supply voltage passes through the threshold voltage point of each detection circuit, the event can be programmed and configured as a reset/interrupt (maskable/non-maskable) /AOS trigger function.

The main features of programmable voltage detection are shown in the following table.

Table 6-2 PVD1/PVD2 Features

| Item                    | PVD1                                                                                                                                                                          | PVD2                                                                                                                                                                                                                                     |
|-------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Test Object             | Whether VCC has passed the threshold voltage point (VPVD1) during the falling/rising process                                                                                  | Whether VCC passes through the threshold voltage point (VPVD2) during the decline/rise process, and whether the rise and fall of the external input voltage passes through the threshold voltage point (VPVD2, PWC_PVDLCR.PVD2LVL=0b111) |
| Detection Voltage Point | Configured by PVD1LVL                                                                                                                                                         | Configured by PVD2LVL                                                                                                                                                                                                                    |
| Reset                   | Reset: $VCC < VPVD1$ ;<br>Reset release: $VCC > VPVD1$ after a certain reset processing time.                                                                                 | Reset: $VCC < VPVD2$ ;<br>Reset release: $VCC > VPVD2$ after a certain reset processing time.                                                                                                                                            |
| Interrupt               | Configured as voltage detection 1 interrupt or non-maskable interrupt                                                                                                         | Configured as voltage detection 2 interrupt or non-maskable interrupt                                                                                                                                                                    |
|                         | VCC drops through the threshold voltage point (VPVD1) or VCC rises through the threshold voltage point (VPVD1) or VCC rises/falls through the threshold voltage point (VPVD1) | VCC drops through the threshold voltage point (VPVD2) or VCC rises through the threshold voltage point (VPVD2) or VCC rises/falls through the threshold voltage point (VPVD2)                                                            |
| Filter Function         | Digital filtering                                                                                                                                                             | Digital filtering                                                                                                                                                                                                                        |
| AOS Trigger Function    | VCC drops past the threshold voltage point (VPVD1)                                                                                                                            | VCC drops past the threshold voltage point (VPVD2)                                                                                                                                                                                       |

### 6.2.2.4 PVD1, PVD2 Interrupt/Reset Block Diagram

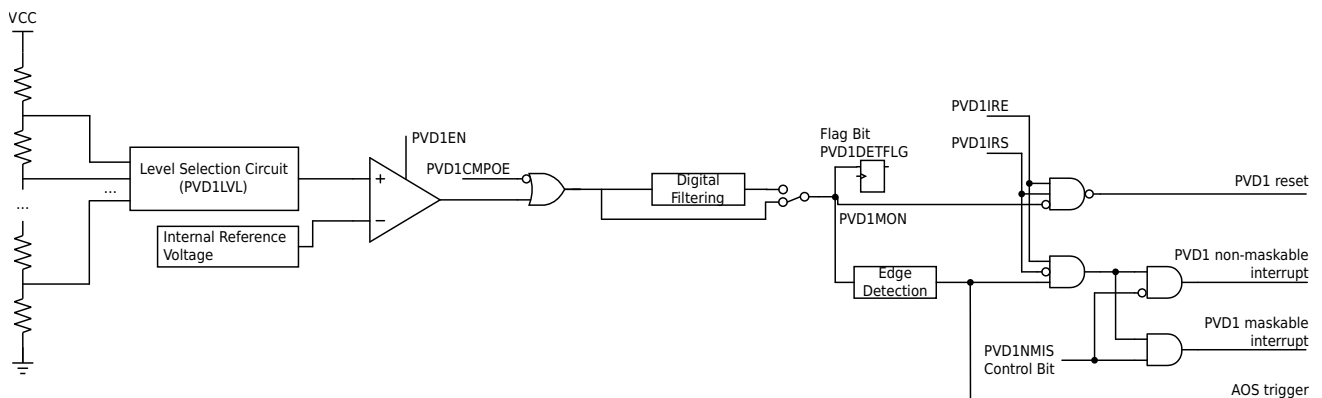
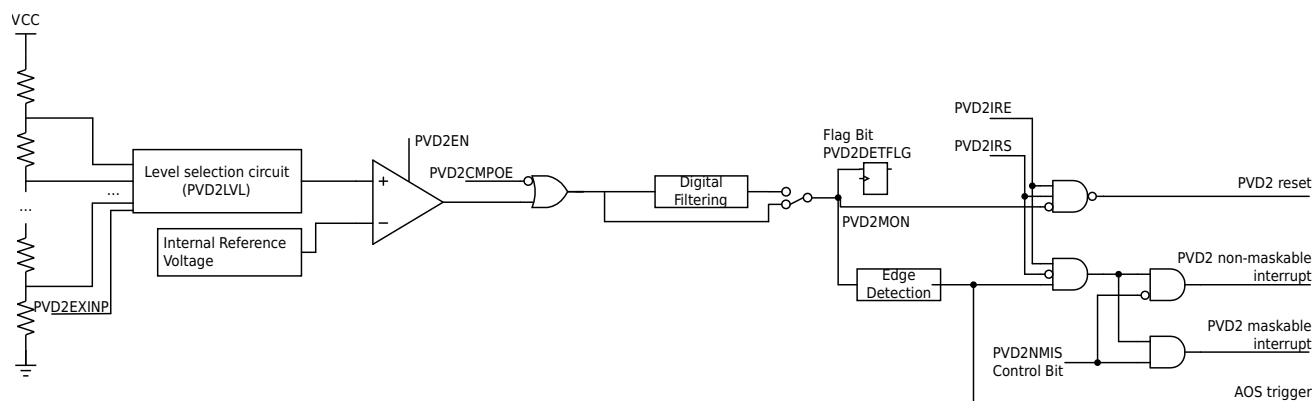


Figure 6-5 PVD1 Interrupt/Reset Block Diagram



### Figure 6-6 PVD2 Interrupt/Reset Block Diagram

#### 6.2.2.5 Input/Output Pin

| Pin Name  | Input/Output | Function                               |
|-----------|--------------|----------------------------------------|
| PVD2EXINP | Input        | External input PVD2 comparison voltage |

#### 6.2.2.6 PVD1 Interrupt and Reset

When using the PVD1 circuit in Stop mode or Power-down mode, please observe the following precautions:

1. Stop Mode
  - The digital filter must be disabled.
2. Power-down Mode
  - The digital filter must be disabled.
  - PVD1IRS is set to 0, and PVD1 is selected to generate an interrupt; when the reset function is selected, Power-down mode cannot be entered.

The figure below is the running timing diagram of the voltage monitoring 1 interrupt. PVD1DETFLG needs to be cleared before the interrupt can occur again.

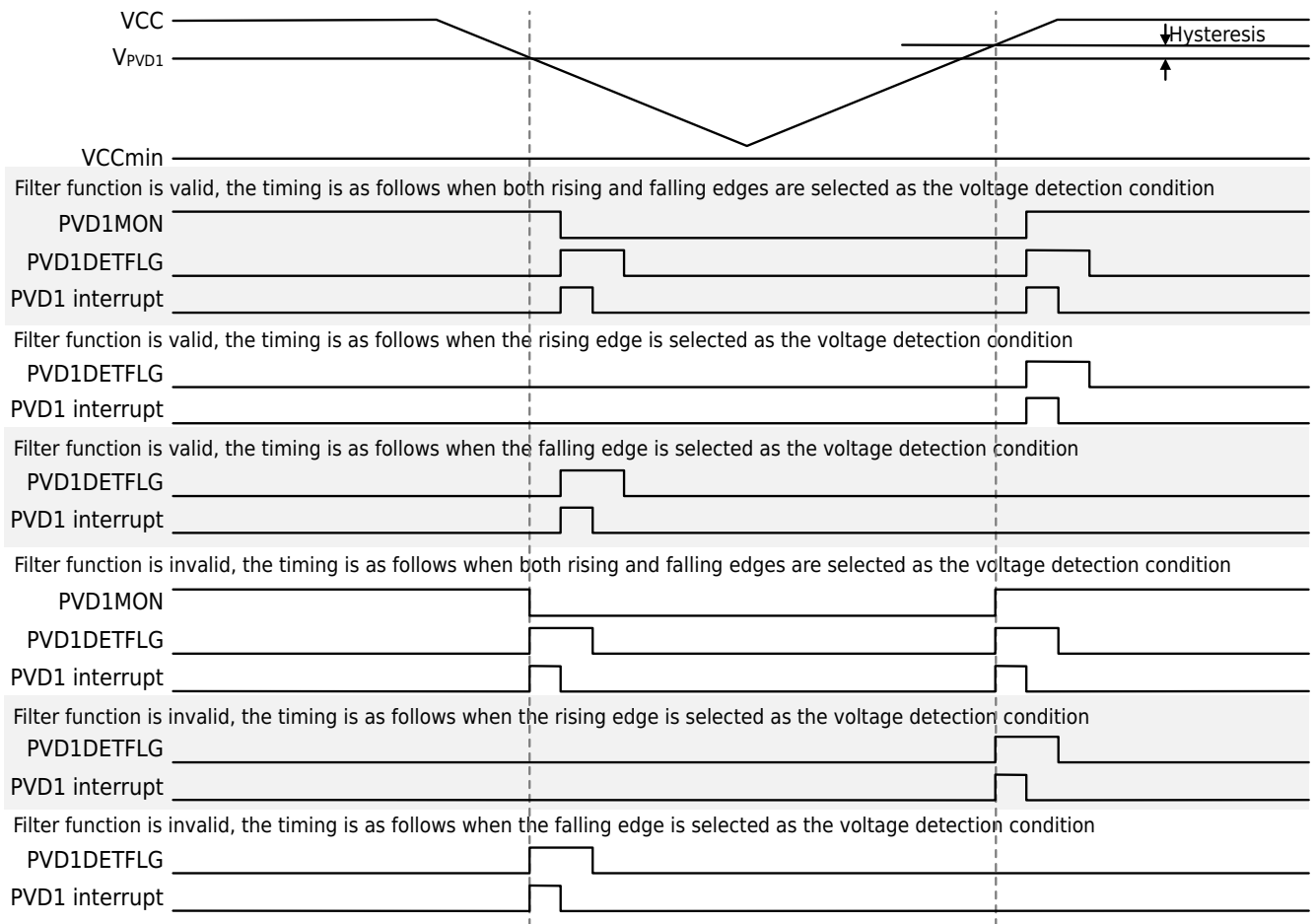


Figure 6-7 Power Monitor 1 Interrupt Timing Diagram

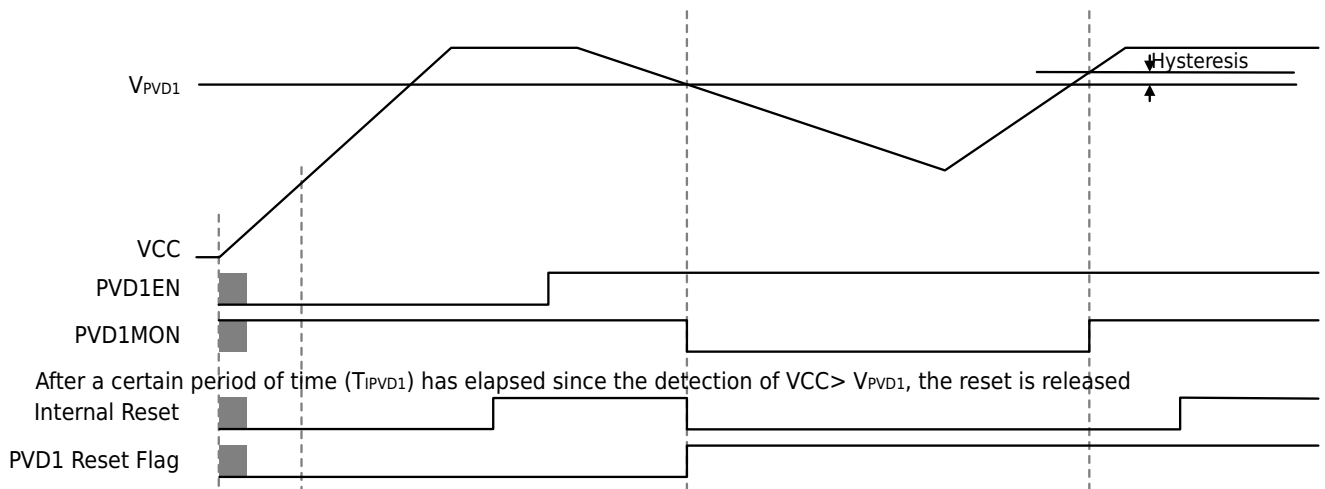


Figure 6-8 Power Monitor 1 Reset Timing Diagram

### 6.2.2.7 PVD2 Interrupt and Reset

When using the PVD2 circuit in Stop mode or Power-down mode, please observe the following precautions:

1. Stop Mode
  - The digital filter must be disabled.
2. Power-down Mode
  - The digital filter must be disabled.

- PVD2IRS is set to 0, and PVD2 is selected to generate an interrupt; when the reset function is selected, Power-down mode cannot be entered.

The figure below is the running timing diagram of the voltage monitoring 2 interrupt. PVD2DETFLG needs to be cleared before the interrupt can occur again.

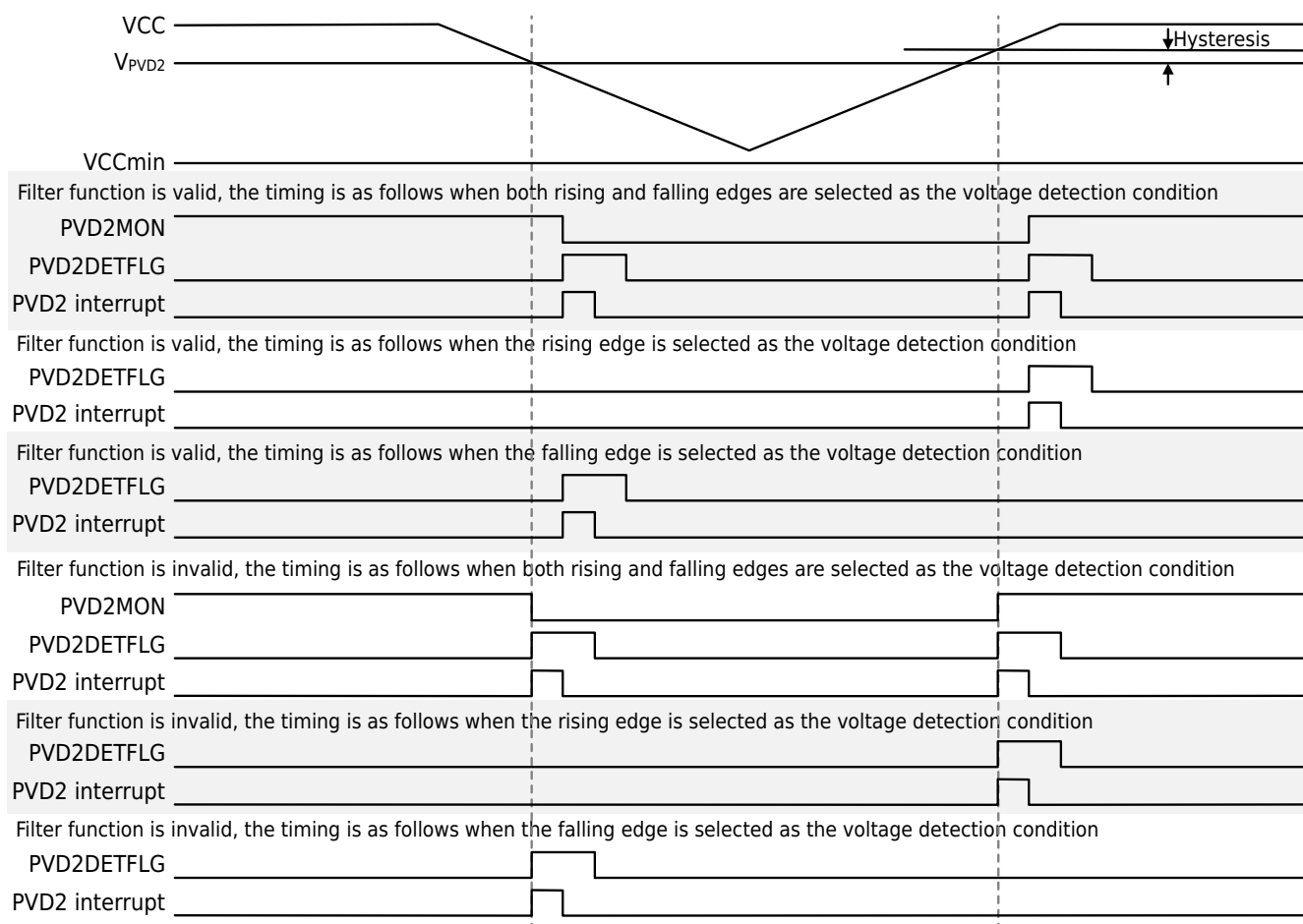


Figure 6-9 Power Monitor 2 Interrupt Timing Diagram

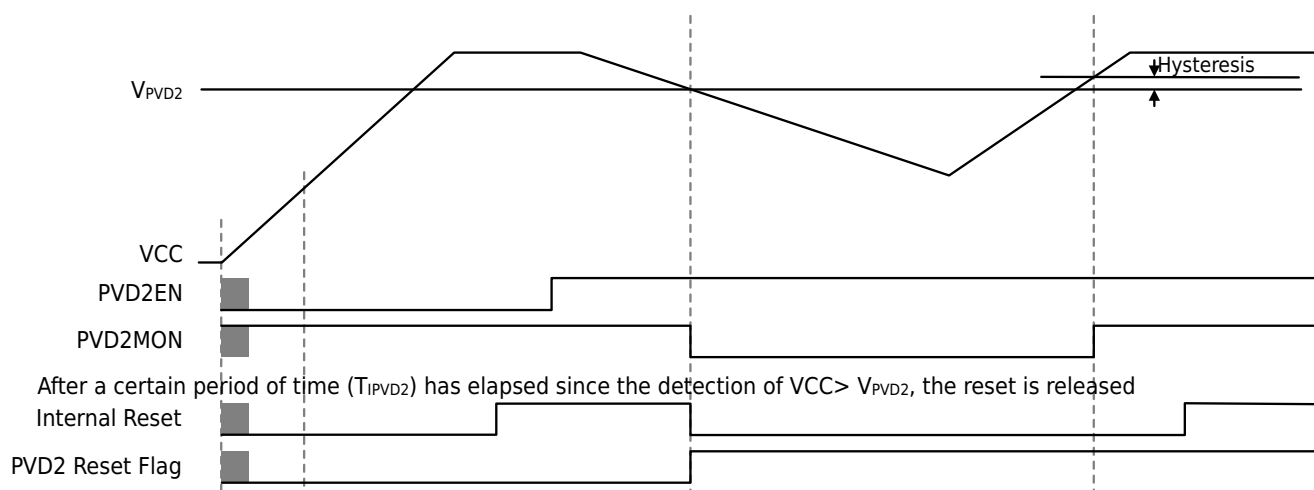


Figure 6-10 Power Monitor 2 Reset Timing Diagram

### 6.2.2.8 Internal Voltage Sampling and Detection Function

The internal voltage sampling and detection functions of the chip comprise three key features: reference voltage measurement, VBAT voltage detection, and VBAT voltage measurement. The reference voltage measurement path is a function for measuring the reference voltage using the ADC. The internal refer-

ence voltage is approximately 1.15V. VBAT voltage detection is a function that reads the register to obtain the VBAT voltage is higher or lower than the VBAT detection voltage. The VBAT detection voltage can be selected as 1.8V or 2.1V using the register. The VBAT voltage measurement function refers to the function of using ADC to measure the 1/2 voltage division of VBAT to obtain the VBAT voltage value.

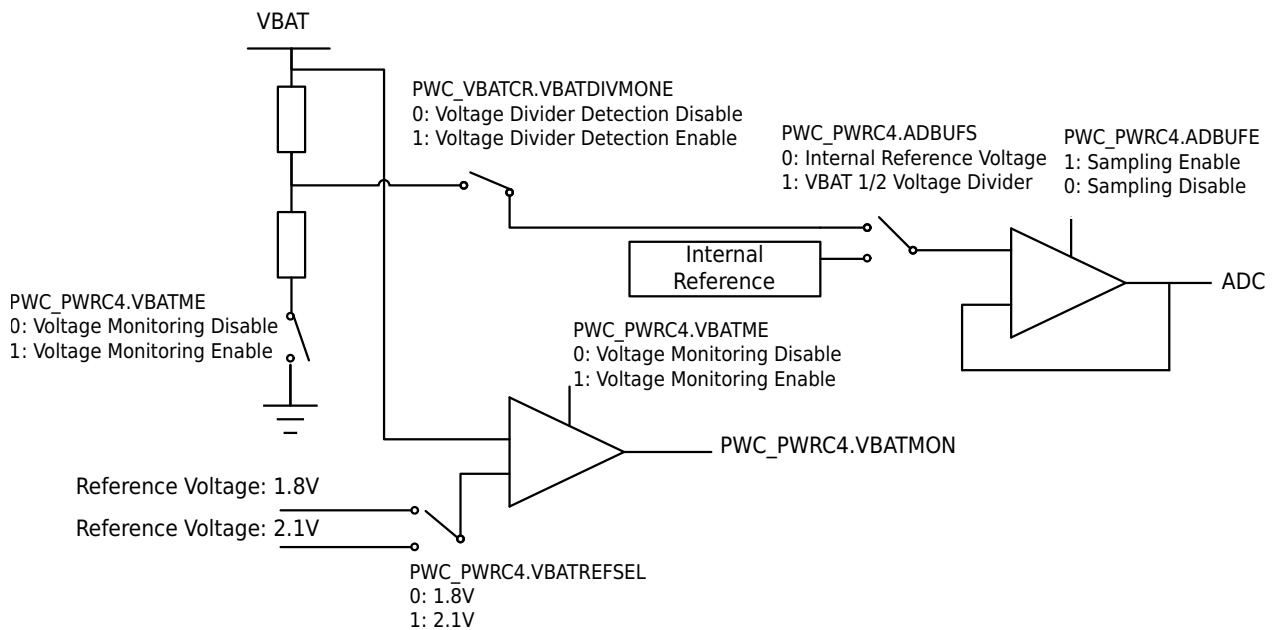


Figure 6-11 Internal Voltage Sampling Diagram

#### ● Reference Voltage Measurement Path

When using the reference voltage measurement path, you need to select the reference voltage measurement path according to the following steps.

1. `PWC_PWRC4.ADBUFS=0b0`, select the internal reference voltage;
2. `PWC_PWRC4.ADBUFE=0b1`, enable internal voltage measurement function;
3. `ADC_EXCHSELR.EXCHSEL=0b1`, enables ADC to select the internal voltage measurement channel;
4. Use ADC to measure internal reference voltage after waiting 50μs.

#### ● VBAT Voltage Detection

When using the VBAT voltage detection path, you need to select the VBAT voltage detection path according to the following steps.

1. `PWC_PWRC4.VBATREFSEL=0b1`, select `VBATREF=2.1V`;  
`PWC_PWRC4.VBATREFSEL=0b0`, select `VBATREF=1.8V`;
2. `PWC_PWRC4.VBATME=0b1`, enable VBAT voltage divider circuit;
3. Wait for 50μs;
4. Read the `PWC_PWRC4.VBATMON` bit. When it reads 0b0, `VBAT>VBATREF`. When read as 0b1, `VBAT<VBATREF`.

#### ● VBAT Voltage Measurement

When using the VBAT voltage measurement path, you need to select the VBAT voltage measurement path according to the following steps.

1. `PWC_VBATCR.VBATDIVMONE=0b1`, enable VBAT voltage divider measurement;
2. `PWC_PWRC4.VBATME=0b1`, enable VBAT voltage divider circuit;
3. `PWC_PWRC4.ADBUFS=0b1`, select 1/2 voltage division of VBAT;
4. `PWC_PWRC4.ADBUFE=0b1`, enable internal voltage measurement function;

5. ADC\_EXCHSELR.EXCHSEL=0b1, enables ADC to select the internal voltage measurement channel;
6. Wait for 50μs and then use ADC to measure the VBAT voltage;
7. PWC\_VBATCR.VBATDIVMONE=0b0, turn off VBAT voltage divider measurement.

### 6.2.3 Wake-up Timer

The chip has a built-in wake-up timer WKTM. The counter can select the RTC internal low-speed oscillator (RTCLRC) and external low-speed oscillator as the clock source. When the RTC uses an external low-speed oscillator as the clock action, the 64Hz internal clock signal can also be selected as the clock source. This counter is an add counter. After PWC\_WKTC2.WKTCE is set, the counter starts counting. When the count value is equal to the WKTCMP[11:0] setting value of PWC\_WKTC1 and PWC\_WKTC0, the counter count is cleared and then counts again, and a match consistency event is generated. This event can be used as a normal interrupt and can be used to wake up the Stop mode and Power-down mode.

The value of the WKTM register is uncertain after power-on reset. After power-on reset, must first set the VBAT area reset register, then set PWC\_WKTC0/PWC\_WKTC1/PWC\_WKTC2. Before starting WKTM, make sure the counting clock is on.

### 6.2.4 Action Mode and Low Power Mode

#### 6.2.4.1 Mode Overview

After the system reset or power-on reset, all the power fields of the chip are in the power supply state, and the chip enters the high-speed run mode. In run mode, the CPU provides the clock through HCLK and executes program code. The chip provides 2 run modes: high-speed run mode and ultra low-speed run mode. The chip can be configured in the following run modes.

Table 6-4 RUN Mode

| RUN Mode                 | Description                  |
|--------------------------|------------------------------|
| High-speed RUN Mode      | Main frequency $\leq$ 240MHz |
| Ultra Low-speed RUN Mode | Main frequency $\leq$ 8MHz   |

In order to save power consumption when the CPU does not need to run, the system provides three low-power modes: Sleep mode, Stop mode, and Power-down mode. The low power consumption modes that the chip can be configured in are shown in the following table. In Sleep mode, the Cortex-M4 core of the chip stops operating, and the peripherals keep running; in Stop mode, the peripherals and CPU of the chip stop operating; in Power-down mode, the power supply of VDD domain is turned off, and the VDD domain peripherals stop operating. The real-time clock and wake-up timer located in the battery backup domain can operate in low-power mode, and the Ret SRAM can retain data.

Table 6-5 Low Power Mode

| Mode       | Description                                            |
|------------|--------------------------------------------------------|
| Sleep Mode | CPU clock stopped, peripherals keep running            |
| Stop Mode  | The chip peripherals and the clocks of CPU are stopped |

| Mode            |       | Description                                                                                                                                                                                                                                                                                     |
|-----------------|-------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Power-down Mode | PDMD1 | VDD domain power down                                                                                                                                                                                                                                                                           |
|                 | PDMD2 | The voltage detection unit is disabled except the VDD domain power down                                                                                                                                                                                                                         |
|                 | PDMD3 | In addition to the VDD domain being powered down, the power-on reset circuit enters low-power mode, and the voltage detection unit (PVD) is disabled. Compared with Power-down mode 4, when waking up from power down, the chip is completely reset except for PWC_PDWKF0/PWC_PDWKF1/RMU_RSTF0. |
|                 | PDMD4 | In addition to the VDD domain power-off, the power-on reset circuit enters the low power consumption mode, and the voltage detection unit (PVD) is disabled                                                                                                                                     |

Users can choose operating mode and low power mode based on the application to find the best balance between low power consumption, short start-up time, wake-up sources, and system execution efficiency.

Power consumption in operating mode can be optimized by the following methods:

1. Set the most suitable run mode
2. Reduce system clock speed
3. Turn off the clock sources that are not in use
4. Set the PWC\_FCGn (n=0-3) register to turn off unused functions
5. Power off the power of the RAM

The operating conditions of low power mode and the status of each module in low power mode are shown in the table below.

**Table 6-6 Operating Conditions of Low Power Mode and Status of Each Module in Low Power Mode**

| Item                             | Sleep mode                                                             | Stop mode                                                                                            | Power-down Mode                                     |
|----------------------------------|------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------|-----------------------------------------------------|
| Entry                            | PWC_STPMCR.STOP=0b0, PWC_PWRC0.PWDN=0b0, WFI                           | PWC_STPMCR.STOP=0b1, PWC_PWRC0.PWDN=0b0, WFI                                                         | PWC_STPMCR.STOP=0b1, PWC_PWRC0.PWDN=0b1, WFI        |
| Release                          | To interrupt or reset arbitrarily                                      | Interrupt or reset that can be used in Stop mode                                                     | Wake-up event or reset available in Power-down mode |
| External high-speed oscillator   | Optional, can be enabled by software                                   | Stop                                                                                                 | Stop                                                |
| External low-speed oscillator    | Optional, can be enabled by software                                   | Optional, can be enabled by software                                                                 | Optional, can be enabled by software                |
| Internal high-speed oscillator   | Optional, can be enabled by software                                   | Stop                                                                                                 | Power down                                          |
| Internal medium speed oscillator | Optional, can be enabled by software                                   | Stop                                                                                                 | Power down                                          |
| Internal low-speed oscillator    | Optional, can be enabled by software                                   | Optional, can be enabled by software                                                                 | Optional, can be enabled by software                |
| WDT dedicated clock oscillator   | Optional, can be enabled by software                                   | Optional, can be enabled by software                                                                 | Power down                                          |
| PLLA                             | Optional, can be enabled by software                                   | Stop                                                                                                 | Power down                                          |
| PLLH                             | Optional, can be enabled by software                                   | Stop                                                                                                 | Power down                                          |
| CPU                              | Stop (hold)                                                            | Stop (hold)                                                                                          | Power down                                          |
| RAM (Except Ret SRAM)            | Optional, can be enabled by software<br>Can be set to work, Power-down | Stop (hold)<br>According to the setting before entering standby, it can maintain Power-down or Sleep | Power down                                          |



| Item                                                                                   | Sleep mode                                                                    | Stop mode                                      | Power-down Mode                                                                                                                                             |
|----------------------------------------------------------------------------------------|-------------------------------------------------------------------------------|------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Ret SRAM                                                                               | Optional, can be enabled by software<br>Can be set to work, Power-down, Sleep | Stop (hold)<br>Can be set to Power-down, Sleep | Power-down or Sleep can be set                                                                                                                              |
| Flash                                                                                  | Optional, can be enabled by software                                          | Stop (hold)                                    | Power down, content holds                                                                                                                                   |
| DMA                                                                                    | Optional, can be enabled by software                                          | Stop (hold)                                    | Power down                                                                                                                                                  |
| Voltage regulator                                                                      | Enable<br>Drive adjustable                                                    | Enable<br>Drive adjustable                     | Stop                                                                                                                                                        |
| Power-on reset circuit                                                                 | Work                                                                          | Work                                           | Work<br>The accuracy of the reset circuit in Power-down mode 1/2 can be guaranteed, and the power-on reset voltage in Power-down mode 3/4 is not guaranteed |
| Brown-out reset BOR                                                                    | Optional, can be enabled by software                                          | Optional, can be enabled by software           | Optional, can be enabled by software in Power-down mode 1<br>Stop in Power-down mode 2/3/4                                                                  |
| Voltage detection module PVD                                                           | Optional, can be enabled by software                                          | Optional, can be enabled by software           | Optional, can be enabled by software in Power-down mode 1<br>Stop in Power-down mode 2/3/4                                                                  |
| WDT                                                                                    | Optional, can be enabled by software                                          | Stop (hold)                                    | Power down                                                                                                                                                  |
| SWDT                                                                                   | Optional, can be enabled by software                                          | Optional, can be enabled by software           | Power down                                                                                                                                                  |
| RTC                                                                                    | Optional, can be enabled by software                                          | Optional, can be enabled by software           | Optional, can be enabled by software                                                                                                                        |
| USBFS                                                                                  | Optional, can be enabled by software                                          | Stop (hold)                                    | Power down                                                                                                                                                  |
| Timer0                                                                                 | Optional, can be enabled by software                                          | Optional, can be enabled by software           | Power down                                                                                                                                                  |
| Timer2                                                                                 | Optional, can be enabled by software                                          | Optional, can be enabled by software           | Power down                                                                                                                                                  |
| WKTm                                                                                   | Optional, can be enabled by software                                          | Optional, can be enabled by software           | Optional, can be enabled by software                                                                                                                        |
| Other peripheral modules                                                               | Optional, can be enabled by software                                          | Stop (hold)                                    | Power down                                                                                                                                                  |
| ADC                                                                                    | Optional, can be enabled by software                                          | Stop                                           | Power down                                                                                                                                                  |
| DAC                                                                                    | Optional, can be enabled by software                                          | Optional, can be enabled by software           | Power down                                                                                                                                                  |
| PGA                                                                                    | Optional, can be enabled by software                                          | Optional, can be enabled by software           | Power down                                                                                                                                                  |
| CMP                                                                                    | Optional, can be enabled by software                                          | Optional, can be enabled by software           | Power down                                                                                                                                                  |
| PA0-PA10, PB0-PB2, PB5-PB13, PC0-PC13, PD0-PD15, PE0-PE15, PH2-PH15, PI0-PI12, PI14-15 | Optional, can be enabled by software                                          | Hold                                           | Hold or high resistance                                                                                                                                     |

| Item                 | Sleep mode                                                                                    | Stop mode                                                                                                                                                                                                             | Power-down Mode                                                                                                                                                                                                                                                           |
|----------------------|-----------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| PC14-PC15            | Optional, can be enabled by software                                                          | When used as an external low-speed oscillator, keep the oscillator active; when set as GPIO or other peripheral functions, please set to keep the two pins at the same level                                          | When set to GPIO or other peripheral functions, the state of PC14 and PC15 can be set to hold or high impedance, please set to keep the two pins at the same level                                                                                                        |
| PH0-PH1              | Optional, can be enabled by software                                                          | When used as an external high-speed oscillator, oscillator stops concussion and the pin state keeps the state before entering STOP mode. When GPIO or other peripheral functions are set, keep the status before STOP | When used as an external high-speed oscillator, the oscillator stops oscillating and the pin status remains the same as before entering the Power-down mode; when set as GPIO or other peripheral functions, the status before entering the Power-down mode is maintained |
| NRST reset pin       | Outside the chip is pulled up to VCC through circuit                                          | Outside the chip is pulled up to VCC through resistance                                                                                                                                                               | Outside the chip is pulled up to VCC through resistance                                                                                                                                                                                                                   |
| PA11-PA12, PB14-PB15 | Optional, can be enabled by software                                                          | Hold<br>Since current will be generated when the level of this pin is pulled high, if the USB STOP mode wake-up function is not used, the pull-up is disabled when entering STOP mode                                 | Hold or high resistance<br>Since the level of this pin will generate redundant current when it is pulled high, the pull-up is disabled when entering Power-down mode                                                                                                      |
| PI13/MD              | Optional, can be enabled by software                                                          | Hold                                                                                                                                                                                                                  | Hold<br>The outside of the chip is connected to ground through a resistor                                                                                                                                                                                                 |
| PA13-PA15, PB3, PB4  | Optional, can be enabled by software<br>The built-in pull-up circuit works as a JTAG function | Hold<br>The built-in pull-up circuit works as a JTAG function                                                                                                                                                         | Hold<br>The built-in pull-up circuit works as a JTAG function                                                                                                                                                                                                             |

#### 6.2.4.2 RUN Mode

The chip has two operating modes: high-speed and ultra low-speed. According to the speed requirements of the system, set the best operating mode, so that the core voltage and drive capability are adapted to the system clock frequency, so as to achieve the purpose of reducing power consumption. As shown in the following table, according to the PWC\_PWRC2.DVS and DDAS bits, the chip can work in the corresponding run mode. When the chip selection action is in ultra low-speed mode, FLASH and RAM also need to be set to work at low core voltage, so register bits FRMC.LVM=0b1, PWC\_RAMOPM=0x9062 need to be set.

Table 6-7 RUN Mode Description

| RUN Mode                  | Frequency Range | Register Setting |                |                |
|---------------------------|-----------------|------------------|----------------|----------------|
|                           |                 | PWC_PWRC2.DVS    | PWC_PWRC2.DDAS | PWC_PWRC3.DDAS |
| High-speed RUN Mode       | ≤240MHz         | 0b11             | 0b1111         | 0b11111111     |
| Ultra Low-speed RUN Mode* | ≤8MHz           | 0b10             | 0b0000         | 0b00000000     |



#### Note

\*: Ultra low-speed run mode only supports  $T_A = -40^{\circ}\text{C} \sim 85^{\circ}\text{C}$ .

The switching between the high-speed run mode and the ultra low-speed operation mode needs to follow the following process.

- **Switch from high-speed run mode to ultra low-speed run mode**

1. Set the clock source used in ultra low-speed run mode to ensure that the clock source meets the frequency requirements in ultra low-speed run mode;
2. Turn off the clock source and modules that are not required in ultra low-speed run mode to verify that Flash is not programmed or erased;
3. Set FLASH's FRMC.LVM to 0b1 and RAM run mode register PWC\_RAMOPM to 0x9062;
4. Confirm FRMC.LVM=0b1, PWC\_RAMOPM=0x9062;
5. Set PWC\_PWRC2.DDAS to 0b0000, set PWC\_PWRC3.DDAS to 0x00, and PWC\_PWRC2.DVS to 0b10;
6. Wait for TSWMD1 (30μs);
7. The chip operates in ultra low-speed run mode.

- **Switch from ultra low-speed run mode to high-speed run mode**

1. Set PWC\_PWRC2.DDAS to 0b1111, PWC\_PWRC3.DDAS to 0xFF, and PWC\_PWRC2.DVS to 0b11 according to the system frequency requirements;
2. Wait for TSWMD2 (30μs);
3. Set FLASH's FRMC.LVM to 0b0 and RAM run mode register PWC\_RAMOPM to 0x8043;
4. Confirm FRMC.LVM=0b0, PWC\_RAMOPM=0x8043;
5. The chip operates in high-speed run mode.

#### 6.2.4.3 Sleep Mode

In Sleep mode, the CPU stops running and its internal registers remain in the state before entering Sleep mode. The operating status of peripherals and other system modules other than the watchdog and dedicated watchdog does not change.

When set to auto-start through ICG, if the ICG0.WDTSLOFF bit is 0b1, the watchdog stops counting in Sleep mode; if the WDTSLOFF bit is 0b0, the watchdog continues counting in Sleep mode. When the ICG is not set to start automatically, start the watchdog via software startup, if the WDT\_CR.SLOFF bit is 0b1, the watchdog stops counting in Sleep mode; if the WDT\_CR.SLOFF bit is 0b0, the watchdog does not stop counting in Sleep mode.

When set to auto-start through ICG, if the ICG0.SWDTSLPOFF bit is 0b1, the dedicated watchdog stops counting in Sleep mode; if the SWDTSLPOFF bit is 0b0, the dedicated watchdog continues counting in Sleep mode.

- **Enter Sleep Mode**

When PWC\_STPMCR.STOP= 0b0, WFI instructions can be executed to enter Sleep mode.

- **Exit Sleep Mode**

Any interrupt and reset can wake the chip from Sleep mode. When waking up by interrupt, the chip enters interrupt processing program. The chip enters the reset state when exiting Sleep mode through reset.

#### 6.2.4.4 Stop Mode

In Stop mode, the CPU, most peripherals, and timer all stop. The chip maintains CPU internal register and SRAM data, peripheral state and pin state. In the Stop mode, because most of the timer stops working, the voltage regulator also reduces the drive capability, so the chip power consumption will be significantly reduced.

The leakage current in STOP mode is different at different voltage temperatures, and the set drive capability must meet the leakage needs of the chip.

When set to auto-start through ICG, if the ICG0.SWDTSLPOFF bit is 0b1, the dedicated watchdog stops counting in Stop mode; if the SWDTSLPOFF bit is 0b0, the dedicated watchdog continues counting in Stop mode.

Before executing the WFI command into Stop mode, make sure that FLASH is not programmed or erased and that the oscillation stop detect function is disabled, otherwise the chip will enter Sleep mode instead of Stop mode.

Before executing the WFI instruction to enter Stop mode, must ensure that PWC\_PWRC0.PDMDS is configured to 0b00.

In Stop mode, the ADC and DAC also consume power unless disabled before entering Stop mode. To disable DAC, clear DACR.DAE, DACR.DA1E, and DACR.DA2E. To disable ADC, clear the ADC\_STR.STRT bit, set PWC\_FCG3.ADC3, PWC\_FCG3.ADC2, and PWC\_FCG3.ADC1, and then execute the WFI instruction to enter Stop mode.

When waking up from STOP mode, select the clock after waking up and whether to wait for the Flash to stabilize through PWC\_STPMCR.CKSMRC and FLNWT. CKSMRC is used to control the clock source after wakeup. When CKSMRC=0b1, the system clock source of wakeup is MRC. When CKSMRC=0b0, the wake-up system clock remains the same as the clock source before STOP. FLNWT is used to control whether to wait for the Flash to stabilize after waking up. When FLNWT=0b0, it is necessary to wait for the Flash to stabilize when waking up; when FLNWT=0b1, it is not necessary to wait for the Flash to stabilize when waking up. FLNWT must be set when the program is running on RAM, otherwise the chip's action after waking up from STOP is not guaranteed. When the program runs on RAM and enters STOP mode, select CKSMRC=0b1, FLNWT=0b1 to wake up the system in the shortest time.

Before executing the WFI instruction to enter the Stop mode, it is necessary to ensure that the DMA is in the stop state, otherwise the chip may have unguaranteed actions.

Before executing the WFI instruction to enter Stop mode, it is necessary to ensure that the frequency division ratio of HCLK to PCLK1/2/3/4 and EXCLK is controlled at 1:1 or 1:2 or 1:4. If the frequency division ratio of HCLK to any peripheral clock exceeds 1:4, it is necessary to configure PCLK1=PCLK3 and set PCLK1 and PCLK3 as the slowest clocks.

Before executing the WFI command to enter the Stop mode, the digital filter of EIRQ needs to be set to inactive, otherwise the interrupt cannot be used for STOP wake-up.

In the ultra low-speed mode, the chip needs to set PWC\_PWRC1.STPDAS=0b11 before entering the STOP mode; if entering STOP mode when PWC\_PWRC1.STPDAS= 0b00, the chip will consume more current in STOP mode.

When releasing the Stop mode through a non-maskable interrupt, it is necessary to set the corresponding bit of INTC\_NMIER to enable the interrupt; When releasing the stop mode through a maskable interrupt, it is necessary to set the corresponding bit of the INTC\_WKEN register to enable the wake-up permission of the interrupt.

- Enter Stop mode

**Step1.** Clear PWC\_LPMCSR.LPMF;

**Step2.** When PWC\_STPMCR.STOP=0b1 and PWC\_PWRC0.PWDN=0b0, execute WFI instruction to enter stop mode;

**Step3.** Read PWC\_LPMCSR.LPMF. If this bit is 1, the chip has entered STOP mode normally. If this bit is 0, return to step 2.

For the status of the chip's peripherals and clock sources in Stop mode, see the table "[Table 6-6 : Operating Conditions of Low Power Mode and Status of Each Module in Low Power Mode](#)".

- Release Stop mode

The Stop mode can be removed by reset and interrupt. The reset methods that can be used to release Stop mode include pin reset, power-on reset, brown-out reset (BOR), programmable voltage detect 1/2 reset and dedicated watchdog reset. Interrupt events that can be used to release stop mode: pin interrupt EIRQ0-15, voltage detection 1 interrupt, voltage detection 2 interrupt, dedicated watchdog underflow interrupt, real-time clock period interrupt, real-time clock alarm interrupt, wake-up timer interrupt, comparator interrupt, USART1 RX interrupt, Timer0 compare match interrupt, Timer2 compare match interrupt.

When the chip releases the Stop mode through interrupt, it first activates the clock source used before entering the Stop mode. After all clock sources are stabilized, the chip releases the Stop mode. When the WFI instruction is executed to enter the stop mode, an interrupt event that is not used to release the Stop mode is generated. The chip will respond to the interrupt first instead of entering the STOP mode. To prioritize entering STOP mode without executing the interrupt, interrupts that are not used to release the Stop mode must be disabled before executing the WFI instruction.

#### 6.2.4.5 Power-down Mode

In Power-down mode, the power supply of all modules in the VDD domain is cut off, and the power consumption can reach the lowest.

When setting SWDT to start automatically via ICG:

- If ICG0.SWDTSLPOFF is 0b1, the dedicated watchdog will be powered off like other modules in the VDD domain and will no longer count;
- If ICG0.SWDTSLPOFF is 0b0, the chip will enter Stop mode instead of Power-down mode, and the dedicated watchdog oscillator and dedicated watchdog will continue to run.

When SWDT starts by setting the software refresh register:

- If SWDT\_CR.SLPOFF is 0b1, the dedicated watchdog will be cut off from power like other modules in the VDD domain and will no longer count;
- If SWDT\_CR.SLPOFF is 0b0, the chip will enter Stop mode instead of Power-down mode, and the dedicated watchdog oscillator and dedicated watchdog will continue to run.

When the reset of voltage detect 1 and voltage detect 2 is enabled, the chip will enter Stop mode instead of Power-down mode.

Before executing the WFI command to enter the Power-down mode, it is necessary to ensure that the FLASH is not in the programming or erasing state, and the oscillation stop monitoring function is disabled, otherwise the chip will enter the Sleep mode instead of the Power-down mode.

The capacitors used on the VCAP\_1/VCAP\_2 pins of the chip are as follows:

- For chips with both VCAP\_1 and VCAP\_2 pins, each pin can use 0.047μF or 0.1μF capacitance (the total capacity is 0.094μF or 0.2μF).
- For the chip with only VCAP\_1 pin, it can use 0.1μF or 0.22μF capacitance

When the Power-down mode is wakened, it needs to charge VCAP\_1/VCAP\_2 during the process of kernel voltage establishment. On the one hand, the smaller total capacity of VCAP\_1/VCAP\_2 can shorten the charging time and bring rapid response to the application; on the other hand, the larger total capacity of VCAP\_1/VCAP\_2 will prolong the charging time, and provide stronger electromagnetic compatibility

(EMC). Users can choose smaller or larger capacitors according to the requirements of electromagnetic compatibility and system response speed. The total capacity of VCAP\_1/VCAP\_2 of the chip must match the assignment of PWC\_PWRC1.PDTS bit. When the total capacity of VCAP\_1/VCAP\_2 is 0.2μF or 0.22μF, it is necessary to ensure that the PWC\_PWRC1.PDTS bit is cleared before entering Power-down mode. When the total capacity of VCAP\_1/VCAP\_2 is 0.094μF or 0.1μF, ensure that PWC\_PWRC1.PDTS position bit is set before entering the Power-down mode.

By setting PWC\_PWRC0.PDMDS, the power consumption in Power-down mode can be further reduced. The sub-modes of Power-down mode are shown in the following table.

- In Power-down mode 1, the voltage monitoring circuit can be used, and the power-on reset detection circuit is in the active state. Since there is no need to wait for the stability of the VCC domain reference voltage circuit, voltage monitoring circuit, and power-on reset detection circuit when waking up, therefore, while achieving low power consumption, the wake-up time is the shortest.
- In Power-down mode 2, the reference voltage circuit in the VCC domain and the voltage monitoring circuit stop working, and the power-on reset detection circuit is in an active state. When waking up, it needs to wait for the stabilization time of the reference voltage circuit in the VCC domain and the voltage monitoring circuit.
- In Power-down mode 3, the VCC domain reference voltage circuit, voltage monitoring circuit, and power-on reset detection circuit all stop working. It is necessary to wait for these circuits to stabilize when waking up. Therefore, while achieving the lowest power consumption, the wake-up time is longer than Power-down mode 2 and Power-down mode 1.
- Power-down mode 4 has the same circuit that stops working in Power-down mode 3, so Power-down mode 4 has the same power consumption as Power-down mode 3.

For specific power consumption values and wake-up time, please refer to the "Electrical Characteristics" chapter of the "Datasheet".

Table 6-8 Power-down Mode Submodes

| Power-down Mode   | PDMDS | Power Consumption | Wake-up Time | Note                                                                                                                  |
|-------------------|-------|-------------------|--------------|-----------------------------------------------------------------------------------------------------------------------|
| Power-down mode 1 | 0b00  | $I_{PD1}$         | $T_{PD1}$    | VCC domain supply voltage detection unit is enable                                                                    |
| Power-down mode 2 | 0b01  | $I_{PD2}$         | $T_{PD2}$    | The POR and PDR detection circuits in the VCC domain are active, while the BOR, PVD1, and PVD2 functions are inactive |
| Power-down mode 3 | 0b10  | $I_{PD3}$         | $T_{PD3}$    | VCC domain POR, PDR, BOR, PVD1, PVD2 inactive                                                                         |
| Power-down mode 4 | 0b11  | $I_{PD4}$         | $T_{PD4}$    | VCC domain POR, PDR, BOR, PVD1, PVD2 inactive                                                                         |

Relationship between power consumption and wake-up time:  $I_{PD1} > I_{PD2} > I_{PD3} = I_{PD4}$ ,  $T_{PD1} < T_{PD2} < T_{PD3} < T_{PD4}$ .

- Enter Power-down mode
  - Step1.** Clear PWC\_LPMCSR.LPMF;
  - Step2.** When PWC\_STPMCR.STOP=0b1 and PWC\_PWRC0.PWDN=0b1, execute WFI instruction to enter Power-down mode;
  - Step3.** Read PWC\_LPMCSR.LPMF. If this bit is 0, return to step 2; if a power down reset is correctly generated, step 3 will not be executed.



- Release Power-down mode

Power-down mode can be released by Power-down mode wake-up event or reset. The resets that can be used to wake up from Power-down mode include pin reset, power-on reset and voltage detection 0 reset. Events that can wake the system from Power-down mode include: WKUPn\_0/1/2/3 (n=0/1/2/3) wake-up events, real-time clock alarm and timer events, voltage detection 1 wake-up event, voltage detection 2 wake-up event, and wake-up timer event.

After waking up from Power-down mode 1 and Power-down mode 2, reset the chip and execute the program again. The wake-up event can be queried through the Power-down wake-up flag bit, and the reset flag bit can be queried through RSTF0.PDRF.

In Power-down mode 3, POR, PDR, BOR, PVD1, and PVD2 circuits are all in an inactive state. After waking up from Power-down mode 3, all registers except PWC\_PDWKFO/PWC\_PDWKFI/RSTF0 are reset, and the chip work follows a similar power-on reset mode; the reset flag bit can be queried through RMU\_RSTF0.PDRF.

In Power-down mode 4, POR, PDR, BOR, PVD1, and PVD2 circuits are all in an inactive state, and the program is re-executed after the chip is reset; the reset flag bit can be queried through RMU\_RSTF0.PDRF.

The Power-down wake-up event is controlled by the Power-down wake-up enable register (PWC\_PDWKE0-PDWKE3) and the Power-down wake-up event edge selection register (PWC\_PDWKES). When a Power-down wake-up event occurs, the corresponding Power-down wake-up flag (PWC\_PDWKFO-PWC\_PDWKFI) is set. After Power-down wake-up, if the Power-down wake-up flag is not cleared, the chip cannot enter Power-down mode again. The edge of Power-down wake-up event can be selected by PWC\_PDWKES.

When waking up from Power-down mode, the VDD domain will re-supply power, the system executes Power-down wake-up reset, and medium speed oscillator inside the working clock. When waking up from Power-down mode, registers that are not reset see the following table.

| Power-down Mode       | Registers that Are not Reset                                                                                                                                                    |
|-----------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Power-down Mode 1/2/4 | PWC_PWRC0, PWC_PWRC1, PWC_PWRC3, PWC_PDWKE0, PWC_PDWKE1, PWC_PDWKE2, PWC_PDWKES, PWC_PDWKFO, PWC_PDWKFI, PWC_PVDCR0, PWC_PVDCR1, PWC_PVDPCR, PWC_PVDLCR, PWC_PVDICR, PWC_PVDDSR |
| Power-down mode 3     | PWC_PDWKFO, PWC_PDWKFI, RMU_RSTF0                                                                                                                                               |

- Pin status after releasing Power-down mode

In Power-down mode, the chip pins will remain in the state before entering Power-down mode or in high-impedance state according to the register setting value. If PWC\_PWRC0.IORTN=0b10 or 0b11, the pin state is high-impedance in Power-down mode, and the pin is initially high-impedance after the Power-down mode is released. If PWC\_PWRC0.IORTN=0b00, the state of the pin remains in the state before the Power-down mode in the Power-down mode, and the pin is initially in a high-impedance state after waking up. If PWC\_PWRC0.IORTN=0b01, the pins of the chip will remain in the state before entering the Power-down mode, and the state of the pins of the chip will not change even if the registers of the peripherals or pins are set after waking up. After PWC\_PWRC0.IORTN is cleared by software, the pin state is controlled by the peripheral or pin register settings.

- WKTM Power-down mode wake-up event

The chip has a built-in counter WKTM for Power-down wake-up. The counter can select an internal low-speed oscillator or an external low-speed oscillator. When the RTC uses an external low-speed oscillator as the clock action, WKTM can also select a 64Hz internal clock signal as the clock source.

This counter is an accumulative counter. After WKTC0.WKTCE is set, the counter starts counting. When the count value is equal to one period of the value set by WKTCMP[11:0], the counting stops and a wake-up event is generated, waking up the chip from Power-down mode. When using WKTM again, you need to reset WKTC0.WKTCE and set it again.

- PTWK Power-down mode wake-up event

The chip has four PTWK events for waking up from Power-down mode: PTWK0, PTWK1, PTWK2, and PTWK3. By setting PWC\_PDWKE0/PWC\_PDWKE1 through software, one of the four pins WKUPn\_0, WKUPn\_1, WKUPn\_2, and WKUPn\_3 can be enabled to trigger a PTWK event. Each PTWK event can select the rising or falling trigger edge of the trigger pin and has an independent flag bit.

The PTWK power-off wake-up event configuration process is as follows:

1. Configure the corresponding bit of PWC\_PDWKES to select the edge;
2. Configure PWC\_PDWKE0 and PWC\_PDWKE1 to enable the corresponding pins;
3. Clear the corresponding flag in PWC\_PDWKF0.

The structural diagram of configuring PTWK<sub>n</sub> is shown in the figure below.

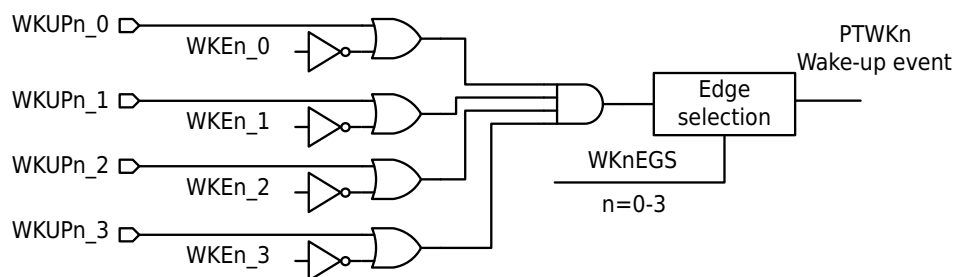


Figure 6-12 PTWK<sub>n</sub> Structure Diagram

- Battery backup domain operation in Power-down mode

After the chip enters Power-down mode, the battery backup domain Ret SRAM continues to be powered by RLDO, and the RTC/WKTM/backup register continues to be powered by VCC, allowing the RTC/WKTM/Ret SRAM to continue operating or retaining data. The LDO powers the VDDR domain after Power-down wake-up. Before entering Power-down mode, the VBAT voltage measurement function must be disabled.

- Methods of reducing power consumption

Power consumption in operating mode can be optimized by the following methods.

1. Set the most suitable run mode;
2. Reduce system clock speed;
3. Turn off the clock sources that are not in use;
4. Set the PWC\_FCG<sub>n</sub> (n=0/1/2/3) register to turn off unused functions;
5. Power off the power of the RAM.

## 6.2.5 Methods to Reduce Power Consumption

### 6.2.5.1 Reduce System Clock Speed

In run mode, the speed of system clock (HCLK), external bus clock (EXCLK) and peripheral clock PCLK0/PCLK1/PCLK2/PCLK3/PCLK4 can be reduced by programming the prescaler register. These pre-dividers can also be used to reduce the speed of peripherals before entering Sleep mode. For more information, see "Clock Module Unit (CMU)".



### 6.2.5.2 Unused Clock Sources Disabled

The system clock of the chip has the following clock sources:

- External high-speed oscillator (XTAL)
- External low-speed oscillator (XTAL32)
- PLLH clock (PLLH)
- Internal high-speed oscillator (HRC)
- Internal medium speed oscillator (MRC)
- Internal low-speed oscillator (LRC)

SWDT has an independent dedicated internal low-speed oscillator (SWDTLRC); RTC can select an external low-speed oscillator or RTC internal low-speed oscillator (RTCLRC) as the clock source.

Both HRC and PLL are equipped with independent power supply circuits. After HRC is turned off, the power supply of HRC can be turned off by setting the PWC\_PWRC1.VHRCSD bit to further reduce power consumption. After PLLA and PLLH are disabled, the power supply of PLLA and PLLH can be turned off by setting PWC\_PWRC1.VPLLSD=0b11.

For more information, see Clock Module Unit (CMU).

### 6.2.5.3 Functional Clock Stop

The peripheral module of the chip is provided with a clock stop function. By setting the register corresponding position bit, the module which does not need to be used can be stopped, and the clock of the corresponding module is also stopped to reduce the power consumption. In the module stop state, the register inside the module remains in the state it was in before stopping.

### 6.2.5.4 Unused RAM Disabled

Each RAM module in the chip is configured with a function clock stop bit and a Power-down control bit. By setting the module stop bit, it stops providing clocks to RAMs that are not need to be used, thereby reducing power consumption. By setting the Power-down control bit of the module, the corresponding RAM module can be powered down, thereby reducing power consumption. The following table shows the correspondence between RAM modules and Power-down control bits. The corresponding RAM can be powered off by setting the PWC\_RAMPC0.RAMPDCn bit, PWC\_VBATCR.VBTRSD bit.

Table 6-10 RAM Module and RAM Power-down Control Bit

| RAM Module | Description                                    | Power-down Control Bit |
|------------|------------------------------------------------|------------------------|
| SRAM1      | 0x20000000-0x2000FFFF<br>RAM for address space | PWC_RAMPC0.RAMPDC0     |
|            | 0x20010000-0x2001FFFF<br>RAM for address space | PWC_RAMPC0.RAMPDC1     |
| SRAM2      | 0x20020000-0x2002FFFF<br>RAM for address space | PWC_RAMPC0.RAMPDC2     |
|            | 0x20030000-0x2003FFFF<br>RAM for address space | PWC_RAMPC0.RAMPDC3     |
| SRAM3      | 0x20040000-0x2004FFFF<br>RAM for address space | PWC_RAMPC0.RAMPDC4     |
|            | 0x20050000-0x20057FFF<br>RAM for address space | PWC_RAMPC0.RAMPDC5     |

| RAM Module    | Description                                    | Power-down Control Bit |
|---------------|------------------------------------------------|------------------------|
| SRAM4         | 0x20058000-0x2005FFFF<br>RAM for address space | PWC_RAMPC0.RAMPDC6     |
| SRAMH         | 0x1FFE0000-0x1FFE7FFF<br>RAM for address space | PWC_RAMPC0.RAMPDC7     |
|               | 0x1FFE8000-0x1FEFFFFF<br>RAM for address space | PWC_RAMPC0.RAMPDC8     |
|               | 0x1FFF0000-0x1FFF7FFF<br>RAM for address space | PWC_RAMPC0.RAMPDC9     |
|               | 0x1FFF8000-0x1FFFFFFF<br>RAM for address space | PWC_RAMPC0.RAMPDC10    |
| Ret SRAM      | 0x200F0000-0x200FFFFF<br>RAM for address space | PWC_VBATCR.VBTRSD      |
| CAN_1 RAM     | RAM for CAN unit 1                             | PWC_PRAMLPC.PRAMDC0    |
| CAN_2 RAM     | RAM for CAN unit 2                             | PWC_PRAMLPC.PRAMDC1    |
| CACHERAM      | RAM for Cache                                  | PWC_PRAMLPC.PRAMDC2    |
| USBFS RAM     | RAM for USBFS                                  | PWC_PRAMLPC.PRAMDC3    |
| USBHS RAM     | RAM for USBHS                                  | PWC_PRAMLPC.PRAMDC4    |
| ETHMAC TX RAM | RAM for ETHMAC TX                              | PWC_PRAMLPC.PRAMDC5    |
| ETHMAC RX RAM | ETHMAC RX for RAM                              | PWC_PRAMLPC.PRAMDC6    |
| SDIOC1 RAM    | RAM for SDIOC1                                 | PWC_PRAMLPC.PRAMDC7    |
| SDIOC2 RAM    | RAM for SDIOC2                                 | PWC_PRAMLPC.PRAMDC8    |
| NFCRAM        | RAM for NFC                                    | PWC_PRAMLPC.PRAMDC9    |
| MCAN RAM      | RAM for MCAN                                   | PWC_PRAMLPC.PRAMDC10   |
| PKE RAM       | PKE for RAM                                    | PWC_PRAMLPC.PRAMDC11   |

## 6.3 Register Description

Register protection function is used to inactivate the write operation of the register to protect the register from accidental overwriting.

Table 6-11 Register Protection List

| Protected Register Bit | Protected Register                                                                                                                                                                                                                                                                                                                                                        |
|------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| PWC_FPRC.FPRCB0        | CMU_XTALCFGR, CMU_XTALSTBCR, CMU_XTALCR, CMU_XTALSTDCR, CMU_XTALSTDSR, CMU_HRCTRM, CMU_HRCCR, CMU_MRCTRM, CMU_MRCCR, CMU_PLLHCFGR, CMU_PLLHCR, CMU_PLLACFGR, CMU_PLLACR, CMU_OSCSTBSR, CMU_CKSWR, CMU_SCFGR, CMU_USBCKCFGR, CMU_CANCKCFGR, CMU_TPIUCKCFGR, CMU_MCO1CFGR, CMU_MCO2CFGR, CMU_XTAL32CR, CMU_XTALC32CFGR, CMU_XTAL32NFR, CMU_LRCCR, CMU_LRCTRM, CMU_RTCLRCTRM |
| PWC_FPRC.FPRCB1        | PWC_PWRC0, PWC_PWRC1, PWC_PWRC2, PWC_PWRC3, PWC_PWRC4, PWC_PDWKE0, PWC_PDWKE1, PWC_PDWKE2, PWC_PDWKE3, PWC_PDWKF0, PWC_PDWKF1, CMU_PERICKSEL, CMU_I2SCKSEL, PWC_STPMCR, PWC_RAMPC0, PWC_RAMOPM, PWC_PRAMLPC, RMU_RSTF0, RMU_FRST0, RMU_FRSTR1, RMU_FRST2, RMU_FRST3, RMU_PRSTCR0, PWC_VBATRSTR, PWC_VBATCR, PWC_WKTC0, PWC_WKTC1, PWC_WKTC2                               |
| PWC_FPRC.FPRCB3        | PWC_PVDCR0, PWC_PVDCR1, PWC_PVDFCR, PWC_PVDLCR, PWC_PVDICR, PWC_PVDDSR                                                                                                                                                                                                                                                                                                    |
| PWC_FCG0PC.PRT0        | PWC_FCG0                                                                                                                                                                                                                                                                                                                                                                  |

### 6.3.1 Register List

Base address: 0x40048000

Table 6-12 Power Control (PWC) Register Offset

| Offset | Register                   | Description                          | Width |
|--------|----------------------------|--------------------------------------|-------|
| 0x0000 | <a href="#">PWC_FCG0</a>   | Function Clock Control Register 0    | 32    |
| 0x0004 | <a href="#">PWC_FCG1</a>   | Function Clock Control Register 1    | 32    |
| 0x0008 | <a href="#">PWC_FCG2</a>   | Function Clock Control Register 2    | 32    |
| 0x000C | <a href="#">PWC_FCG3</a>   | Function Clock Control Register 3    | 32    |
| 0x0010 | <a href="#">PWC_FCG0PC</a> | PWC_FCG0 Protection Control Register | 32    |
| 0x4C00 | <a href="#">PWC_PWRC0</a>  | Power Mode Control Register 0        | 8     |
| 0x4C04 | <a href="#">PWC_PWRC1</a>  | Power Mode Control Register 1        | 8     |
| 0x4C08 | <a href="#">PWC_PWRC2</a>  | Power Mode Control Register 2        | 8     |
| 0x4C0C | <a href="#">PWC_PWRC3</a>  | Power Mode Control Register 3        | 8     |
| 0x4C10 | <a href="#">PWC_PWRC4</a>  | Power Mode Control Register 4        | 8     |
| 0x4C14 | <a href="#">PWC_PVDCR0</a> | PVD Control Register 0               | 8     |

| Offset       | Register        | Description                                      | Width |
|--------------|-----------------|--------------------------------------------------|-------|
| 0x4C18       | PWC_PVDCR1      | PVD Control Register 1                           | 8     |
| 0x4C1C       | PWC_PVDFCR      | PVD Filter Control Register                      | 8     |
| 0x4C20       | PWC_PVDLCR      | PVD Level Control Register                       | 8     |
| 0x4C28       | PWC_PDWKE0      | Power-down Wake-up Enable Register 0             | 8     |
| 0x4C2C       | PWC_PDWKE1      | Power-down Wake-up Enable Register 1             | 8     |
| 0x4C30       | PWC_PDWKE2      | Power-down Wake-up Enable Register 2             | 8     |
| 0x4C34       | PWC_PDWKES      | Power-down Wake-up Event Edge Select Register    | 8     |
| 0x4C38       | PWC_PDWKFO      | Power-down Wake-up Flag Register 0               | 8     |
| 0x4C3C       | PWC_PDWKFI      | Power-down Wake-up Flag Register 1               | 8     |
| 0x4CE0       | PWC_RAMPC0      | RAM Power Control Register 0                     | 32    |
| 0x4CE4       | PWC_RAMOPM      | RAM Operating Condition Register                 | 16    |
| 0x4CE8       | PWC_PRAMLPC     | Peripheral RAM Low Power Control Register        | 32    |
| 0x4CEC       | PWC_LPMCSR      | Low Power Mode Status Register                   | 16    |
| 0x4CEE       | PWC_LPMCSCR     | Low Power Mode Status Clear Register             | 16    |
| 0x4CF0       | PWC_PVDICR      | PVD Interrupt Control Register                   | 8     |
| 0x4CF4       | PWC_PVDDSR      | PVD Detection Status Register                    | 8     |
| 0xC00C       | PWC_STPMCRR     | STOP Mode Control Register                       | 16    |
| 0xC3FE       | PWC_FPRC        | Function Protection Control Register             | 16    |
| 0x4430       | PWC_VBATRSTR    | Backup Domain Reset Register                     | 8     |
| 0x4440       | PWC_VBATCR      | Backup Domain Control Register                   | 8     |
| 0x4448       | PWC_VBATPORF    | Backup Domain Power-on Reset Flag Register       | 8     |
| 0x444C       | PWC_VBATPORFCLR | Backup Domain Power-on Reset Flag Clear Register | 8     |
| 0x4450       | PWC_WKTC0       | Wake-up Timer Control Register 0                 | 8     |
| 0x4454       | PWC_WKTC1       | Wake-up Timer Control Register 1                 | 8     |
| 0x4458       | PWC_WKTC2       | Wake-up Timer Control Register 2                 | 8     |
| 0x4600+0x4*n | PWC_BKRn        | VBAT Backup Register n (n=0-127)                 | 8     |

## 6.3.2 Power Mode Control Register 0 (PWC\_PWRC0)

| Offset | Bit Position |          |       |   |          |   |       |   |
|--------|--------------|----------|-------|---|----------|---|-------|---|
| 0x4C00 | 7            | 6        | 5     | 4 | 3        | 2 | 1     | 0 |
| Reset  | 0x00         |          |       |   |          |   |       |   |
| Name   | PWDN         | Reserved | IORTN |   | Reserved |   | PDMDS |   |
| Access | RW           |          | RW    |   |          |   | RW    |   |

| Bit | Name            | Function                              | Description                                                                                                                                                                                                                                                                                                                                                                           | Access |
|-----|-----------------|---------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 7   | PWDN            | Power-down mode control bit           | <ul style="list-style-type: none"> <li>0b0: Power-down mode disable</li> <li>0b1: Power-down mode enable</li> </ul>                                                                                                                                                                                                                                                                   | RW     |
| 6   | <i>Reserved</i> | <i>Reserved</i>                       | -                                                                                                                                                                                                                                                                                                                                                                                     | -      |
| 5:4 | IORTN           | IO remains control in Power-down mode | <ul style="list-style-type: none"> <li>0b00: IO hold status in Power-down mode, and the hardware releases IO hold state after Power-down wake-up</li> <li>0b01: IO hold status in Power-down mode. After Power-down wake-up, set IORTN[1:0] to 0b00 to release the IO hold status</li> <li>0b10/0b11: IO is high impedance in Power-down mode and after Power-down wake-up</li> </ul> | RW     |
| 3:2 | <i>Reserved</i> | <i>Reserved</i>                       | -                                                                                                                                                                                                                                                                                                                                                                                     | -      |
| 1:0 | PDMDS           | Power-down mode selection control     | <ul style="list-style-type: none"> <li>0b00: Power-down mode 1</li> <li>0b01: Power-down mode 2</li> <li>0b10: Power-down mode 3</li> <li>0b11: Power-down mode 4</li> </ul>                                                                                                                                                                                                          | RW     |

## 6.3.3 Power Mode Control Register 1 (PWC\_PWRC1)

| Offset | Bit Position |   |          |   |      |        |         |   |
|--------|--------------|---|----------|---|------|--------|---------|---|
| 0x4C04 | 7            | 6 | 5        | 4 | 3    | 2      | 1       | 0 |
| Reset  | 0x00         |   |          |   |      |        |         |   |
| Name   | STPDAS       |   | Reserved |   | PDTS | VHRCSD | VPL LSD |   |
| Access | RW           |   |          |   | RW   | RW     | RW      |   |

| Bit | Name     | Function                            | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                     | Access |
|-----|----------|-------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 7:6 | STPDAS   | STOP mode LDO driver selection      | <ul style="list-style-type: none"> <li>0b00: The driving capability set when the high-speed mode enters the STOP mode</li> <li>0b11: The drive capability set when ultra-low speed mode enters STOP mode</li> <li>Others: Setting prohibited</li> </ul> <div>  <b>Note</b><br/>           When STPDAS=0b11 is set, only T<sub>A</sub>=-40°C~105°C is supported.         </div> | RW     |
| 5:4 | Reserved | Reserved                            | -                                                                                                                                                                                                                                                                                                                                                                                                                                                               | -      |
| 3   | PDTS     | Power-down wake-up time control bit | <ul style="list-style-type: none"> <li>0b0: When the total capacitance of VCAP_1/VCAP_2 is two 0.1μF or one 0.22μF, the PDTS bit must be cleared before entering Power-down mode</li> <li>0b1: When the total capacitance of VCAP_1/VCAP_2 is two 0.047μF or one 0.1μF, the PDTS bit must be set before entering Power-down mode</li> </ul>                                                                                                                     | RW     |
| 2   | VHRCSD   | HRC power disable                   | <ul style="list-style-type: none"> <li>0b0: HRC power enable</li> <li>0b1: HRC power disable</li> </ul> <div>  <b>Note</b><br/>           When HRC is not in use, set VHRCSD and turn off the power supply for HRC to further reduce power consumption. After VHRCSD is cleared, you need to wait 25μs before starting the HRC module.         </div>                        | RW     |

| Bit | Name  | Function          | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | Access |
|-----|-------|-------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 1:0 | VPLLS | PLL power disable | <ul style="list-style-type: none"> <li>● 0b00: PLL power enable</li> <li>● 0b11: PLL power disable</li> <li>● Others: Setting prohibited</li> </ul> <div>  <b>Note</b><br/>           After PLLA and PLLH is turned off and waiting for 50μs, set VPLLS=0b11 and then turn off the PLL power supply to further reduce power consumption. After setting VPLLS=0b00, you need to wait for 25μs before starting the PLLA or PLLH module.         </div> | RW     |

## 6.3.4 Power Mode Control Register 2 (PWC\_PWRC2)

| Offset | Bit Position |   |     |   |            |   |   |   |
|--------|--------------|---|-----|---|------------|---|---|---|
| 0x4C08 | 7            | 6 | 5   | 4 | 3          | 2 | 1 | 0 |
| Reset  | 0xFF         |   |     |   |            |   |   |   |
| Name   | Reserved     |   | DVS |   | DDAS[11:8] |   |   |   |
| Access |              |   | RW  |   | RW         |   |   |   |

| Bit | Name       | Function                         | Description                                                                                                                                                                                                                                        | Access |
|-----|------------|----------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 7:6 | Reserved   | Reserved                         | All bits read as 1, Write as 1                                                                                                                                                                                                                     | -      |
| 5:4 | DVS        | Voltage selection in action mode | <ul style="list-style-type: none"> <li>0b00: Select ultra low-speed action voltage</li> <li>0b11: Select high-speed run mode voltage</li> <li>Others: Setting prohibited</li> </ul>                                                                | RW     |
| 3:0 | DDAS[11:8] | Power drive selection            | Combined with DDAS[7:0] in PWC_PWRC3 to adjust the drive capability. DDAS[11:0] value: <ul style="list-style-type: none"> <li>0x000: Ultra low-speed drive mode</li> <li>0xFFF: High-speed run mode</li> <li>Others: Setting prohibited</li> </ul> | RW     |

## 6.3.5 Power Mode Control Register 3 (PWC\_PWRC3)

| Offset | Bit Position |   |   |   |   |   |   |   |
|--------|--------------|---|---|---|---|---|---|---|
| 0x4C0C | 7            | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0xFF         |   |   |   |   |   |   |   |
| Name   | DDAS[7:0]    |   |   |   |   |   |   |   |
| Access | RW           |   |   |   |   |   |   |   |

| Bit | Name      | Function              | Description                                                                                                                                                                                                                                         | Access |
|-----|-----------|-----------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 7:0 | DDAS[7:0] | Power drive selection | Combined with DDAS[11:8] in PWC_PWRC2 to adjust the drive capability. DDAS[11:0] value: <ul style="list-style-type: none"> <li>0x000: Ultra low-speed drive mode</li> <li>0xFFF: High-speed run mode</li> <li>Others: Setting prohibited</li> </ul> | RW     |



## 6.3.6 Power Mode Control Register 4 (PWC\_PWRC4)

| Offset | Bit Position |        |          |         |          |   |        |            |
|--------|--------------|--------|----------|---------|----------|---|--------|------------|
| 0x4C10 | 7            | 6      | 5        | 4       | 3        | 2 | 1      | 0          |
| Reset  | 0x00         |        |          |         |          |   |        |            |
| Name   | ADBUFE       | ADBUFS | Reserved | VBATMON | Reserved |   | VBATME | VBATREFSEL |
| Access | RW           | RW     |          | RO      |          |   | RW     | RW         |

| Bit | Name            | Function                                             | Description                                                                                                                                                                                                                                                                                                     | Access |
|-----|-----------------|------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 7   | ADBUFE          | Internal voltage sampling enable                     | <ul style="list-style-type: none"> <li>0b0: Disable</li> <li>0b1: Enable</li> </ul> <div>  <b>Note</b><br/>           When using ADC to sample the internal voltage of the chip, this bit needs to be set to 1.         </div> | RW     |
| 6   | ADBUFS          | Sample internal voltage selection bit                | <ul style="list-style-type: none"> <li>0b0: Internal reference voltage</li> <li>0b1: 1/2 voltage division of VBAT</li> </ul>                                                                                                                                                                                    | RW     |
| 5   | <i>Reserved</i> | <i>Reserved</i>                                      | -                                                                                                                                                                                                                                                                                                               | -      |
| 4   | VBATMON         | VBAT voltage detection status                        | <ul style="list-style-type: none"> <li>0b0: VBAT &gt; VBATREF</li> <li>0b1: VBAT &lt; VBATREF</li> </ul>                                                                                                                                                                                                        | RO     |
| 3:2 | <i>Reserved</i> | <i>Reserved</i>                                      | <i>Writing 1 to this bit is prohibited</i>                                                                                                                                                                                                                                                                      | -      |
| 1   | VBATME          | VBAT detection enable                                | <ul style="list-style-type: none"> <li>0b0: Detection disable</li> <li>0b1: Detection enable</li> </ul>                                                                                                                                                                                                         | RW     |
| 0   | VBATREFSEL      | VBAT detection reference voltage (VBATREF) selection | <ul style="list-style-type: none"> <li>0b0: 1.8V</li> <li>0b1: 2.1V</li> </ul>                                                                                                                                                                                                                                  | RW     |

## 6.3.7 Power-down Wake-up Enable Register 0 (PWC\_PDWKE0)

| Offset | Bit Position |       |       |       |       |       |       |       |
|--------|--------------|-------|-------|-------|-------|-------|-------|-------|
| 0x4C28 | 7            | 6     | 5     | 4     | 3     | 2     | 1     | 0     |
| Reset  | 0x00         |       |       |       |       |       |       |       |
| Name   | WKE13        | WKE12 | WKE11 | WKE10 | WKE03 | WKE02 | WKE01 | WKE00 |
| Access | RW           | RW    | RW    | RW    | RW    | RW    | RW    | RW    |

| Bit | Name  | Function                          | Description                                                                             | Access |
|-----|-------|-----------------------------------|-----------------------------------------------------------------------------------------|--------|
| 7   | WKE13 | WKUP1_3<br>wakeup event<br>enable | <ul style="list-style-type: none"> <li>● 0b0: Disable</li> <li>● 0b1: Enable</li> </ul> | RW     |
| 6   | WKE12 | WKUP1_2<br>wakeup event<br>enable | <ul style="list-style-type: none"> <li>● 0b0: Disable</li> <li>● 0b1: Enable</li> </ul> | RW     |
| 5   | WKE11 | WKUP1_1<br>wakeup event<br>enable | <ul style="list-style-type: none"> <li>● 0b0: Disable</li> <li>● 0b1: Enable</li> </ul> | RW     |
| 4   | WKE10 | WKUP1_0<br>wakeup event<br>enable | <ul style="list-style-type: none"> <li>● 0b0: Disable</li> <li>● 0b1: Enable</li> </ul> | RW     |
| 3   | WKE03 | WKUP0_3<br>wakeup event<br>enable | <ul style="list-style-type: none"> <li>● 0b0: Disable</li> <li>● 0b1: Enable</li> </ul> | RW     |
| 2   | WKE02 | WKUP0_2<br>wakeup event<br>enable | <ul style="list-style-type: none"> <li>● 0b0: Disable</li> <li>● 0b1: Enable</li> </ul> | RW     |
| 1   | WKE01 | WKUP0_1<br>wakeup event<br>enable | <ul style="list-style-type: none"> <li>● 0b0: Disable</li> <li>● 0b1: Enable</li> </ul> | RW     |
| 0   | WKE00 | WKUP0_0<br>wakeup event<br>enable | <ul style="list-style-type: none"> <li>● 0b0: Disable</li> <li>● 0b1: Enable</li> </ul> | RW     |

## 6.3.8 Power-down Wake-up Enable Register 1 (PWC\_PDWKE1)

| Offset | Bit Position |       |       |       |       |       |       |       |
|--------|--------------|-------|-------|-------|-------|-------|-------|-------|
| 0x4C2C | 7            | 6     | 5     | 4     | 3     | 2     | 1     | 0     |
| Reset  | 0x00         |       |       |       |       |       |       |       |
| Name   | WKE33        | WKE32 | WKE31 | WKE30 | WKE23 | WKE22 | WKE21 | WKE20 |
| Access | RW           | RW    | RW    | RW    | RW    | RW    | RW    | RW    |

| Bit | Name  | Function                          | Description                                                                             | Access |
|-----|-------|-----------------------------------|-----------------------------------------------------------------------------------------|--------|
| 7   | WKE33 | WKUP3_3<br>wakeup event<br>enable | <ul style="list-style-type: none"> <li>● 0b0: Disable</li> <li>● 0b1: Enable</li> </ul> | RW     |
| 6   | WKE32 | WKUP3_2<br>wakeup event<br>enable | <ul style="list-style-type: none"> <li>● 0b0: Disable</li> <li>● 0b1: Enable</li> </ul> | RW     |
| 5   | WKE31 | WKUP3_1<br>wakeup event<br>enable | <ul style="list-style-type: none"> <li>● 0b0: Disable</li> <li>● 0b1: Enable</li> </ul> | RW     |
| 4   | WKE30 | WKUP3_0<br>wakeup event<br>enable | <ul style="list-style-type: none"> <li>● 0b0: Disable</li> <li>● 0b1: Enable</li> </ul> | RW     |
| 3   | WKE23 | WKUP2_3<br>wakeup event<br>enable | <ul style="list-style-type: none"> <li>● 0b0: Disable</li> <li>● 0b1: Enable</li> </ul> | RW     |
| 2   | WKE22 | WKUP2_2<br>wakeup event<br>enable | <ul style="list-style-type: none"> <li>● 0b0: Disable</li> <li>● 0b1: Enable</li> </ul> | RW     |
| 1   | WKE21 | WKUP2_1<br>wakeup event<br>enable | <ul style="list-style-type: none"> <li>● 0b0: Disable</li> <li>● 0b1: Enable</li> </ul> | RW     |
| 0   | WKE20 | WKUP2_0<br>wakeup event<br>enable | <ul style="list-style-type: none"> <li>● 0b0: Disable</li> <li>● 0b1: Enable</li> </ul> | RW     |

## 6.3.9 Power-down Wake-up Enable Register 2 (PWC\_PDWKE2)

| Offset | Bit Position |          |           |           |          |   |        |        |
|--------|--------------|----------|-----------|-----------|----------|---|--------|--------|
| 0x4C30 | 7            | 6        | 5         | 4         | 3        | 2 | 1      | 0      |
| Reset  | 0x00         |          |           |           |          |   |        |        |
| Name   | WKTMWKE      | Reserved | RTCALMWKE | RTCPRDWKE | Reserved |   | VD2WKE | VD1WKE |
| Access | RW           |          | RW        | RW        |          |   | RW     | RW     |

| Bit | Name            | Function                             | Description                                                                         | Access |
|-----|-----------------|--------------------------------------|-------------------------------------------------------------------------------------|--------|
| 7   | WKTMWKE         | WKTW wakeup event enable             | <ul style="list-style-type: none"> <li>0b0: Disable</li> <li>0b1: Enable</li> </ul> | RW     |
| 6   | <i>Reserved</i> | <i>Reserved</i>                      | -                                                                                   | -      |
| 5   | RTCALMWKE       | RTC alarm clock wake-up event enable | <ul style="list-style-type: none"> <li>0b0: Disable</li> <li>0b1: Enable</li> </ul> | RW     |
| 4   | RTCPRDWKE       | RTC period wakeup event enable       | <ul style="list-style-type: none"> <li>0b0: Disable</li> <li>0b1: Enable</li> </ul> | RW     |
| 3:2 | <i>Reserved</i> | <i>Reserved</i>                      | -                                                                                   | -      |
| 1   | VD2WKE          | PVD2 wakeup event enable             | <ul style="list-style-type: none"> <li>0b0: Disable</li> <li>0b1: Enable</li> </ul> | RW     |
| 0   | VD1WKE          | PVD1 wakeup event enable             | <ul style="list-style-type: none"> <li>0b0: Disable</li> <li>0b1: Enable</li> </ul> | RW     |

## 6.3.10 Power-down Wake-up Event Edge Select Register (PWC\_PDWKES)

| Offset | Bit Position |   |        |        |        |        |        |        |
|--------|--------------|---|--------|--------|--------|--------|--------|--------|
| 0x4C34 | 7            | 6 | 5      | 4      | 3      | 2      | 1      | 0      |
| Reset  | 0x00         |   |        |        |        |        |        |        |
| Name   | Reserved     |   | VD2EGS | VD1EGS | WK3EGS | WK2EGS | WK1EGS | WK0EGS |
| Access |              |   | RW     | RW     | RW     | RW     | RW     | RW     |

| Bit | Name            | Function             | Description                                                                                    | Access |
|-----|-----------------|----------------------|------------------------------------------------------------------------------------------------|--------|
| 7:6 | <i>Reserved</i> | <i>Reserved</i>      | -                                                                                              | -      |
| 5   | VD2EGS          | VD2 edge selection   | <ul style="list-style-type: none"> <li>0b0: VCC&lt;VPVD2</li> <li>0b1: VCC&gt;VPVD2</li> </ul> | RW     |
| 4   | VD1EGS          | VD1 edge selection   | <ul style="list-style-type: none"> <li>0b0: VCC&lt;VPVD1</li> <li>0b1: VCC&gt;VPVD1</li> </ul> | RW     |
| 3   | WK3EGS          | PTWK3 edge selection | <ul style="list-style-type: none"> <li>0b0: Falling edge</li> <li>0b1: Rising edge</li> </ul>  | RW     |
| 2   | WK2EGS          | PTWK2 edge selection | <ul style="list-style-type: none"> <li>0b0: Falling edge</li> <li>0b1: Rising edge</li> </ul>  | RW     |
| 1   | WK1EGS          | PTWK1 edge select    | <ul style="list-style-type: none"> <li>0b0: Falling edge</li> <li>0b1: Rising edge</li> </ul>  | RW     |
| 0   | WK0EGS          | PTWK0 edge selection | <ul style="list-style-type: none"> <li>0b0: Falling edge</li> <li>0b1: Rising edge</li> </ul>  | RW     |

## 6.3.11 Power-down Wake-up Flag Register 0 (PWC\_PDWKFO)

| Offset | Bit Position |   |        |        |        |        |        |        |
|--------|--------------|---|--------|--------|--------|--------|--------|--------|
| 0x4C38 | 7            | 6 | 5      | 4      | 3      | 2      | 1      | 0      |
| Reset  | 0x00         |   |        |        |        |        |        |        |
| Name   | Reserved     |   | VD2WKF | VD1WKF | PTWK3F | PTWK2F | PTWK1F | PTWK0F |
| Access |              |   | RW     | RW     | RW     | RW     | RW     | RW     |

| Bit | Name            | Function              | Description                                                                                                                                                                                                      | Access |
|-----|-----------------|-----------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 7:6 | <i>Reserved</i> | <i>Reserved</i>       | -                                                                                                                                                                                                                | -      |
| 5   | VD2WKF          | PVD2 wakeup flag bit  | After waking up from Power-down mode, the flag must be cleared by writing 0.<br>Read:<br><ul style="list-style-type: none"> <li>0b0: No wake-up event occurred.</li> <li>0b1: Wake-up event occurred.</li> </ul> | RW     |
| 4   | VD1WKF          | PVD1 wakeup flag bit  | After a Power-down wake-up, the flag must be cleared by writing 0.<br>Read:<br><ul style="list-style-type: none"> <li>0b0: No wake-up event occurred.</li> <li>0b1: Wake-up event occurred.</li> </ul>           | RW     |
| 3   | PTWK3F          | PTWK3 wakeup flag bit | After a Power-down wake-up, the flag must be cleared by writing 0.<br>Read:<br><ul style="list-style-type: none"> <li>0b0: No wake-up event occurred.</li> <li>0b1: Wake-up event occurred.</li> </ul>           | RW     |
| 2   | PTWK2F          | PTWK2 wakeup flag bit | After a Power-down wake-up, the flag must be cleared by writing 0.<br>Read:<br><ul style="list-style-type: none"> <li>0b0: No wake-up event occurred.</li> <li>0b1: Wake-up event occurred.</li> </ul>           | RW     |
| 1   | PTWK1F          | PTWK1 wakeup flag bit | After a Power-down wake-up, the flag must be cleared by writing 0.<br>Read:<br><ul style="list-style-type: none"> <li>0b0: No wake-up event occurred.</li> <li>0b1: Wake-up event occurred.</li> </ul>           | RW     |
| 0   | PTWK0F          | PTWK0 wakeup flag bit | After a Power-down wake-up, the flag must be cleared by writing 0.<br>Read:<br><ul style="list-style-type: none"> <li>0b0: No wake-up event occurred.</li> <li>0b1: Wake-up event occurred.</li> </ul>           | RW     |

## 6.3.12 Power-down Wake-up Flag Register 1 (PWC\_PDWKF1)

| Offset | Bit Position |          |           |           |          |   |   |   |
|--------|--------------|----------|-----------|-----------|----------|---|---|---|
| 0x4C3C | 7            | 6        | 5         | 4         | 3        | 2 | 1 | 0 |
| Reset  | 0x00         |          |           |           |          |   |   |   |
| Name   | WKTMWKF      | Reserved | RTCALMWKF | RTCPRDWKF | Reserved |   |   |   |
| Access | RW           |          | RW        | RW        |          |   |   |   |

| Bit | Name            | Function                        | Description                                                                                                                                                                                            | Access |
|-----|-----------------|---------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 7   | WKTMWKF         | WKTW wakeup flag bit            | After a Power-down wake-up, the flag must be cleared by writing 0.<br>Read:<br><ul style="list-style-type: none"> <li>0b0: No wake-up event occurred.</li> <li>0b1: Wake-up event occurred.</li> </ul> | RW     |
| 6   | <i>Reserved</i> | <i>Reserved</i>                 | -                                                                                                                                                                                                      | -      |
| 5   | RTCALMWKF       | RTC alarm clock wakeup flag bit | After a Power-down wake-up, the flag must be cleared by writing 0.<br>Read:<br><ul style="list-style-type: none"> <li>0b0: No wake-up event occurred.</li> <li>0b1: Wake-up event occurred.</li> </ul> | RW     |
| 4   | RTCPRDWKF       | RTC period wakeup flag bit      | After a Power-down wake-up, the flag must be cleared by writing 0.<br>Read:<br><ul style="list-style-type: none"> <li>0b0: No wake-up event occurred.</li> <li>0b1: Wake-up event occurred.</li> </ul> | RW     |
| 3:0 | <i>Reserved</i> | <i>Reserved</i>                 | -                                                                                                                                                                                                      | -      |

## 6.3.13 RAM Power Control Register 0 (PWC\_RAMPC0)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |          |         |         |         |         |         |         |         |         |         |         |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|
| 0x4CE0 | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10       | 9       | 8       | 7       | 6       | 5       | 4       | 3       | 2       | 1       | 0       |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |          |         |         |         |         |         |         |         |         |         |         |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RAMPDC10 | RAMPDC9 | RAMPDC8 | RAMPDC7 | RAMPDC6 | RAMPDC5 | RAMPDC4 | RAMPDC3 | RAMPDC2 | RAMPDC1 | RAMPDC0 |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RW       | RW      | RW      | RW      | RW      | RW      | RW      | RW      | RW      | RW      | RW      |

| Bit   | Name     | Function                      | Description                                                                                                                                            | Access |
|-------|----------|-------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:11 | Reserved | Reserved                      | -                                                                                                                                                      | -      |
| 10    | RAMPDC10 | RAM power down control bit 10 | <ul style="list-style-type: none"> <li>0b0: 0x1FFF8000-0x1FFFFFFF space RAM action</li> <li>0b1: 0x1FFF8000-0x1FFFFFFF space RAM power down</li> </ul> | RW     |
| 9     | RAMPDC9  | RAM power down control bit 9  | <ul style="list-style-type: none"> <li>0b0: 0x1FFF0000-0x1FFF7FFF space RAM action</li> <li>0b1: 0x1FFF0000-0x1FFF7FFF space RAM power down</li> </ul> | RW     |
| 8     | RAMPDC8  | RAM power down control bit 8  | <ul style="list-style-type: none"> <li>0b0: 0x1FFE8000-0x1FFEFFFF space RAM action</li> <li>0b1: 0x1FFE8000-0x1FFEFFFF space RAM power down</li> </ul> | RW     |
| 7     | RAMPDC7  | RAM power down control bit 7  | <ul style="list-style-type: none"> <li>0b0: 0x1FFE0000-0x1FFE7FFF space RAM action</li> <li>0b1: 0x1FFE0000-0x1FFE7FFF space RAM power down</li> </ul> | RW     |
| 6     | RAMPDC6  | RAM power down control bit 6  | <ul style="list-style-type: none"> <li>0b0: 0x20058000-0x2005FFFF space RAM action</li> <li>0b1: 0x20058000-0x2005FFFF space RAM power down</li> </ul> | RW     |
| 5     | RAMPDC5  | RAM power down control bit 5  | <ul style="list-style-type: none"> <li>0b0: 0x20050000-0x20057FFF space RAM action</li> <li>0b1: 0x20050000-0x20057FFF space RAM power down</li> </ul> | RW     |
| 4     | RAMPDC4  | RAM power down control bit 4  | <ul style="list-style-type: none"> <li>0b0: 0x20040000-0x2004FFFF space RAM action</li> <li>0b1: 0x20040000-0x2004FFFF space RAM power down</li> </ul> | RW     |
| 3     | RAMPDC3  | RAM power down control bit 3  | <ul style="list-style-type: none"> <li>0b0: 0x20030000-0x2003FFFF space RAM action</li> <li>0b1: 0x20030000-0x2003FFFF space RAM power down</li> </ul> | RW     |



| Bit | Name    | Function                     | Description                                                                                                                                            | Access |
|-----|---------|------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 2   | RAMPDC2 | RAM power down control bit 2 | <ul style="list-style-type: none"> <li>0b0: 0x20020000-0x2002FFFF space RAM action</li> <li>0b1: 0x20020000-0x2002FFFF space RAM power down</li> </ul> | RW     |
| 1   | RAMPDC1 | RAM power down control bit 1 | <ul style="list-style-type: none"> <li>0b0: 0x20010000-0x2001FFFF space RAM action</li> <li>0b1: 0x20010000-0x2001FFFF space RAM power down</li> </ul> | RW     |
| 0   | RAMPDC0 | RAM power down control bit 0 | <ul style="list-style-type: none"> <li>0b0: 0x20000000-0x2000FFFF space RAM action</li> <li>0b1: 0x20000000-0x2000FFFF space RAM power down</li> </ul> | RW     |

## 6.3.14 RAM Operating Condition Register (PWC\_RAMOPM)

| Offset | Bit Position |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0x4CE4 | 15           | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x8043       |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | PWC_RAMOPM   |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access | RW           |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit  | Name       | Function                      | Description                                                                                                                                                                                             | Access |
|------|------------|-------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 15:0 | PWC_RAMOPM | RAM motion mode selection Bit | <ul style="list-style-type: none"> <li>When the chip works in high-speed mode, PWC_RAMOPM is set to 0x8043</li> <li>When the chip works in ultra low-speed mode, PWC_RAMOPM is set to 0x9062</li> </ul> | RW     |

## 6.3.15 Peripheral RAM Low Power Control Register (PWC\_PRAMLPC)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |           |           |          |          |          |          |          |          |          |          |          |          |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-----------|-----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|
| 0x4CE8 | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11        | 10        | 9        | 8        | 7        | 6        | 5        | 4        | 3        | 2        | 1        | 0        |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |           |           |          |          |          |          |          |          |          |          |          |          |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | PRAMPDC11 | PRAMPDC10 | PRAMPDC9 | PRAMPDC8 | PRAMPDC7 | PRAMPDC6 | PRAMPDC5 | PRAMPDC4 | PRAMPDC3 | PRAMPDC2 | PRAMPDC1 | PRAMPDC0 |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RW        | RW        | RW       | RW       | RW       | RW       | RW       | RW       | RW       | RW       | RW       | RW       |

| Bit   | Name      | Function                           | Description                                                                                    | Access |
|-------|-----------|------------------------------------|------------------------------------------------------------------------------------------------|--------|
| 31:12 | Reserved  | Reserved                           | -                                                                                              | -      |
| 11    | PRAMPDC11 | PKE RAM power down control         | <ul style="list-style-type: none"> <li>0b0: RAM action</li> <li>0b1: RAM power down</li> </ul> | RW     |
| 10    | PRAMPDC10 | MCAN1/MCAN2 RAM power down control | <ul style="list-style-type: none"> <li>0b0: RAM action</li> <li>0b1: RAM power down</li> </ul> | RW     |
| 9     | PRAMPDC9  | NFC_RAM power down control         | <ul style="list-style-type: none"> <li>0b0: RAM action</li> <li>0b1: RAM power down</li> </ul> | RW     |
| 8     | PRAMPDC8  | SDIO2_RAM power down control       | <ul style="list-style-type: none"> <li>0b0: RAM action</li> <li>0b1: RAM power down</li> </ul> | RW     |
| 7     | PRAMPDC7  | SDIO1_RAM power down control       | <ul style="list-style-type: none"> <li>0b0: RAM action</li> <li>0b1: RAM power down</li> </ul> | RW     |
| 6     | PRAMPDC6  | ETHER RX RAM power down control    | <ul style="list-style-type: none"> <li>0b0: RAM action</li> <li>0b1: RAM power down</li> </ul> | RW     |
| 5     | PRAMPDC5  | ETHER TX RAM power down control    | <ul style="list-style-type: none"> <li>0b0: RAM action</li> <li>0b1: RAM power down</li> </ul> | RW     |
| 4     | PRAMPDC4  | USBHS RAM power down control       | <ul style="list-style-type: none"> <li>0b0: RAM action</li> <li>0b1: RAM power down</li> </ul> | RW     |
| 3     | PRAMPDC3  | USBFS RAM power down control       | <ul style="list-style-type: none"> <li>0b0: RAM action</li> <li>0b1: RAM power down</li> </ul> | RW     |
| 2     | PRAMPDC2  | CACHE_RAM power down control       | <ul style="list-style-type: none"> <li>0b0: RAM action</li> <li>0b1: RAM power down</li> </ul> | RW     |
| 1     | PRAMPDC1  | CAN_2 RAM power down control       | <ul style="list-style-type: none"> <li>0b0: RAM action</li> <li>0b1: RAM power down</li> </ul> | RW     |

| Bit | Name     | Function                     | Description                                                                                     | Access |
|-----|----------|------------------------------|-------------------------------------------------------------------------------------------------|--------|
| 0   | PRAMPDC0 | CAN_1 RAM power down control | <ul style="list-style-type: none"><li>● 0b0: RAM action</li><li>● 0b1: RAM power down</li></ul> | RW     |

## 6.3.16 Low Power Mode Status Register (PWC\_LPMCSR)

| Offset | Bit Position |    |    |    |    |    |   |   |   |   |   |   |   |   |   |      |
|--------|--------------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|------|
| 0x4CEC | 15           | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0    |
| Reset  | 0x0000       |    |    |    |    |    |   |   |   |   |   |   |   |   |   |      |
| Name   | Reserved     |    |    |    |    |    |   |   |   |   |   |   |   |   |   | LPMF |
| Access |              |    |    |    |    |    |   |   |   |   |   |   |   |   |   | RO   |

| Bit  | Name     | Function                | Description                                                                                                                                                                                                                                                                                                                        | Access |
|------|----------|-------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 15:1 | Reserved | Reserved                | -                                                                                                                                                                                                                                                                                                                                  | -      |
| 0    | LPMF     | Low power mode flag bit | <ul style="list-style-type: none"> <li>0b0: The chip has not entered Stop mode</li> <li>0b1: The chip has entered Stop mode</li> </ul> <div>  <b>Note</b><br/>           This bit is cleared when the chip enters Power-down mode.         </div> | RO     |

## 6.3.17 Low Power Mode Status Clear Register (PWC\_LPMCSR)

| Offset | Bit Position |    |    |    |    |    |   |   |   |   |   |   |   |   |   |       |
|--------|--------------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|-------|
| 0x4CEE | 15           | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0     |
| Reset  | 0x0000       |    |    |    |    |    |   |   |   |   |   |   |   |   |   |       |
| Name   | Reserved     |    |    |    |    |    |   |   |   |   |   |   |   |   |   | LPMFC |
| Access |              |    |    |    |    |    |   |   |   |   |   |   |   |   |   | RW    |

| Bit  | Name     | Function        | Description                                                          | Access |
|------|----------|-----------------|----------------------------------------------------------------------|--------|
| 15:1 | Reserved | Reserved        | -                                                                    | -      |
| 0    | LPMFC    | LPMF flag clear | Write 1 to clear flag, write 0 has no effect; read always returns 0. | RW     |

## 6.3.18 PVD Control Register 0 (PWC\_PVDCR0)

| Offset | Bit Position |        |        |          |   |   |   |           |
|--------|--------------|--------|--------|----------|---|---|---|-----------|
| 0x4C14 | 7            | 6      | 5      | 4        | 3 | 2 | 1 | 0         |
| Reset  | 0x00         |        |        |          |   |   |   |           |
| Name   | Reserved     | PVD2EN | PVD1EN | Reserved |   |   |   | EXVCCINEN |
| Access |              | RW     | RW     |          |   |   |   | RW        |

| Bit | Name            | Function                          | Description                                                                                                                                   | Access |
|-----|-----------------|-----------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 7   | <i>Reserved</i> | <i>Reserved</i>                   | -                                                                                                                                             | -      |
| 6   | PVD2EN          | Voltage detection 2 permissible   | <ul style="list-style-type: none"> <li>0b0: Voltage detection 2 circuit disabled</li> <li>0b1: Voltage detection 2 circuit enabled</li> </ul> | RW     |
| 5   | PVD1EN          | Voltage detection 1 permissible   | <ul style="list-style-type: none"> <li>0b0: Voltage detection 1 circuit disabled</li> <li>0b1: Voltage detection 1 circuit enabled</li> </ul> | RW     |
| 4:1 | <i>Reserved</i> | <i>Reserved</i>                   | -                                                                                                                                             | -      |
| 0   | EXVCCINEN       | External VCC voltage input enable | <ul style="list-style-type: none"> <li>0b0: External VCC voltage input disabled</li> <li>0b1: External VCC voltage input enabled</li> </ul>   | RW     |

## 6.3.19 PVD Control Register 1 (PWC\_PVDCR1)

| Offset | Bit Position |           |         |         |          |           |         |         |
|--------|--------------|-----------|---------|---------|----------|-----------|---------|---------|
| 0x4C18 | 7            | 6         | 5       | 4       | 3        | 2         | 1       | 0       |
| Reset  | 0x00         |           |         |         |          |           |         |         |
| Name   | Reserved     | PVD2CMPOE | PVD2IRS | PVD2IRE | Reserved | PVD1CMPOE | PVD1IRS | PVD1IRE |
| Access |              | RW        | RW      | RW      |          | RW        | RW      | RW      |

| Bit | Name            | Function                           | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | Access |
|-----|-----------------|------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 7   | <i>Reserved</i> | <i>Reserved</i>                    | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | -      |
| 6   | PVD2CMPOE       | PVD2 compare result output enabled | <ul style="list-style-type: none"> <li>0b0: Disable output of comparison result of comparator 2</li> <li>0b1: Enable output the comparison result of comparator 2</li> </ul>                                                                                                                                                                                                                                                                                                                                        | RW     |
| 5   | PVD2IRS         | PVD2 interrupt reset selection     | <ul style="list-style-type: none"> <li>0b0: Generate PVD2 interrupt when VCC changes to meet the detection conditions</li> <li>0b1: PVD2 reset is generated during the falling process through VPVD2</li> </ul> <div>  <b>Note</b><br/>           When the PVD1IRS bit is 1 or the PVD2IRS bit is 1, the Power-down mode cannot be entered. To enter PD mode, both the PVD1IRS and PVD2IRS bits must be set to 0.         </div> | RW     |
| 4   | PVD2IRE         | PVD2 interrupt reset enabled       | <ul style="list-style-type: none"> <li>0b0: Disable</li> <li>0b1: Enable</li> </ul> <div>  <b>Note</b><br/>           Please write 1 to this bit when the PWC_PVDCR0.PVD2EN bit is 1 and the PVD2CMPOE bit is 1.         </div>                                                                                                                                                                                                  | RW     |
| 3   | <i>Reserved</i> | <i>Reserved</i>                    | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | -      |
| 2   | PVD1CMPOE       | PVD1 compare result output enabled | <ul style="list-style-type: none"> <li>0b0: Disable output of comparison result of comparator 1</li> <li>0b1: Enable output the comparison result of comparator 1</li> </ul>                                                                                                                                                                                                                                                                                                                                        | RW     |

| Bit | Name    | Function                       | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | Access |
|-----|---------|--------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 1   | PVD1IRS | PVD1 interrupt reset selection | <ul style="list-style-type: none"> <li>● 0b0: Generate PVD1 interrupt when VCC changes to meet the detection conditions</li> <li>● 0b1: PVD1 reset is generated during the falling process through VPVD1</li> </ul> <div>  <b>Note</b><br/>           When the PVD1IRS bit is 1 or the PVD2IRS bit is 1, the Power-down mode cannot be entered. To enter PD mode, both the PVD1IRS and PVD2IRS bits must be set to 0.         </div> | RW     |
| 0   | PVD1IRE | PVD1 interrupt reset enabled   | <ul style="list-style-type: none"> <li>● 0b0: Disable</li> <li>● 0b1: Enable</li> </ul> <div>  <b>Note</b><br/>           Please write 1 to this bit when the PWC_PVDCR0.PVD1EN bit is 1 and the PVD1CMPOE bit is 1.         </div>                                                                                                                                                                                                  | RW     |



## 6.3.20 PVD Filter Control Register (PWC\_PVDFCR)

| Offset | Bit Position |           |   |           |          |           |   |           |
|--------|--------------|-----------|---|-----------|----------|-----------|---|-----------|
| 0x4C1C | 7            | 6         | 5 | 4         | 3        | 2         | 1 | 0         |
| Reset  | 0x11         |           |   |           |          |           |   |           |
| Name   | Reserved     | PVD2NFCKS |   | PVD2NFDIS | Reserved | PVD1NFCKS |   | PVD1NFDIS |
| Access |              | RW        |   | RW        |          | RW        |   | RW        |

| Bit | Name            | Function                                     | Description                                                                                                                                                                                                                                                                                                                                                                                                                                   | Access |
|-----|-----------------|----------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 7   | <i>Reserved</i> | <i>Reserved</i>                              | -                                                                                                                                                                                                                                                                                                                                                                                                                                             | -      |
| 6:5 | PVD2NFCKS       | PVD2 Digital Filter Sampling Clock Selection | <ul style="list-style-type: none"> <li>● 0b00: 2 filter clock periods</li> <li>● 0b01: 4 filter clock periods</li> <li>● 0b10: 8 filter clock periods</li> <li>● 0b11: 16 filter clock periods</li> </ul> Filter clock period=LRC period/8<br><div>  <b>Note</b><br/>           This bit can only be rewritten when the PVD2NFDIS bit is 1.         </div>  | RW     |
| 4   | PVD2NFDIS       | PVD2 Digital Filter Shield                   | <ul style="list-style-type: none"> <li>● 0b0: Digital filter enabled</li> <li>● 0b1: Digital filter disabled</li> </ul>                                                                                                                                                                                                                                                                                                                       | RW     |
| 3   | <i>Reserved</i> | <i>Reserved</i>                              | -                                                                                                                                                                                                                                                                                                                                                                                                                                             | -      |
| 2:1 | PVD1NFCKS       | PVD1 Digital Filter Sampling Clock Selection | <ul style="list-style-type: none"> <li>● 0b00: 2 filter clock periods</li> <li>● 0b01: 4 filter clock periods</li> <li>● 0b10: 8 filter clock periods</li> <li>● 0b11: 16 filter clock periods</li> </ul> Filter clock period=LRC period/8<br><div>  <b>Note</b><br/>           This bit can only be rewritten when the PVD1NFDIS bit is 1.         </div> | RW     |
| 0   | PVD1NFDIS       | PVD1 Digital Filter Shield                   | <ul style="list-style-type: none"> <li>● 0b0: Digital filter enabled</li> <li>● 0b1: Digital filter disabled</li> </ul>                                                                                                                                                                                                                                                                                                                       | RW     |

## 6.3.21 PVD Level Control Register (PWC\_PVDLCR)

| Offset | Bit Position |         |   |   |          |         |   |   |
|--------|--------------|---------|---|---|----------|---------|---|---|
| 0x4C20 | 7            | 6       | 5 | 4 | 3        | 2       | 1 | 0 |
| Reset  | 0x00         |         |   |   |          |         |   |   |
| Name   | Reserved     | PVD2LVL |   |   | Reserved | PVD1LVL |   |   |
| Access |              | RW      |   |   |          | RW      |   |   |

| Bit | Name            | Function                         | Description                                                                                                                                                                                                                                                                                                                        | Access |
|-----|-----------------|----------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 7   | <i>Reserved</i> | <i>Reserved</i>                  | -                                                                                                                                                                                                                                                                                                                                  | -      |
| 6:4 | PVD2LVL         | PVD2 Threshold Voltage Selection | <ul style="list-style-type: none"> <li>● 0b000: 2.20V</li> <li>● 0b001: 2.40V</li> <li>● 0b010: 2.60V</li> <li>● 0b011: 2.70V</li> <li>● 0b100: 2.85V</li> <li>● 0b101: 2.95V</li> <li>● 0b110: 3.05V</li> <li>● 0b111: 1.15V (only active when PWC_PVDCR0.EXVCCINEN= 0b1, please do not set this value in other cases)</li> </ul> | RW     |
| 3   | <i>Reserved</i> | <i>Reserved</i>                  | -                                                                                                                                                                                                                                                                                                                                  | -      |
| 2:0 | PVD1LVL         | PVD1 Threshold Voltage Selection | <ul style="list-style-type: none"> <li>● 0b000: 2.10V</li> <li>● 0b001: 2.20V</li> <li>● 0b010: 2.40V</li> <li>● 0b011: 2.60V</li> <li>● 0b100: 2.70V</li> <li>● 0b101: 2.80V</li> <li>● 0b110: 2.95V</li> <li>● 0b111: 3.05V</li> </ul>                                                                                           | RW     |

## 6.3.22 PVD Interrupt Control Register (PWC\_PVDICR)

| Offset | Bit Position |          |   |          |          |          |   |          |
|--------|--------------|----------|---|----------|----------|----------|---|----------|
| 0x4CF0 | 7            | 6        | 5 | 4        | 3        | 2        | 1 | 0        |
| Reset  | 0x00         |          |   |          |          |          |   |          |
| Name   | Reserved     | PVD2EDGS |   | PVD2NMIS | Reserved | PVD1EDGS |   | PVD1NMIS |
| Access |              | RW       |   | RW       |          | RW       |   | RW       |

| Bit | Name            | Function                           | Description                                                                                                                                                                                                                                                                                                                                          | Access |
|-----|-----------------|------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 7   | <i>Reserved</i> | <i>Reserved</i>                    | -                                                                                                                                                                                                                                                                                                                                                    | -      |
| 6:5 | PVD2EDGS        | PVD2 detection condition selection | <ul style="list-style-type: none"> <li>0b00: When <math>VCC &lt; VPVD2</math> (falling) is detected</li> <li>0b01: When <math>VCC \geq VPVD2</math> (rising) is detected</li> <li>0b10: When <math>VCC &lt; VPVD2</math> (falling) is detected or when <math>VCC \geq VPVD2</math> (rising) is detected</li> <li>0b11: Setting prohibited</li> </ul> | RW     |
| 4   | PVD2NMIS        | PVD2 interrupt type selection      | <ul style="list-style-type: none"> <li>0b0: PVD2 interrupt as non-maskable interrupt</li> <li>0b1: PVD2 interrupt as maskable interrupt</li> </ul>                                                                                                                                                                                                   | RW     |
| 3   | <i>Reserved</i> | <i>Reserved</i>                    | -                                                                                                                                                                                                                                                                                                                                                    | -      |
| 2:1 | PVD1EDGS        | PVD1 detection condition selection | <ul style="list-style-type: none"> <li>0b00: When <math>VCC &lt; VPVD1</math> (falling) is detected</li> <li>0b01: When <math>VCC \geq VPVD1</math> (rising) is detected</li> <li>0b10: When <math>VCC &lt; VPVD1</math> (falling) is detected or when <math>VCC \geq VPVD1</math> (rising) is detected</li> <li>0b11: Setting prohibited</li> </ul> | RW     |
| 0   | PVD1NMIS        | PVD1 interrupt type selection      | <ul style="list-style-type: none"> <li>0b0: PVD1 interrupt as non-maskable interrupt</li> <li>0b1: PVD1 interrupt as maskable interrupt</li> </ul>                                                                                                                                                                                                   | RW     |

## 6.3.23 PVD Detection Status Register (PWC\_PVDDSR)

| Offset | Bit Position |   |            |         |          |   |            |         |
|--------|--------------|---|------------|---------|----------|---|------------|---------|
| 0x4CF4 | 7            | 6 | 5          | 4       | 3        | 2 | 1          | 0       |
| Reset  | 0x11         |   |            |         |          |   |            |         |
| Name   | Reserved     |   | PVD2DETFLG | PVD2MON | Reserved |   | PVD1DETFLG | PVD1MON |
| Access |              |   | RW         | RO      |          |   | RW         | RO      |

| Bit | Name       | Function                | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | Access |
|-----|------------|-------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 7:6 | Reserved   | Reserved                | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | -      |
| 5   | PVD2DETFLG | PVD2 Detection Flag Bit | <p>After reading this bit, write 0 to clear this flag bit; write 1 has no effect</p> <p>Read:</p> <ul style="list-style-type: none"> <li>0b0: PVD2 does not detect VCC passing through VPVD2</li> <li>0b1: PVD2 detects that VCC passes through VPVD2</li> </ul> <div>  <b>Note</b><br/>           This bit is valid when the PWC_PVDCR0.PVD2EN bit is 1 and the PWC_PVDCR1.PVD2CMPOE bit is 1.         </div> | RW     |
| 4   | PVD2MON    | PVD2 Monitoring Bit     | <ul style="list-style-type: none"> <li>0b0: <math>VCC \leq VPVD2</math> or external input comparison voltage <math>\leq</math> PVD2 internal reference voltage</li> <li>0b1: When PVD2 is disabled or <math>VCC &gt; VPVD2</math> or external input comparison voltage <math>&gt;</math> PVD2 internal reference voltage</li> </ul>                                                                                                                                                               | RO     |
| 3:2 | Reserved   | Reserved                | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | -      |
| 1   | PVD1DETFLG | PVD1 Detection Flag Bit | <p>After reading this bit, write 0 to clear this flag bit; write 1 has no effect</p> <p>Read:</p> <ul style="list-style-type: none"> <li>0b0: PVD1 does not detect VCC passing through VPVD1</li> <li>0b1: PVD1 detects that VCC passes through VPVD1</li> </ul> <div>  <b>Note</b><br/>           This bit is valid when the PWC_PVDCR0.PVD1EN bit is 1 and the PWC_PVDCR1.PVD1CMPOE bit is 1.         </div> | RW     |
| 0   | PVD1MON    | PVD1 Monitoring Bit     | <ul style="list-style-type: none"> <li>0b0: <math>VCC &lt; VPVD1</math></li> <li>0b1: When PVD1 is disabled or <math>VCC &gt; VPVD1</math></li> </ul>                                                                                                                                                                                                                                                                                                                                             | RO     |

## 6.3.24 Function Clock Control Register 0 (PWC\_FCG0)

| Offset | Bit Position |      |      |      |      |      |      |      |     |      |      |     |     |     |     |     |      |      |         |     |          |       |          |      |       |       |       |       |          |   |   |       |
|--------|--------------|------|------|------|------|------|------|------|-----|------|------|-----|-----|-----|-----|-----|------|------|---------|-----|----------|-------|----------|------|-------|-------|-------|-------|----------|---|---|-------|
| 0x0000 | 31           | 30   | 29   | 28   | 27   | 26   | 25   | 24   | 23  | 22   | 21   | 20  | 19  | 18  | 17  | 16  | 15   | 14   | 13      | 12  | 11       | 10    | 9        | 8    | 7     | 6     | 5     | 4     | 3        | 2 | 1 | 0     |
| Reset  | 0xFFFFFA0E   |      |      |      |      |      |      |      |     |      |      |     |     |     |     |     |      |      |         |     |          |       |          |      |       |       |       |       |          |   |   |       |
| Name   | DCU8         | DCU7 | DCU6 | DCU5 | DCU4 | DCU3 | DCU2 | DCU1 | CRC | TRNG | HASH | SKE | MAU | CTC | AOS | FCM | DMA2 | DMA1 | KEYSCAN | PKE | Reserved | SRAMB | Reserved | ERMU | SRAM4 | SRAM3 | SRAM2 | SRAM1 | Reserved |   |   | SRAMH |
| Access | RW           | RW   | RW   | RW   | RW   | RW   | RW   | RW   | RW  | RW   | RW   | RW  | RW  | RW  | RW  | RW  | RW   | RW   | RW      | RW  |          | RW    |          | RW   | RW    | RW    | RW    | RW    | RW       |   |   | RW    |

| Bit | Name | Function              | Description                                                                                                                                                                                                                               | Access |
|-----|------|-----------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31  | DCU8 | DCU8 Function Control | <ul style="list-style-type: none"> <li>0b0: Function enabled</li> <li>0b1: Function disabled</li> </ul>                                                                                                                                   | RW     |
| 30  | DCU7 | DCU7 Function Control | <ul style="list-style-type: none"> <li>0b0: Function enabled</li> <li>0b1: Function disabled</li> </ul>                                                                                                                                   | RW     |
| 29  | DCU6 | DCU6 Function Control | <ul style="list-style-type: none"> <li>0b0: Function enabled</li> <li>0b1: Function disabled</li> </ul>                                                                                                                                   | RW     |
| 28  | DCU5 | DCU5 Function Control | <ul style="list-style-type: none"> <li>0b0: Function enabled</li> <li>0b1: Function disabled</li> </ul>                                                                                                                                   | RW     |
| 27  | DCU4 | DCU4 Function Control | <ul style="list-style-type: none"> <li>0b0: Function enabled</li> <li>0b1: Function disabled</li> </ul>                                                                                                                                   | RW     |
| 26  | DCU3 | DCU3 Function Control | <ul style="list-style-type: none"> <li>0b0: Function enabled</li> <li>0b1: Function disabled</li> </ul>                                                                                                                                   | RW     |
| 25  | DCU2 | DCU2 Function Control | <ul style="list-style-type: none"> <li>0b0: Function enabled</li> <li>0b1: Function disabled</li> </ul>                                                                                                                                   | RW     |
| 24  | DCU1 | DCU1 Function Control | <ul style="list-style-type: none"> <li>0b0: Function enabled</li> <li>0b1: Function disabled</li> </ul>                                                                                                                                   | RW     |
| 23  | CRC  | CRC Function Control  | <ul style="list-style-type: none"> <li>0b0: Function enabled</li> <li>0b1: Function disabled</li> </ul>                                                                                                                                   | RW     |
| 22  | TRNG | TRNG Function Control | <ul style="list-style-type: none"> <li>0b0: Enables the True Random Number Generator (TRNG) function in the CPM module.</li> <li>0b1: Disables the True Random Number Generator (TRNG) function in the CPM module.</li> </ul>             | RW     |
| 21  | HASH | HASH Function Control | <ul style="list-style-type: none"> <li>0b0: Enables the Secure Hash Algorithm (HASH) function in the CPM.</li> <li>0b1: Disables the Secure Hash Algorithm (HASH) function in the CPM.</li> </ul>                                         | RW     |
| 20  | SKE  | SKE Function Control  | <ul style="list-style-type: none"> <li>0b0: Enables the Symmetric Encryption Algorithm Module SKE function in the CPM module.</li> <li>0b1: Disables the Symmetric Encryption Algorithm Module SKE function in the CPM module.</li> </ul> | RW     |

| Bit | Name            | Function                  | Description                                                                                                                                                                                                                       | Access |
|-----|-----------------|---------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 19  | MAU             | MAU Function Control      | <ul style="list-style-type: none"> <li>● 0b0: Function enabled</li> <li>● 0b1: Function disabled</li> </ul>                                                                                                                       | RW     |
| 18  | CTC             | CTC Function Control      | <ul style="list-style-type: none"> <li>● 0b0: Function enabled</li> <li>● 0b1: Function disabled</li> </ul>                                                                                                                       | RW     |
| 17  | AOS             | AOS Function Control      | <ul style="list-style-type: none"> <li>● 0b0: Function enabled</li> <li>● 0b1: Function disabled</li> </ul>                                                                                                                       | RW     |
| 16  | FCM             | FCM Function Control      | <ul style="list-style-type: none"> <li>● 0b0: Enables the Frequency Counter Measurement (FCM) function in the CMU module.</li> <li>● 0b1: Disables the Frequency Counter Measurement (FCM) function in the CMU module.</li> </ul> | RW     |
| 15  | DMA2            | DMA2 Function Control     | <ul style="list-style-type: none"> <li>● 0b0: Function enabled</li> <li>● 0b1: Function disabled</li> </ul>                                                                                                                       | RW     |
| 14  | DMA1            | DMA1 Function Control     | <ul style="list-style-type: none"> <li>● 0b0: Function enabled</li> <li>● 0b1: Function disabled</li> </ul>                                                                                                                       | RW     |
| 13  | KEYSCAN         | KEYSCAN Function Control  | <ul style="list-style-type: none"> <li>● 0b0: Function enabled</li> <li>● 0b1: Function disabled</li> </ul>                                                                                                                       | RW     |
| 12  | PKE             | PKE Function Control      | <ul style="list-style-type: none"> <li>● 0b0: Enables the Public Key Encryption (PKE) function in the CPM module.</li> <li>● 0b1: Disables the Public Key Encryption (PKE) function in the CPM module.</li> </ul>                 | RW     |
| 11  | <i>Reserved</i> | <i>Reserved</i>           | <i>Reads as 1, Write as 1</i>                                                                                                                                                                                                     | -      |
| 10  | SRAMB           | Ret SRAM Function Control | <ul style="list-style-type: none"> <li>● 0b0: Function enabled</li> <li>● 0b1: Function disabled</li> </ul>                                                                                                                       | RW     |
| 9   | <i>Reserved</i> | <i>Reserved</i>           | <i>Reads as 1, Write as 1</i>                                                                                                                                                                                                     | -      |
| 8   | ERMU            | ERMU Function Control     | <ul style="list-style-type: none"> <li>● 0b0: Function enabled</li> <li>● 0b1: Function disabled</li> </ul>                                                                                                                       | RW     |
| 7   | SRAM4           | SRAM4 Function Control    | <ul style="list-style-type: none"> <li>● 0b0: Function enabled</li> <li>● 0b1: Function disabled</li> </ul>                                                                                                                       | RW     |
| 6   | SRAM3           | SRAM3 Function Control    | <ul style="list-style-type: none"> <li>● 0b0: Function enabled</li> <li>● 0b1: Function disabled</li> </ul>                                                                                                                       | RW     |
| 5   | SRAM2           | SRAM2 Function Control    | <ul style="list-style-type: none"> <li>● 0b0: Function enabled</li> <li>● 0b1: Function disabled</li> </ul>                                                                                                                       | RW     |
| 4   | SRAM1           | SRAM1 Function Control    | <ul style="list-style-type: none"> <li>● 0b0: Function enabled</li> <li>● 0b1: Function disabled</li> </ul>                                                                                                                       | RW     |
| 3:1 | <i>Reserved</i> | <i>Reserved</i>           | <i>All bits read as 1, Write as 1</i>                                                                                                                                                                                             | -      |
| 0   | SRAMH           | SRAMH Function Control    | <ul style="list-style-type: none"> <li>● 0b0: Function enabled</li> <li>● 0b1: Function disabled</li> </ul>                                                                                                                       | RW     |

## 6.3.25 Function Clock Control Register 1 (PWC\_FCG1)

| Offset | Bit Position |       |       |       |       |       |       |       |       |      |      |      |      |      |      |      |      |      |      |        |        |      |      |      |      |      |      |      |        |      |      |    |    |
|--------|--------------|-------|-------|-------|-------|-------|-------|-------|-------|------|------|------|------|------|------|------|------|------|------|--------|--------|------|------|------|------|------|------|------|--------|------|------|----|----|
| 0x0004 | 31           | 30    | 29    | 28    | 27    | 26    | 25    | 24    | 23    | 22   | 21   | 20   | 19   | 18   | 17   | 16   | 15   | 14   | 13   | 12     | 11     | 10   | 9    | 8    | 7    | 6    | 5    | 4    | 3      | 2    | 1    | 0  |    |
| Reset  | 0xFFFFFFFF   |       |       |       |       |       |       |       |       |      |      |      |      |      |      |      |      |      |      |        |        |      |      |      |      |      |      |      |        |      |      |    |    |
| Name   | Reserved     | MCAN2 | MCAN1 | FMAC4 | FMAC3 | FMAC2 | FMAC1 | USBHS | USBFS | SPI6 | SPI5 | SPI4 | SPI3 | SPI2 | SPI1 | I2S4 | I2S3 | I2S2 | I2S1 | SDIOC2 | SDIOC1 | I2C6 | I2C5 | I2C4 | I2C3 | I2C2 | I2C1 | QSPI | ETHMAC | CAN2 | CAN1 |    |    |
| Access |              | RW    | RW    | RW    | RW    | RW    | RW    | RW    | RW    | RW   | RW   | RW   | RW   | RW   | RW   | RW   | RW   | RW   | RW   | RW     | RW     | RW   | RW   | RW   | RW   | RW   | RW   | RW   | RW     | RW   | RW   | RW | RW |

| Bit   | Name     | Function               | Description                                                                                             | Access |
|-------|----------|------------------------|---------------------------------------------------------------------------------------------------------|--------|
| 31:30 | Reserved | Reserved               | All bits read as 1, Write as 1                                                                          | -      |
| 29    | MCAN2    | MCAN2 Function Control | <ul style="list-style-type: none"> <li>0b0: Function enabled</li> <li>0b1: Function disabled</li> </ul> | RW     |
| 28    | MCAN1    | MCAN1 Function Control | <ul style="list-style-type: none"> <li>0b0: Function enabled</li> <li>0b1: Function disabled</li> </ul> | RW     |
| 27    | FMAC4    | FMAC4 Function Control | <ul style="list-style-type: none"> <li>0b0: Function enabled</li> <li>0b1: Function disabled</li> </ul> | RW     |
| 26    | FMAC3    | FMAC3 Function Control | <ul style="list-style-type: none"> <li>0b0: Function enabled</li> <li>0b1: Function disabled</li> </ul> | RW     |
| 25    | FMAC2    | FMAC2 Function Control | <ul style="list-style-type: none"> <li>0b0: Function enabled</li> <li>0b1: Function disabled</li> </ul> | RW     |
| 24    | FMAC1    | FMAC1 Function Control | <ul style="list-style-type: none"> <li>0b0: Function enabled</li> <li>0b1: Function disabled</li> </ul> | RW     |
| 23    | USBHS    | USBHS Function Control | <ul style="list-style-type: none"> <li>0b0: Function enabled</li> <li>0b1: Function disabled</li> </ul> | RW     |
| 22    | USBFS    | USBFS Function Control | <ul style="list-style-type: none"> <li>0b0: Function enabled</li> <li>0b1: Function disabled</li> </ul> | RW     |
| 21    | SPI6     | SPI6 Function Control  | <ul style="list-style-type: none"> <li>0b0: Function enabled</li> <li>0b1: Function disabled</li> </ul> | RW     |
| 20    | SPI5     | SPI5 Function Control  | <ul style="list-style-type: none"> <li>0b0: Function enabled</li> <li>0b1: Function disabled</li> </ul> | RW     |
| 19    | SPI4     | SPI4 Function Control  | <ul style="list-style-type: none"> <li>0b0: Function enabled</li> <li>0b1: Function disabled</li> </ul> | RW     |
| 18    | SPI3     | SPI3 Function Control  | <ul style="list-style-type: none"> <li>0b0: Function enabled</li> <li>0b1: Function disabled</li> </ul> | RW     |
| 17    | SPI2     | SPI2 Function Control  | <ul style="list-style-type: none"> <li>0b0: Function enabled</li> <li>0b1: Function disabled</li> </ul> | RW     |

| Bit | Name   | Function                | Description                                                                                                 | Access |
|-----|--------|-------------------------|-------------------------------------------------------------------------------------------------------------|--------|
| 16  | SPI1   | SPI1 Function Control   | <ul style="list-style-type: none"> <li>● 0b0: Function enabled</li> <li>● 0b1: Function disabled</li> </ul> | RW     |
| 15  | I2S4   | I2S4 Function Control   | <ul style="list-style-type: none"> <li>● 0b0: Function enabled</li> <li>● 0b1: Function disabled</li> </ul> | RW     |
| 14  | I2S3   | I2S3 Function Control   | <ul style="list-style-type: none"> <li>● 0b0: Function enabled</li> <li>● 0b1: Function disabled</li> </ul> | RW     |
| 13  | I2S2   | I2S2 Function Control   | <ul style="list-style-type: none"> <li>● 0b0: Function enabled</li> <li>● 0b1: Function disabled</li> </ul> | RW     |
| 12  | I2S1   | I2S1 Function Control   | <ul style="list-style-type: none"> <li>● 0b0: Function enabled</li> <li>● 0b1: Function disabled</li> </ul> | RW     |
| 11  | SDIOC2 | SDIOC2 Function Control | <ul style="list-style-type: none"> <li>● 0b0: Function enabled</li> <li>● 0b1: Function disabled</li> </ul> | RW     |
| 10  | SDIOC1 | SDIOC1 Function Control | <ul style="list-style-type: none"> <li>● 0b0: Function enabled</li> <li>● 0b1: Function disabled</li> </ul> | RW     |
| 9   | I2C6   | I2C6 Function Control   | <ul style="list-style-type: none"> <li>● 0b0: Function enabled</li> <li>● 0b1: Function disabled</li> </ul> | RW     |
| 8   | I2C5   | I2C5 Function Control   | <ul style="list-style-type: none"> <li>● 0b0: Function enabled</li> <li>● 0b1: Function disabled</li> </ul> | RW     |
| 7   | I2C4   | I2C4 Function Control   | <ul style="list-style-type: none"> <li>● 0b0: Function enabled</li> <li>● 0b1: Function disabled</li> </ul> | RW     |
| 6   | I2C3   | I2C3 Function Control   | <ul style="list-style-type: none"> <li>● 0b0: Function enabled</li> <li>● 0b1: Function disabled</li> </ul> | RW     |
| 5   | I2C2   | I2C2 Function Control   | <ul style="list-style-type: none"> <li>● 0b0: Function enabled</li> <li>● 0b1: Function disabled</li> </ul> | RW     |
| 4   | I2C1   | I2C1 Function Control   | <ul style="list-style-type: none"> <li>● 0b0: Function enabled</li> <li>● 0b1: Function disabled</li> </ul> | RW     |
| 3   | QSPI   | QSPI Function Control   | <ul style="list-style-type: none"> <li>● 0b0: Function enabled</li> <li>● 0b1: Function disabled</li> </ul> | RW     |
| 2   | ETHMAC | ETHMAC Function Control | <ul style="list-style-type: none"> <li>● 0b0: Function enabled</li> <li>● 0b1: Function disabled</li> </ul> | RW     |
| 1   | CAN2   | CAN2 Function Control   | <ul style="list-style-type: none"> <li>● 0b0: Function enabled</li> <li>● 0b1: Function disabled</li> </ul> | RW     |
| 0   | CAN1   | CAN1 Function Control   | <ul style="list-style-type: none"> <li>● 0b0: Function enabled</li> <li>● 0b1: Function disabled</li> </ul> | RW     |



## 6.3.26 Function Clock Control Register 2 (PWC\_FCG2)

| Offset | Bit Position |         |         |        |        |        |        |        |        |        |        |        |        |        |        |        |     |        |        |        |          |        |        |        |        |        |        |        |        |        |        |        |    |
|--------|--------------|---------|---------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|-----|--------|--------|--------|----------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|----|
| 0x0008 | 31           | 30      | 29      | 28     | 27     | 26     | 25     | 24     | 23     | 22     | 21     | 20     | 19     | 18     | 17     | 16     | 15  | 14     | 13     | 12     | 11       | 10     | 9      | 8      | 7      | 6      | 5      | 4      | 3      | 2      | 1      | 0      |    |
| Reset  | 0xFFFFFFFF   |         |         |        |        |        |        |        |        |        |        |        |        |        |        |        |     |        |        |        |          |        |        |        |        |        |        |        |        |        |        |        |    |
| Name   | TMRA_12      | TMRA_11 | TMRA_10 | TMRA_9 | TMRA_8 | TMRA_7 | TMRA_6 | TMRA_5 | TMRA_4 | TMRA_3 | TMRA_2 | TMRA_1 | TMR2_4 | TMR2_3 | TMR2_2 | TMR2_1 | EMB | TMR0_3 | TMR0_2 | TMR0_1 | Reserved | TMR4_3 | TMR4_2 | TMR4_1 | TMR6_8 | TMR6_7 | TMR6_6 | TMR6_5 | TMR6_4 | TMR6_3 | TMR6_2 | TMR6_1 |    |
| Access | RW           | RW      | RW      | RW     | RW     | RW     | RW     | RW     | RW     | RW     | RW     | RW     | RW     | RW     | RW     | RW     | RW  | RW     | RW     | RW     |          | RW     | RW     | RW     | RW     | RW     | RW     | RW     | RW     | RW     | RW     | RW     | RW |

| Bit | Name    | Function                   | Description                                                                                             | Access |
|-----|---------|----------------------------|---------------------------------------------------------------------------------------------------------|--------|
| 31  | TMRA_12 | TimerA_12 Function Control | <ul style="list-style-type: none"> <li>0b0: Function enabled</li> <li>0b1: Function disabled</li> </ul> | RW     |
| 30  | TMRA_11 | TimerA_11 Function Control | <ul style="list-style-type: none"> <li>0b0: Function enabled</li> <li>0b1: Function disabled</li> </ul> | RW     |
| 29  | TMRA_10 | TimerA_10 Function Control | <ul style="list-style-type: none"> <li>0b0: Function enabled</li> <li>0b1: Function disabled</li> </ul> | RW     |
| 28  | TMRA_9  | TimerA_9 Function Control  | <ul style="list-style-type: none"> <li>0b0: Function enabled</li> <li>0b1: Function disabled</li> </ul> | RW     |
| 27  | TMRA_8  | TimerA_8 Function Control  | <ul style="list-style-type: none"> <li>0b0: Function enabled</li> <li>0b1: Function disabled</li> </ul> | RW     |
| 26  | TMRA_7  | TimerA_7 Function Control  | <ul style="list-style-type: none"> <li>0b0: Function enabled</li> <li>0b1: Function disabled</li> </ul> | RW     |
| 25  | TMRA_6  | TimerA_6 Function Control  | <ul style="list-style-type: none"> <li>0b0: Function enabled</li> <li>0b1: Function disabled</li> </ul> | RW     |
| 24  | TMRA_5  | TimerA_5 Function Control  | <ul style="list-style-type: none"> <li>0b0: Function enabled</li> <li>0b1: Function disabled</li> </ul> | RW     |
| 23  | TMRA_4  | TimerA_4 Function Control  | <ul style="list-style-type: none"> <li>0b0: Function enabled</li> <li>0b1: Function disabled</li> </ul> | RW     |
| 22  | TMRA_3  | TimerA_3 Function Control  | <ul style="list-style-type: none"> <li>0b0: Function enabled</li> <li>0b1: Function disabled</li> </ul> | RW     |
| 21  | TMRA_2  | TimerA_2 Function Control  | <ul style="list-style-type: none"> <li>0b0: Function enabled</li> <li>0b1: Function disabled</li> </ul> | RW     |
| 20  | TMRA_1  | TimerA_1 Function Control  | <ul style="list-style-type: none"> <li>0b0: Function enabled</li> <li>0b1: Function disabled</li> </ul> | RW     |
| 19  | TMR2_4  | Timer2_4 Function Control  | <ul style="list-style-type: none"> <li>0b0: Function enabled</li> <li>0b1: Function disabled</li> </ul> | RW     |

| Bit | Name            | Function                  | Description                                                                                                 | Access |
|-----|-----------------|---------------------------|-------------------------------------------------------------------------------------------------------------|--------|
| 18  | TMR2_3          | Timer2_3 Function Control | <ul style="list-style-type: none"> <li>● 0b0: Function enabled</li> <li>● 0b1: Function disabled</li> </ul> | RW     |
| 17  | TMR2_2          | Timer2_2 Function Control | <ul style="list-style-type: none"> <li>● 0b0: Function enabled</li> <li>● 0b1: Function disabled</li> </ul> | RW     |
| 16  | TMR2_1          | Timer2_1 Function Control | <ul style="list-style-type: none"> <li>● 0b0: Function enabled</li> <li>● 0b1: Function disabled</li> </ul> | RW     |
| 15  | EMB             | EMB Function Control      | <ul style="list-style-type: none"> <li>● 0b0: Function enabled</li> <li>● 0b1: Function disabled</li> </ul> | RW     |
| 14  | TMR0_3          | Timer0_3 Function Control | <ul style="list-style-type: none"> <li>● 0b0: Function enabled</li> <li>● 0b1: Function disabled</li> </ul> | RW     |
| 13  | TMR0_2          | Timer0_2 Function Control | <ul style="list-style-type: none"> <li>● 0b0: Function enabled</li> <li>● 0b1: Function disabled</li> </ul> | RW     |
| 12  | TMR0_1          | Timer0_1 Function Control | <ul style="list-style-type: none"> <li>● 0b0: Function enabled</li> <li>● 0b1: Function disabled</li> </ul> | RW     |
| 11  | <i>Reserved</i> | <i>Reserved</i>           | <i>Reads as 1, Write as 1</i>                                                                               | -      |
| 10  | TMR4_3          | Timer4_3 Function Control | <ul style="list-style-type: none"> <li>● 0b0: Function enabled</li> <li>● 0b1: Function disabled</li> </ul> | RW     |
| 9   | TMR4_2          | Timer4_2 Function Control | <ul style="list-style-type: none"> <li>● 0b0: Function enabled</li> <li>● 0b1: Function disabled</li> </ul> | RW     |
| 8   | TMR4_1          | Timer4_1 Function Control | <ul style="list-style-type: none"> <li>● 0b0: Function enabled</li> <li>● 0b1: Function disabled</li> </ul> | RW     |
| 7   | TMR6_8          | Timer6_8 Function Control | <ul style="list-style-type: none"> <li>● 0b0: Function enabled</li> <li>● 0b1: Function disabled</li> </ul> | RW     |
| 6   | TMR6_7          | Timer6_7 Function Control | <ul style="list-style-type: none"> <li>● 0b0: Function enabled</li> <li>● 0b1: Function disabled</li> </ul> | RW     |
| 5   | TMR6_6          | Timer6_6 Function Control | <ul style="list-style-type: none"> <li>● 0b0: Function enabled</li> <li>● 0b1: Function disabled</li> </ul> | RW     |
| 4   | TMR6_5          | Timer6_5 Function Control | <ul style="list-style-type: none"> <li>● 0b0: Function enabled</li> <li>● 0b1: Function disabled</li> </ul> | RW     |
| 3   | TMR6_4          | Timer6_4 Function Control | <ul style="list-style-type: none"> <li>● 0b0: Function enabled</li> <li>● 0b1: Function disabled</li> </ul> | RW     |
| 2   | TMR6_3          | Timer6_3 Function Control | <ul style="list-style-type: none"> <li>● 0b0: Function enabled</li> <li>● 0b1: Function disabled</li> </ul> | RW     |
| 1   | TMR6_2          | Timer6_2 Function Control | <ul style="list-style-type: none"> <li>● 0b0: Function enabled</li> <li>● 0b1: Function disabled</li> </ul> | RW     |
| 0   | TMR6_1          | Timer6_1 Function Control | <ul style="list-style-type: none"> <li>● 0b0: Function enabled</li> <li>● 0b1: Function disabled</li> </ul> | RW     |

## 6.3.27 Function Clock Control Register 3 (PWC\_FCG3)

| Offset | Bit Position |         |        |        |        |        |        |        |        |        |        |          |     |     |     |     |          |     |        |        |       |       |          |      |      |        |      |      |      |    |    |   |
|--------|--------------|---------|--------|--------|--------|--------|--------|--------|--------|--------|--------|----------|-----|-----|-----|-----|----------|-----|--------|--------|-------|-------|----------|------|------|--------|------|------|------|----|----|---|
| 0x000C | 31           | 30      | 29     | 28     | 27     | 26     | 25     | 24     | 23     | 22     | 21     | 20       | 19  | 18  | 17  | 16  | 15       | 14  | 13     | 12     | 11    | 10    | 9        | 8    | 7    | 6      | 5    | 4    | 3    | 2  | 1  | 0 |
| Reset  | 0xFFFFFFFF   |         |        |        |        |        |        |        |        |        |        |          |     |     |     |     |          |     |        |        |       |       |          |      |      |        |      |      |      |    |    |   |
| Name   | Reserved     | USART10 | USART9 | USART8 | USART7 | USART6 | USART5 | USART4 | USART3 | USART2 | USART1 | Reserved | NFC | DMC | SMC | DVP | Reserved | OTS | TMR0_5 | TMR0_4 | CMP34 | CMP12 | Reserved | DAC2 | DAC1 | CMBIAS | ADC3 | ADC2 | ADC1 |    |    |   |
| Access |              | RW      | RW     | RW     | RW     | RW     | RW     | RW     | RW     | RW     | RW     |          | RW  | RW  | RW  | RW  |          | RW  | RW     | RW     | RW    | RW    |          | RW   | RW   | RW     | RW   | RW   | RW   | RW | RW |   |

| Bit   | Name     | Function                 | Description                                                                                                                                                                                                                                   | Access |
|-------|----------|--------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:30 | Reserved | Reserved                 | All bits read as 1, Write as 1                                                                                                                                                                                                                | -      |
| 29    | USART10  | USART10 Function Control | <ul style="list-style-type: none"> <li>0b0: Function enabled</li> <li>0b1: Function disabled</li> </ul>                                                                                                                                       | RW     |
| 28    | USART9   | USART9 Function Control  | <ul style="list-style-type: none"> <li>0b0: Function enabled</li> <li>0b1: Function disabled</li> </ul>                                                                                                                                       | RW     |
| 27    | USART8   | USART8 Function Control  | <ul style="list-style-type: none"> <li>0b0: Function enabled</li> <li>0b1: Function disabled</li> </ul>                                                                                                                                       | RW     |
| 26    | USART7   | USART7 Function Control  | <ul style="list-style-type: none"> <li>0b0: Function enabled</li> <li>0b1: Function disabled</li> </ul>                                                                                                                                       | RW     |
| 25    | USART6   | USART6 Function Control  | <ul style="list-style-type: none"> <li>0b0: Function enabled</li> <li>0b1: Function disabled</li> </ul>                                                                                                                                       | RW     |
| 24    | USART5   | USART5 Function Control  | <ul style="list-style-type: none"> <li>0b0: Function enabled</li> <li>0b1: Function disabled</li> </ul>                                                                                                                                       | RW     |
| 23    | USART4   | USART4 Function Control  | <ul style="list-style-type: none"> <li>0b0: Function enabled</li> <li>0b1: Function disabled</li> </ul>                                                                                                                                       | RW     |
| 22    | USART3   | USART3 Function Control  | <ul style="list-style-type: none"> <li>0b0: Function enabled</li> <li>0b1: Function disabled</li> </ul>                                                                                                                                       | RW     |
| 21    | USART2   | USART2 Function Control  | <ul style="list-style-type: none"> <li>0b0: Function enabled</li> <li>0b1: Function disabled</li> </ul>                                                                                                                                       | RW     |
| 20    | USART1   | USART1 Function Control  | <ul style="list-style-type: none"> <li>0b0: Function enabled</li> <li>0b1: Function disabled</li> </ul>                                                                                                                                       | RW     |
| 19    | Reserved | Reserved                 | Reads as 1, Write as 1                                                                                                                                                                                                                        | -      |
| 18    | NFC      | NFC Function Control     | <ul style="list-style-type: none"> <li>0b0: Enables the NAND Flash controller function of the External Memory Controller (EXMC)</li> <li>0b1: Disables the NAND Flash controller function of the External Memory Controller (EXMC)</li> </ul> | RW     |

| Bit   | Name     | Function                                    | Description                                                                                                                                                                                                                   | Access |
|-------|----------|---------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 17    | DMC      | DMC Function Control                        | <ul style="list-style-type: none"> <li>● 0b0: Enables the DMC Controller Function of External Memory Controller (EXMC).</li> <li>● 0b1: Disables the DMC Controller Function of External Memory Controller (EXMC).</li> </ul> | RW     |
| 16    | SMC      | SMC Function Control                        | <ul style="list-style-type: none"> <li>● 0b0: Enables the SMC Controller Function of External Memory Controller (EXMC).</li> <li>● 0b1: Disables the SMC Controller Function of External Memory Controller (EXMC).</li> </ul> | RW     |
| 15    | DVP      | DVP Function Control                        | <ul style="list-style-type: none"> <li>● 0b0: Function enabled</li> <li>● 0b1: Function disabled</li> </ul>                                                                                                                   | RW     |
| 14:13 | Reserved | Reserved                                    | All bits read as 1, Write as 1                                                                                                                                                                                                | -      |
| 12    | OTS      | OTS Function Control                        | <ul style="list-style-type: none"> <li>● 0b0: Function enabled</li> <li>● 0b1: Function disabled</li> </ul>                                                                                                                   | RW     |
| 11    | TMR0_5   | Timer0_5 Function Control                   | <ul style="list-style-type: none"> <li>● 0b0: Function enabled</li> <li>● 0b1: Function disabled</li> </ul>                                                                                                                   | RW     |
| 10    | TMR0_4   | Timer0_4 Function Control                   | <ul style="list-style-type: none"> <li>● 0b0: Function enabled</li> <li>● 0b1: Function disabled</li> </ul>                                                                                                                   | RW     |
| 9     | CMP34    | CMP34 Function Control                      | <ul style="list-style-type: none"> <li>● 0b0: Function enabled</li> <li>● 0b1: Function disabled</li> </ul>                                                                                                                   | RW     |
| 8     | CMP12    | CMP12 Function Control                      | <ul style="list-style-type: none"> <li>● 0b0: Function enabled</li> <li>● 0b1: Function disabled</li> </ul>                                                                                                                   | RW     |
| 7:6   | Reserved | Reserved                                    | All bits read as 1, Write as 1                                                                                                                                                                                                | -      |
| 5     | DAC2     | DAC2 Function Control                       | <ul style="list-style-type: none"> <li>● 0b0: Function enabled</li> <li>● 0b1: Function disabled</li> </ul>                                                                                                                   | RW     |
| 4     | DAC1     | DAC1 Function Control                       | <ul style="list-style-type: none"> <li>● 0b0: Function enabled</li> <li>● 0b1: Function disabled</li> </ul>                                                                                                                   | RW     |
| 3     | CMBIAS   | CMP/PGA/SH reference current source control | <ul style="list-style-type: none"> <li>● 0b0: CMP/PGA/SH reference current source enabled</li> <li>● 0b1: CMP/PGA/SH reference current source disabled</li> </ul>                                                             | RW     |
| 2     | ADC3     | ADC3 Function Control                       | <ul style="list-style-type: none"> <li>● 0b0: Function enabled</li> <li>● 0b1: Function disabled</li> </ul>                                                                                                                   | RW     |
| 1     | ADC2     | ADC2 Function Control                       | <ul style="list-style-type: none"> <li>● 0b0: Function enabled</li> <li>● 0b1: Function disabled</li> </ul>                                                                                                                   | RW     |
| 0     | ADC1     | ADC1 Function Control                       | <ul style="list-style-type: none"> <li>● 0b0: Function enabled</li> <li>● 0b1: Function disabled</li> </ul>                                                                                                                   | RW     |

## 6.3.28 PWC\_FCG0 Protection Control Register (PWC\_FCG0PC)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |          |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |      |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|------|
| 0x0010 | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15       | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |      |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |          |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |      |
| Name   | FCG0PCWE     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | Reserved |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   | PRT0 |
| Access | RW           |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |          |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   | RW   |

| Bit   | Name     | Function                          | Description                                                                                                         | Access |
|-------|----------|-----------------------------------|---------------------------------------------------------------------------------------------------------------------|--------|
| 31:16 | FCG0PCWE | PWC_FCG0PC Write Enabled          | When writing PRT0, 0xA5A5 must be written to FCG0PCWE at the same time; read always returns 0                       | RW     |
| 15:1  | Reserved | Reserved                          | -                                                                                                                   | -      |
| 0     | PRT0     | PWC_FCG0 Write Enable Control Bit | <ul style="list-style-type: none"> <li>0b0: PWC_FCG0 write disabled</li> <li>0b1: PWC_FCG0 write enabled</li> </ul> | RW     |

**Note**

Writing 0xA5A50001 to this register can set PRT0 to 1; writing 0xA5A50000 can clear PRT0 to 0; writing other values to modify PRT0 has no effect.

## 6.3.29 Function Protection Control Register (PWC\_FPRC)

| Offset | Bit Position |    |    |    |    |    |   |   |        |        |        |        |        |        |        |        |
|--------|--------------|----|----|----|----|----|---|---|--------|--------|--------|--------|--------|--------|--------|--------|
| 0xC3FE | 15           | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7      | 6      | 5      | 4      | 3      | 2      | 1      | 0      |
| Reset  | 0x0000       |    |    |    |    |    |   |   |        |        |        |        |        |        |        |        |
| Name   | PWC_FPRCWE   |    |    |    |    |    |   |   | FPRCB7 | FPRCB6 | FPRCB5 | FPRCB4 | FPRCB3 | FPRCB2 | FPRCB1 | FPRCB0 |
| Access | RW           |    |    |    |    |    |   |   | RW     | RW     | RW     | RW     | RW     | RW     | RW     | RW     |

| Bit  | Name       | Function                        | Description                                                                                                                                                                                                      | Access |
|------|------------|---------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 15:8 | PWC_FPRCWE | PWC_FPRC Register Write Enabled | When writing FPRCB7-FPRCB0, 0xA5 must be written to PWC_FPRCWE at the same time; read always returns 0                                                                                                           | RW     |
| 7    | FPRCB7     | FPRC Bit7                       | Reserved, Read as 0, Write as 0                                                                                                                                                                                  | RW     |
| 6    | FPRCB6     | FPRC Bit6                       | Reserved, Read as 0, Write as 0                                                                                                                                                                                  | RW     |
| 5    | FPRCB5     | FPRC Bit5                       | Reserved, Read as 0, Write as 0                                                                                                                                                                                  | RW     |
| 4    | FPRCB4     | FPRC Bit4                       | Reserved, Read as 0, Write as 0                                                                                                                                                                                  | RW     |
| 3    | FPRCB3     | FPRC Bit3                       | For the protection objects, please refer to the " <a href="#">Register Protection List</a> ":<br><ul style="list-style-type: none"> <li>0b0: Object write disabled</li> <li>0b1: Object write enabled</li> </ul> | RW     |
| 2    | FPRCB2     | FPRC Bit2                       | Reserved, Read as 0, Write as 0                                                                                                                                                                                  | RW     |
| 1    | FPRCB1     | FPRC Bit1                       | For the protection objects, please refer to the " <a href="#">Register Protection List</a> ":<br><ul style="list-style-type: none"> <li>0b0: Object write disabled</li> <li>0b1: Object write enabled</li> </ul> | RW     |
| 0    | FPRCB0     | FPRC Bit0                       | For the protection objects, please refer to the " <a href="#">Register Protection List</a> ":<br><ul style="list-style-type: none"> <li>0b0: Object write disabled</li> <li>0b1: Object write enabled</li> </ul> | RW     |

## 6.3.30 STOP Mode Control Register (PWC\_STPMCR)

| Offset | Bit Position |         |          |    |    |    |   |   |   |   |   |   |   |        |       |   |
|--------|--------------|---------|----------|----|----|----|---|---|---|---|---|---|---|--------|-------|---|
| 0xC00C | 15           | 14      | 13       | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2      | 1     | 0 |
| Reset  | 0x0000       |         |          |    |    |    |   |   |   |   |   |   |   |        |       |   |
| Name   | STOP         | EXBUSOE | Reserved |    |    |    |   |   |   |   |   |   |   | CKSMRC | FLNWT |   |
| Access | RW           | RW      |          |    |    |    |   |   |   |   |   |   |   | RW     | RW    |   |

| Bit  | Name     | Function                                                  | Description                                                                                                                                                                                                                                                  | Access |
|------|----------|-----------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 15   | STOP     | STOP Mode Selection Bit                                   | <ul style="list-style-type: none"> <li>0b0: STOP mode disabled</li> <li>0b1: STOP mode enabled</li> </ul>                                                                                                                                                    | RW     |
| 14   | EXBUSOE  | EXMC Address Bus and Output Control Signal Output Enabled | State control of address bus signals and output control signals of the external bus in Power-down mode and Stop mode <ul style="list-style-type: none"> <li>0b0: High impedance</li> <li>0b1: Keep the state before Power-down mode and Stop mode</li> </ul> | RW     |
| 13:2 | Reserved | Reserved                                                  | -                                                                                                                                                                                                                                                            | -      |
| 1    | CKSMRC   | Clock Switch to MRC Option                                | <ul style="list-style-type: none"> <li>0b0: Maintain the system clock and frequency division before entering STOP mode</li> <li>0b1: The system clock switches to MRC and the SCKCFGR register is initialized when waking up from STOP mode</li> </ul>       | RW     |
| 0    | FLNWT    | FLASH Stable Waiting Control                              | <ul style="list-style-type: none"> <li>0b0: Wait for FLASH to stabilize when waking up from STOP mode</li> <li>0b1: Do not wait for FLASH to stabilize when waking up from STOP mode</li> </ul>                                                              | RW     |

## 6.3.31 Backup Domain Reset Register (PWC\_VBATRSTR)

| Offset | Bit Position |   |   |   |   |   |   |   |
|--------|--------------|---|---|---|---|---|---|---|
| 0x4430 | 7            | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00         |   |   |   |   |   |   |   |
| Name   | RST          |   |   |   |   |   |   |   |
| Access | WO           |   |   |   |   |   |   |   |

| Bit | Name | Function                     | Description                           | Access |
|-----|------|------------------------------|---------------------------------------|--------|
| 7:0 | RST  | Backup Domain Reset Register | Write 0xA5 to reset the backup domain | WO     |

## 6.3.32 Backup Domain Control Register (PWC\_VBATCR)

| Offset | Bit Position |          |   |   |             |        |          |        |
|--------|--------------|----------|---|---|-------------|--------|----------|--------|
| 0x4440 | 7            | 6        | 5 | 4 | 3           | 2      | 1        | 0      |
| Reset  | 0x00         |          |   |   |             |        |          |        |
| Name   | CSDIS        | Reserved |   |   | VBATDIVMONE | RAMPDF | RAMVALID | VBTRSD |
| Access | RW           |          |   |   | RW          | RO     | RO       | RW     |

| Bit | Name            | Function                                    | Description                                                                                                                                                                                                                     | Access |
|-----|-----------------|---------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 7   | CSDIS           | VBAT Domain Current Source Inactive Control | <ul style="list-style-type: none"> <li>0b0: VBAT domain current source enabled</li> <li>0b1: VBAT domain current source is disabled. This bit can be set to 1 only when Ret SRAM, XTAL32, and XTAL32NF are not used.</li> </ul> | RW     |
| 6:4 | <i>Reserved</i> | <i>Reserved</i>                             | -                                                                                                                                                                                                                               | -      |
| 3   | VBATDIVMONE     | VBAT Voltage Divider Circuit Enable Bit     | <ul style="list-style-type: none"> <li>0b0: Voltage divider circuit disabled</li> <li>0b1: Voltage divider circuit enabled</li> </ul>                                                                                           | RW     |
| 2   | RAMPDF          | RAM Power-down Flag Bit                     | <ul style="list-style-type: none"> <li>0b0: Ret SRAM no power down</li> <li>0b1: Ret SRAM power down</li> </ul>                                                                                                                 | RO     |
| 1   | RAMVALID        | RAM Active Flag                             | <ul style="list-style-type: none"> <li>0b0: Ret SRAM is not readable or writable</li> <li>0b1: Ret SRAM is readable or writable</li> </ul>                                                                                      | RO     |
| 0   | VBTRSD          | VBAT RLDO Shutdown Control Bit              | <ul style="list-style-type: none"> <li>0b0: VBAT RLDO enabled</li> <li>0b1: VBAT RLDO disabled</li> </ul>                                                                                                                       | RW     |



## 6.3.33 Backup Domain Power-on Flag Register (PWC\_VBATPORF)

| Offset | Bit Position |   |   |   |   |   |   |          |
|--------|--------------|---|---|---|---|---|---|----------|
| 0x4448 | 7            | 6 | 5 | 4 | 3 | 2 | 1 | 0        |
| Reset  | 0x01         |   |   |   |   |   |   |          |
| Name   | Reserved     |   |   |   |   |   |   | VBATPORF |
| Access |              |   |   |   |   |   |   | RO       |

| Bit | Name     | Function                        | Description                                                                                                                                                                                                                                                                                                                                                                                                  | Access |
|-----|----------|---------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 7:1 | Reserved | Reserved                        | -                                                                                                                                                                                                                                                                                                                                                                                                            | -      |
| 0   | VBATPORF | Backup Domain Power-on Flag Bit | <ul style="list-style-type: none"> <li>0b0: No power-on event in the backup domain.</li> <li>0b1: Power-on event in the backup domain.</li> </ul> Clearing conditions: <ul style="list-style-type: none"> <li>Cleared by setting the PWC_VBATPORFCLR.VBATPORFCLR bit.</li> </ul> Setting conditions: <ul style="list-style-type: none"> <li>Set by hardware when the backup domain is powered on.</li> </ul> | RO     |

## 6.3.34 Backup Domain Power-on Flag Clear Register (PWC\_VBATPORFCLR)

| Offset | Bit Position |   |   |   |   |   |   |             |
|--------|--------------|---|---|---|---|---|---|-------------|
| 0x444C | 7            | 6 | 5 | 4 | 3 | 2 | 1 | 0           |
| Reset  | 0x00         |   |   |   |   |   |   |             |
| Name   | Reserved     |   |   |   |   |   |   | VBATPORFCLR |
| Access |              |   |   |   |   |   |   | RW          |

| Bit | Name        | Function                            | Description                                                                                                                                                                                                                                                                                                                                                                            | Access |
|-----|-------------|-------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 7:1 | Reserved    | Reserved                            | -                                                                                                                                                                                                                                                                                                                                                                                      | -      |
| 0   | VBATPORFCLR | Backup domain power-on flag cleared | Writing 1 clears the VBATPORF flag (backup domain power-on flag bit), write 0 has no effect; read always returns 0<br><br><div style="border: 1px solid black; padding: 5px;">  <b>Note</b><br/>             The write 1 action must be performed when PWC_VBATPORF.VBATPORF is 1.           </div> | RW     |

## 6.3.35 Wake-up Timer Control Register 0 (PWC\_WKTC0)

| Offset | Bit Position |   |   |   |   |   |   |   |
|--------|--------------|---|---|---|---|---|---|---|
| 0x4450 | 7            | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00         |   |   |   |   |   |   |   |
| Name   | WKTMCMP[7:0] |   |   |   |   |   |   |   |
| Access | RW           |   |   |   |   |   |   |   |

| Bit | Name         | Function                         | Description                                                     | Access |
|-----|--------------|----------------------------------|-----------------------------------------------------------------|--------|
| 7:0 | WKTMCMP[7:0] | Compare value of WKTMCMP counter | The lower 8 bits of the comparison value of the WKTMCMP counter | RW     |

## 6.3.36 Wake-up Timer Control Register 1 (PWC\_WKTC1)

| Offset | Bit Position |   |   |   |               |   |   |   |
|--------|--------------|---|---|---|---------------|---|---|---|
| 0x4454 | 7            | 6 | 5 | 4 | 3             | 2 | 1 | 0 |
| Reset  | 0x00         |   |   |   |               |   |   |   |
| Name   | Reserved     |   |   |   | WKTMCMP[11:8] |   |   |   |
| Access |              |   |   |   | RW            |   |   |   |

| Bit | Name          | Function                         | Description                                                      | Access |
|-----|---------------|----------------------------------|------------------------------------------------------------------|--------|
| 7:4 | Reserved      | Reserved                         | -                                                                | -      |
| 3:0 | WKTMCMP[11:8] | Compare value of WKTMCMP counter | The higher 4 bits of the comparison value of the WKTMCMP counter | RW     |

## 6.3.37 Wake-up Timer Control Register 2 (PWC\_WKTC2)

| Offset | Bit Position |       |   |       |          |   |   |   |
|--------|--------------|-------|---|-------|----------|---|---|---|
| 0x4458 | 7            | 6     | 5 | 4     | 3        | 2 | 1 | 0 |
| Reset  | 0x00         |       |   |       |          |   |   |   |
| Name   | WKTCE        | WKCKS |   | WKOVF | Reserved |   |   |   |
| Access | RW           | RW    |   | RW    |          |   |   |   |

| Bit | Name     | Function                      | Description                                                                                                                                                                                                                                                                                                                        | Access |
|-----|----------|-------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 7   | WKTCE    | WKTMCMP Enabled               | <ul style="list-style-type: none"> <li>0b0: WKTMCMP stop</li> <li>0b1: WKTMCMP counting</li> </ul> <div>  <b>Note</b><br/>           Clearing the WKTCE bit while WKTMCMP is counting will clear the counter and stop the counter.         </div> | RW     |
| 6:5 | WKCKS    | WKTMCMP Clock Selection       | <ul style="list-style-type: none"> <li>0b00: 64Hz clock</li> <li>0b01: XTAL32</li> <li>0b10: RTCLRC</li> <li>0b11: Setting prohibited</li> </ul>                                                                                                                                                                                   | RW     |
| 4   | WKOVF    | Timer Compare Result Flag Bit | Write 0 to clear flag, write 1 has no effect<br>Read: <ul style="list-style-type: none"> <li>0b0: The counter is inconsistent with the WKTMCMP value</li> <li>0b1: The counter matches the WKTMCMP value.</li> </ul>                                                                                                               | RW     |
| 3:0 | Reserved | Reserved                      | -                                                                                                                                                                                                                                                                                                                                  | -      |

## 6.3.38 VBAT Backup Register n (PWC\_BRKn) (n=0-127)

Offset: 0x4600+0x4\*n

| Offset | Bit Position |   |   |   |   |   |   |   |
|--------|--------------|---|---|---|---|---|---|---|
| -      | 7            | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0xFF         |   |   |   |   |   |   |   |
| Name   | BRKnB        |   |   |   |   |   |   |   |
| Access | RW           |   |   |   |   |   |   |   |

| Bit | Name  | Function        | Description     | Access |
|-----|-------|-----------------|-----------------|--------|
| 7:0 | BRKnB | Backup Register | Backup Register | RW     |

## 7 Initialization Configuration (ICG)

### 7.1 Introduction

After the chip's reset is released, the hardware circuit reads data from FLASH addresses 0x00000400-0x0000046F and loads it into the Initialization Configuration (ICG) register. Addresses 0x00000408-0x0000040B, 0x00000410-0x0000041F, and 0x00000438-0x0000045F are reserved; please write all 1s to ensure normal chip action. When FLASH boot swap is enabled and OTP is not enabled, this region is located in FLASH block 1, sector 0; Otherwise, this region is in FLASH block 0, sector 0. Users can modify the ICG register by programming or erasing sector 0. Addresses 0x00000420-0x00000437 are the data security protection enable region; for details, please refer to the "Functional Description-FLASH Programming and Erase Operations-Data Security Protection" chapter in "Embedded Flash Memory (EFM)". The register reset value is determined by FLASH data.

### 7.2 Register Description

#### 7.2.1 Register List

Base address: 0x00000400

Table 7-1 Initialization Configuration (ICG) Register Offset

| Offset | Register | Description                              | Width |
|--------|----------|------------------------------------------|-------|
| 0x000  | ICG0     | Initialization Configuration Register 0  | 32    |
| 0x004  | ICG1     | Initialization Configuration Register 1  | 32    |
| 0x00C  | ICG3     | Initialization Configuration Register 3  | 32    |
| 0x060  | ICG8A    | Initialization Configuration Register 8A | 32    |
| 0x064  | ICG8B    | Initialization Configuration Register 8B | 32    |
| 0x068  | ICG8C    | Initialization Configuration Register 8C | 32    |

## 7.2.2 Initialization Configuration Register 0 (ICG0)

| Offset | Bit Position |    |           |    |         |    |    |    |        |    |    |    |         |    |        |    |         |    |          |    |           |    |          |   |   |   |         |   |   |   |          |   |         |  |          |  |
|--------|--------------|----|-----------|----|---------|----|----|----|--------|----|----|----|---------|----|--------|----|---------|----|----------|----|-----------|----|----------|---|---|---|---------|---|---|---|----------|---|---------|--|----------|--|
| 0x000  | 31           | 30 | 29        | 28 | 27      | 26 | 25 | 24 | 23     | 22 | 21 | 20 | 19      | 18 | 17     | 16 | 15      | 14 | 13       | 12 | 11        | 10 | 9        | 8 | 7 | 6 | 5       | 4 | 3 | 2 | 1        | 0 |         |  |          |  |
| Reset  | 0xFFFFFFFF   |    |           |    |         |    |    |    |        |    |    |    |         |    |        |    |         |    |          |    |           |    |          |   |   |   |         |   |   |   |          |   |         |  |          |  |
| Name   | Reserved     |    | WDTSLPOFF |    | WDTWDPT |    |    |    | WDTCKS |    |    |    | WDTPERI |    | WDTITS |    | WDTAITS |    | Reserved |    | SWDTSLOFF |    | SWDTWDPT |   |   |   | SWDTCKS |   |   |   | SWDTPERI |   | SWDTITS |  | SWDTAITS |  |
| Access |              |    | RO        |    | RO      |    |    |    | RO     |    |    |    | RO      |    | RO     |    | RO      |    |          |    | RO        |    | RO       |   |   |   | RO      |   |   |   | RO       |   | RO      |  | RO       |  |

| Bit   | Name      | Function                                            | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | Access |
|-------|-----------|-----------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:29 | Reserved  | Reserved                                            | All bits read as 1, Write as 1                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | -      |
| 28    | WDTSLPOFF | WDT Counting Stops in Sleep Mode                    | <ul style="list-style-type: none"> <li>0b0: WDT counting does not stop in Sleep mode</li> <li>0b1: WDT counting stops in Sleep mode</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                             | RO     |
| 27:24 | WDTWDPT   | Allowed Interval for Refreshing the WDT Count Value | Allowed interval for refreshing the WDT count value: <ul style="list-style-type: none"> <li>0b0000: 0%-100%</li> <li>0b0001: 0%-25%</li> <li>0b0010: 25%-50%</li> <li>0b0011: 0%-50%</li> <li>0b0100: 50%-75%</li> <li>0b0101: 0%-25%, 50%-75%</li> <li>0b0110: 25%-75%</li> <li>0b0111: 0%-75%</li> <li>0b1000: 75%-100%</li> <li>0b1001: 0%-25%, 75%-100%</li> <li>0b1010: 25%-50%, 75%-100%</li> <li>0b1011: 0%-50%, 75%-100%</li> <li>0b1100: 50%-100%</li> <li>0b1101: 0%-25%, 50%-100%</li> <li>0b1110: 25%-100%</li> <li>0b1111: 0%-100%</li> </ul> | RO     |
| 23:20 | WDTCKS    | WDT Count Clock                                     | <ul style="list-style-type: none"> <li>0b0010: PCLK3/4</li> <li>0b0110: PCLK3/64</li> <li>0b0111: PCLK3/128</li> <li>0b1000: PCLK3/256</li> <li>0b1001: PCLK3/512</li> <li>0b1010: PCLK3/1024</li> <li>0b1011: PCLK3/2048</li> <li>0b1101: PCLK3/8192</li> <li>Others: Reserved</li> </ul>                                                                                                                                                                                                                                                                 | RO     |

| Bit   | Name            | Function                                             | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | Access |
|-------|-----------------|------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 19:18 | WDTPERI         | WDT Count Overflow Period                            | <ul style="list-style-type: none"> <li>● 0b00: 256 periods</li> <li>● 0b01: 4096 periods</li> <li>● 0b10: 16384 periods</li> <li>● 0b11: 65536 periods</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                  | RO     |
| 17    | WDTITS          | WDT Interrupt Selection                              | <ul style="list-style-type: none"> <li>● 0b0: Interrupt request</li> <li>● 0b1: Reset request</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | RO     |
| 16    | WDTAUTS         | WDT Auto Start                                       | <ul style="list-style-type: none"> <li>● 0b0: After reset, WDT auto-starts (hardware start)</li> <li>● 0b1: After reset, WDT is in stopped state</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                        | RO     |
| 15:13 | <i>Reserved</i> | <i>Reserved</i>                                      | <i>All bits read as 1, Write as 1</i>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | -      |
| 12    | SWDTSLOPF       | SWDT Counting Stops in Sleep/Stop Mode               | <ul style="list-style-type: none"> <li>● 0b0: SWDT counting does not stop in Sleep mode/Stop mode</li> <li>● 0b1: SWDT counting stops in Sleep mode/Stop mode</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                           | RO     |
| 11:8  | SWDWDPT         | Allowed Interval for Refreshing the SWDT Count Value | <p>Allowed interval for refreshing the SWDT count value:</p> <ul style="list-style-type: none"> <li>● 0b0000: 0%-100%</li> <li>● 0b0001: 0%-25%</li> <li>● 0b0010: 25%-50%</li> <li>● 0b0011: 0%-50%</li> <li>● 0b0100: 50%-75%</li> <li>● 0b0101: 0%-25%, 50%-75%</li> <li>● 0b0110: 25%-75%</li> <li>● 0b0111: 0%-75%</li> <li>● 0b1000: 75%-100%</li> <li>● 0b1001: 0%-25%, 75%-100%</li> <li>● 0b1010: 25%-50%, 75%-100%</li> <li>● 0b1011: 0%-50%, 75%-100%</li> <li>● 0b1100: 50%-100%</li> <li>● 0b1101: 0%-25%, 50%-100%</li> <li>● 0b1110: 25%-100%</li> <li>● 0b1111: 0%-100%</li> </ul> | RO     |
| 7:4   | SWDTCKS         | SWDT Counter-Clock                                   | <ul style="list-style-type: none"> <li>● 0b0000: SWDTCLK</li> <li>● 0b0100: SWDTCLK/16</li> <li>● 0b0101: SWDTCLK/32</li> <li>● 0b0110: SWDTCLK/64</li> <li>● 0b0111: SWDTCLK/128</li> <li>● 0b1000: SWDTCLK/256</li> <li>● 0b1011: SWDTCLK/2048</li> <li>● Others: Reserved</li> </ul>                                                                                                                                                                                                                                                                                                            | RO     |
| 3:2   | SWDTPERI        | SWDT Count Overflow Cycle                            | <ul style="list-style-type: none"> <li>● 0b00: 256 periods</li> <li>● 0b01: 4096 periods</li> <li>● 0b10: 16384 periods</li> <li>● 0b11: 65536 periods</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                  | RO     |

| Bit | Name    | Function                 | Description                                                                                                                                                | Access |
|-----|---------|--------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 1   | SWDTITS | SWDT Interrupt Selection | <ul style="list-style-type: none"><li>● 0b0: Interrupt request</li><li>● 0b1: Reset request</li></ul>                                                      | RO     |
| 0   | SWDTAUS | SWDT Auto Start          | <ul style="list-style-type: none"><li>● 0b0: After reset, SWDT auto-starts (hardware start)</li><li>● 0b1: After reset, SWDT is in stopped state</li></ul> | RO     |

## 7.2.3 Initialization Configuration Register 1 (ICG1)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |        |    |         |    |          |    |    |    |   |   |         |   |          |   |   |   |   |   |           |  |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|--------|----|---------|----|----------|----|----|----|---|---|---------|---|----------|---|---|---|---|---|-----------|--|
| 0x004  | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17     | 16 | 15      | 14 | 13       | 12 | 11 | 10 | 9 | 8 | 7       | 6 | 5        | 4 | 3 | 2 | 1 | 0 |           |  |
| Reset  | 0xFFFFFFFF   |    |    |    |    |    |    |    |    |    |    |    |    |    |        |    |         |    |          |    |    |    |   |   |         |   |          |   |   |   |   |   |           |  |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    | BORDIS |    | BOR_LEV |    | Reserved |    |    |    |   |   | HRCSTOP |   | Reserved |   |   |   |   |   | HRCFRESEL |  |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    | RO     |    | RO      |    |          |    |    |    |   |   | RO      |   |          |   |   |   |   |   | RO        |  |

| Bit   | Name       | Function                        | Description                                                                                                                       | Access |
|-------|------------|---------------------------------|-----------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:19 | Reserved   | Reserved                        | All bits read as 1, Write as 1                                                                                                    | -      |
| 18    | BORDIS     | BOR action selection            | <ul style="list-style-type: none"> <li>0b0: Enable BOR action after reset</li> <li>0b1: Disable BOR action after reset</li> </ul> | RO     |
| 17:16 | BOR_LEV    | BOR threshold voltage selection | <ul style="list-style-type: none"> <li>0b00: 2.00V</li> <li>0b01: 2.10V</li> <li>0b10: 2.20V</li> <li>0b11: 2.40V</li> </ul>      | RO     |
| 15:9  | Reserved   | Reserved                        | All bits read as 1, Write as 1                                                                                                    | -      |
| 8     | HRCSTOP    | HRC oscillation stop bit        | <ul style="list-style-type: none"> <li>0b0: HRC oscillation</li> <li>0b1: HRC stops</li> </ul>                                    | RO     |
| 7:1   | Reserved   | Reserved                        | All bits read as 1, Write as 1                                                                                                    | -      |
| 0     | HRCFREQSEL | HRC Frequency Selection         | <ul style="list-style-type: none"> <li>0b0: 20MHz</li> <li>0b1: 16MHz</li> </ul>                                                  | RO     |



## 7.2.4 Initialization Configuration Register 3 (ICG3)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |         |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0x00C  | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15      | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0xFFFFFFFF   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |         |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | DBUSPRT |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RO      |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit   | Name     | Function                                            | Description                                                                                                                                     | Access |
|-------|----------|-----------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:16 | Reserved | Reserved                                            | All bits read as 1, Write as 1                                                                                                                  | -      |
| 15:0  | DBUSPRT  | D-BUS Read Protection Function for 128KBytes Region | When DBUSPRT[15:0]=0x4450, D-BUS read protection is enabled for the 0x00000000-0x0001FFFF region; otherwise, D-BUS read protection is disabled. | RO     |

## 7.2.5 Initialization Configuration Register 8x (ICG8x) (x=A/B/C)



### Note

Program the same data into FLASH addresses 0x00000460-0x00000463, 0x00000464-0x00000467, and 0x00000468-0x0000046B. ICG8A/8B/8C require the use of three registers together, with the same value.

Offset: 0x060 (ICG8A), 0x064 (ICG8B), 0x068 (ICG8C)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |            |   |          |  |            |  |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|------------|---|----------|--|------------|--|
| -      | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1          | 0 |          |  |            |  |
| Reset  | 0xFFFFFFFF   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |            |   |          |  |            |  |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   | ERMUOUTDIS |   | Reserved |  | ERMUOUTSEL |  |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   | RO         |   |          |  | RO         |  |

| Bit  | Name       | Function                            | Description                                                                                                                                                                                                               | Access |
|------|------------|-------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:5 | Reserved   | Reserved                            | All bits read as 1, Write as 1                                                                                                                                                                                            | -      |
| 4    | ERMUOUTDIS | ERMU Function Port Output Control   | <ul style="list-style-type: none"> <li>0b0: Output is enabled</li> <li>0b1: Output is disabled</li> </ul>                                                                                                                 | RO     |
| 3:2  | Reserved   | Reserved                            | All bits read as 1, Write as 1                                                                                                                                                                                            | -      |
| 1:0  | ERMUOUTSEL | ERMU Function Port Output Selection | <ul style="list-style-type: none"> <li>0b00: ERMU_EOUT0M is output by PA5</li> <li>0b01: ERMU_EOUT0M is output by PH5</li> <li>0b10: ERMU_EOUT0M is output by PH13</li> <li>0b11: ERMU_EOUT0M is output by PC1</li> </ul> | RO     |

## 8 Embedded Flash Memory (EFM)

### 8.1 Introduction

The Flash interface accesses Flash through the ICODE, DCODE, and MCODE buses. The interface can perform programming, erasing and full erasing operations on Flash. Besides, it can accelerate code execution through command prefetching and caching mechanisms.

### 8.2 Features

- Two independent Flash blocks form a dual bank, enabling BGO (BackGround Operation) functionality
- 134KBytes OTP space
- ICODE Bus 16Bytes prefetching
- Two independent cache areas: ICODE bus cache space 4KBytes (256x128); DCODE bus cache space 1KBytes (64x128)
- Supports boot swap function
- Support data security protection
- Features ECC (Error Checking and Correcting), which can correct one-bit errors in 128-bit data and 9-bit ECC, detect one-bit errors in 20-bit address-related information, and detect two-bit errors in 128-bit data with 9-bit ECC and 20-bit address-related information.

## 8.3 Functional Description

### 8.3.1 Embedded Flash

Flash has the following main features:

- Capacity: Up to 2MBytes, consisting of two 1MByte Flash, with a total of 256 sectors, each 8KBytes. Sectors 0-15 in Block 0 can be configured as OTP areas.
- The OTP (One Time Program) area is 134KBytes in total, of which 128KBytes are configured at block 0 address 0x00000000-0x0001FFFF, 6KBytes are configured at address 0x03000000-0x030017FF. Addresses 0x03001800-0x03001F6F are for OTP block latches, 0x03001F70-0x03001FEF are reserved for OTP, and 0x03001FF0-0x03001FFF are for OTP enable addresses.
- 128-bit wide data read: 128-bit wide read cache buffer accelerates code execution.
- The programming unit is 16Bytes. Erase unit: 8KBytes.

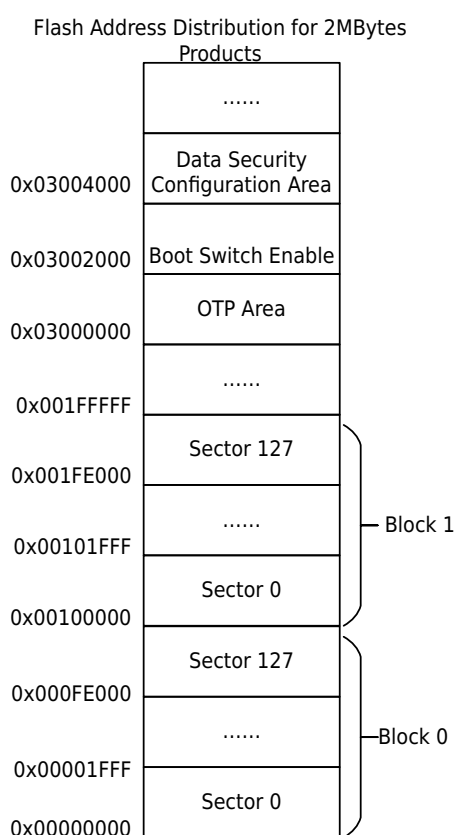


**Figure 8-1 Flash Address Distribution**

The OTP area, boot swap enable address, and data security configuration area each have a dedicated sector for programming, erasing, but full erase is disabled. These three sectors physically belong to Flash block 0, and their erase/write flags follow Flash block 0, but their data remains unchanged during a full erase of Flash block 0.

The OTP region spans from 0x03000000 to 0x03001FFF, the boot swap enable function is configured at addresses 0x03002000 to 0x0300200F, and the data security configuration area is located from 0x03004000 to 0x0300400F.

## 8.3.2 Read Interface

### 8.3.2.1 CPU Clock and Flash Read Time Relationship

To correctly read Flash data, users need to set the wait period count (EFM\_FRMC.FLWT) based on the CPU operation frequency.

After system reset, the CPU clock source is MRC (8MHz), and the Flash read wait period is 0. Users are recommended to follow these steps to modify the CPU frequency and Flash read wait period bits: The number of waiting periods is shown in the table below.

**Table 8-1 Compare Table of CPU Clock Frequency (HCLK) and Flash Read Waiting Cycles**

| CPU Clock Frequency (HCLK) | Read Waiting Cycle Setting |
|----------------------------|----------------------------|
| 200MHz<HCLK≤240MHz         | EFM_FRMC.FLWT=0b0101       |
| 160MHz<HCLK≤200MHz         | EFM_FRMC.FLWT=0b0100       |
| 120MHz<HCLK≤160MHz         | EFM_FRMC.FLWT=0b0011       |
| 80MHz<HCLK≤120MHz          | EFM_FRMC.FLWT=0b0010       |
| 40MHz<HCLK≤80MHz           | EFM_FRMC.FLWT=0b0001       |
| HCLK≤40MHz                 | EFM_FRMC.FLWT=0b0000       |

#### CPU Frequency Increase Steps:

- Step1.** Write the new read wait period setting to the EFM\_FRMC.FLWT register.
- Step2.** Read the EFM\_FRMC register to check if the new wait period setting is successful.
- Step3.** Increase the CPU clock frequency by setting the system clock source switch register CMU\_CKSWR.CKSW or the system clock configuration register CMU\_SCFGR.HCLKS.
- Step4.** Read the CMU\_CKSWR or CMU\_SCFGR register to check if the new setting is successful.

#### CPU Frequency Reduction Steps:

- Step1.** Decrease the CPU clock frequency by setting the system clock source switch register CMU\_CKSWR.CKSW or the system clock configuration register CMU\_SCFGR.HCLKS.
- Step2.** Read the register CMU\_CKSWR or CMU\_SCFGR to check if the new setting is successful.
- Step3.** Write the new read wait period setting to the EFM\_FRMC.FLWT register.
- Step4.** Read the EFM\_FRMC register to check if the new wait period setting is successful.

### 8.3.2.2 Normal Read

By default, Flash is in Normal read mode, which is the most commonly used by customers.

### 8.3.2.3 Blank Read

After a Flash sector is erased, all Flash data, including ECC data, is checked to be "1".

Blank read mode setup steps:

1. Disable the Flash register write protection. (Write 0x0123 to EFM\_FAPRT, then write 0x3210)
  2. Set Blank read mode. (EFM\_FRMC2.BRM0/1=1)
  3. Read the Flash address to be checked.
  4. Check if the read data contains any "0" flags (EFM\_FSR.BRMER0/1).
- If the read data (including ECC data) is not all "1" in the Blank state, EFM\_FSR.BRMER0/1=1.
5. Write 1 to the corresponding flag clear bit in the EFM\_FSCLR register to clear the status bit.

#### 6. Set back to Normal read mode.(EFM\_FRMC2.BRM0/1=0)

The above procedure should be executed on a different Flash block from the check address, or on other memory (e.g., RAM).

### 8.3.3 Flash Read Acceleration Cache

Each Flash read operation is a 128-bit read, with data sent to the CPU and also stored in a buffer register. These 128 bits of data can be 4 rows of 32-bit instructions or 8 rows of 16-bit instructions, depending on the program burned into Flash.

To enable fast reading of Flash data, the Flash controller is configured with a read acceleration cache that optimizes read wait periods. To maximize processor performance, this accelerator stores ICODE and DCODE bus access data in cache memory, improving program execution speed.

The system provides 5KBytes of space as cache memory to actively reduce the time loss caused by instruction jump. Enable the cache function by setting the EFM\_FRMC.ICACHE/DCACHE bit to 1. Whenever an instruction or data miss occurs (i.e., the requested instruction or data is not present in the currently used instruction line or cache memory), the system copies the newly read data line (128 bits) into cache memory. If the instruction or data requested by the CPU is already in the cache, it can be obtained immediately without any delay. When the cache memory is full, the LRU (Least Recently Used) strategy is used to determine which data in the cache memory to replace.

When the CPU reads instructions or data, the Flash address is buffered. When there is a cache hit, the number of Flash read periods changes, as shown in the table below.

Table 8-2 Actual Flash Read Cycle Count

| EFM_FRMC.<br>FLWT Setting | Cache Disable |             | Cache Enable        |                               |
|---------------------------|---------------|-------------|---------------------|-------------------------------|
|                           | Buffer Hit    | Buffer Miss | Buffer or Cache Hit | Miss in Both Buffer and Cache |
| 0                         | 1             | 1           | 1                   | 1                             |
| 1                         | 1             | 2           | 1                   | 2                             |
| 2                         | 1             | 3           | 1                   | 3                             |
| N(N>2)                    | 1             | N+1         | 1                   | N+1                           |

### 8.3.4 Flash Programming and Erasing Operations

#### 8.3.4.1 Overview of Programming and Erase Operations

Flash supports programming, sector erase, and full erase operations.

Flash programming, sector/full erase address must be 4-byte aligned (last bit addresses are: 0x0, 0x4, 0x8, 0xC). The programming unit is 16Bytes. The sector erase unit is 8KBytes. Full erase can be either a single Flash block or two Flash blocks according to register settings. Flash programming methods are divided into three types: single programming without read-back mode, single programming with read-back mode, and continuous programming mode. During Flash programming and erasing, if EFM\_FWMC.BUSHLCTL=0 is set, the bus is occupied until the write/erase is completed; if EFM\_FWMC.BUSHLCTL=1, the bus is released, the bus can continue to access another Flash address. Before Flash programming and erasing, please disable cache enable and prefetch. The following describes the setting steps for programming and erasing operations.

### 8.3.4.2 Unlock EFM\_KEY1 Register

After reset, the Flash programming and erasing mode register (EFM\_FWMC) is in write-prohibited status. It is necessary to first disable the Flash access protection register (EFM\_FAPRT), and then disable the protection of EFM\_KEY1. The following steps are used to disable the above protection:

**Step1.** Remove the Flash register access write protection (EFM\_FAPRT first writes 0x0123, then writes 0x3210)

**Step2.** Unlock EFM\_KEY1 ( write 0x01234567 to EFM\_KEY1 first, then write 0xFEDCBA98 )

If an incorrect sequence is written to EFM\_KEY1, a bus error will occur, and it will self-lock. The EFM\_FWMC register will be write-prohibited until a reset occurs.

After correctly writing the set value to the EFM\_FWMC register, if you need to return to the EFM\_FWMC write-prohibited state, you can write 1 to EFM\_FWMC.KEY1LOCK.

### 8.3.4.3 Write Protection Function

Each Flash sector has 1 write protection bit, located in the write protection register EFM\_F0/1NWPRTx (x=0-3). Each write protection register has 1 write protection lock bit (WLOCK[7:0]). Once the write protection lock bit is set to 1, the write protection register bit can only be set from write-permitted to write-prohibited. When a Flash sector is set to write-prohibited, programming and erasing operations to that sector address will not occur, and an error flag (EFM\_FSR.PRTWERR0/1=1) will be generated.

In full erase mode, if there is one or more sectors in the Flash block set to write-prohibited, full erase will not occur, and an error flag (EFM\_FSR.PRTWERR0/1=1) will be generated.

### 8.3.4.4 Single Programming Without Read-back Function

The steps for single programming without read-back mode are as follows:

**Step1.** Disable the Flash register write protection. (Write 0x0123 to EFM\_FAPRT first, then write 0x3210)

**Step2.** Unlock the EFM\_KEY1 lock. (Write 0x01234567 to EFM\_KEY1 first, then write 0xFEDCBA98)

**Step3.** Set the single programming mode. (EFM\_FWMC.PEMOD=0b001)

**Step4.** Disable write protection. (EFM\_F0/1NWPRTx (x=0-3) corresponding bit is set to 1)

**Step5.** Write 128 of data to the programming address, writing 32-bit data consecutively to programming addresses with last bit addresses 0x0, 0x4, 0x8, 0xC (totaling 128 bits).

**Step6.** Wait for the Flash operation to end and for it to be in an idle state. (EFM\_FSR.OPTEND0/1=1, EFM\_FSR.RDY0/1=1)

**Step7.** Read the programming address value and check if it matches the written value:  
If they match, programming is successful. If they do not match, the Flash address has been corrupted and should be permanently discarded.

**Step8.** Clear the end of programming flag. (EFM\_FSR.OPTEND0/1)

Performing a single programming write operation on a locked OTP address without read-back will fail, and the flag bit EFM\_FSR.OTPWERR0 will be set.

If the programming addresses with last bit addresses 0x0, 0x4, 0x8, 0xC are not written consecutively, the programming action will not be executed, and the flag bit EFM\_FSR.PGSZERR0/1 will be set.

### 8.3.4.5 Single Programming With Read-back Function

Single programming with read-back mode refers to the hardware automatically reading the programming address data after programming ends and comparing it with the written data to verify correctness by checking the flag bit EFM\_FSR.MISMTCH0/1.

The steps for setting the read mode for single programming are as follows:

- Step1.** Disable the Flash register write protection. (Write 0x0123 to EFM\_FAPRT, then write 0x3210)
- Step2.** Unlock the EFM\_KEY1 lock. (Write 0x01234567 to EFM\_KEY1 first, then write 0xFEDCBA98)
- Step3.** Set single programming with read-back mode. (EFM\_FWMC.PEMOD=0b010)
- Step4.** Disable write protection. (EFM\_F0/1NWPRTx (x=0-3) corresponding bit is set to 1)
- Step5.** Write 128 of data to the programming address, writing 32-bit data consecutively to programming addresses with last bit addresses 0x0, 0x4, 0x8, 0xC (totaling 128 bits).
- Step6.** Wait for the Flash operation to end and for it to be in an idle state. (EFM\_FSR.OPTEND0/1=1, EFM\_FSR.RDY0/1=1)
- Step7.** Check the programming self-read result flag bit (EFM\_FSR.MISMTCH0/1) :  
If 0, programming is successful. If 1, the Flash address has been corrupted and should be permanently discarded.
- Step8.** Clear the programming end flag bit (EFM\_FSR.OPTEND0/1).

Performing a single programming read-and-write operation on a locked OTP address will fail, and the flag bit EFM\_FSR.OTPWERR0 will be set.

If the programming addresses with last bit addresses 0x0, 0x4, 0x8, 0xC are not written consecutively, the programming action will not be executed, and the flag bit EFM\_FSR.PGSZERR0/1 will be set.

#### 8.3.4.6 Continuous Programming Function

When programming Flash addresses consecutively, it is recommended to use continuous programming mode. Continuous programming mode can save over 50% of the time compared to single programming mode. In continuous programming mode, the interval between Flash address write commands should not exceed 16μs. The steps for setting up continuous programming operations are as follows:

- Step1.** Disable the Flash register write protection. (Write 0x0123 to EFM\_FAPRT, then write 0x3210)
- Step2.** Unlock the EFM\_KEY1. (Write 0x01234567 to EFM\_KEY1 first, then write 0xFEDCBA98)
- Step3.** Set continuous programming mode. (EFM\_FWMC.PEMOD=0b011)
- Step4.** Disable write protection. (EFM\_F0/1NWPRTx (x=0-3) corresponding bit is set to 1)
- Step5.** Write 128 of data to the programming address, writing 32-bit data consecutively to programming addresses with last bit addresses 0x0, 0x4, 0x8, 0xC (totaling 128 bits).  
The programming address should be in a different Flash block from the executing program, or the program should run in a region outside of Flash.
- Step6.** Wait for the operation end flag bit (EFM\_FSR.OPTEND0/1) to be set.
- Step7.** Clear the operation end flag bit until EFM\_FSR.OPTEND0/1 reads 0.
- Step8.** Repeat steps 5-7 until all data is written.
- Step9.** Change continuous programming mode to non-continuous programming mode. (EFM\_FWMC.PEMOD=0b000)
- Step10.** Wait for Flash to be in an idle state. (EFM\_FSR.RDY0/1=1)
- Step11.** Read the programming address and check if it matches the written value.  
If they match, programming is successful. If they do not match, the Flash address has been corrupted and should be permanently discarded.

Performing a continuous programming write operation on a locked OTP address will fail, and the flag bit EFM\_FSR.OTPWERR0 will be set.

If the programming addresses with last bit addresses 0x0, 0x4, 0x8, 0xC are not written consecutively, the programming action will not be executed, and the flag bit EFM\_FSR.PGSZERR0/1 will be set.



### 8.3.4.7 Sector Erase Function

After performing a sector erase operation on Flash, the data in the sector (8KBytes of space) is refreshed to all 1s.

The steps for setting the sector erase operation are as follows:

- Step1.** Remove the Flash register write protection (EFM\_FAPRT first writes 0x0123, then writes 0x3210).
- Step2.** Unlock the EFM\_KEY1 lock. (Write 0x01234567 to EFM\_KEY1 first, then write 0xFEDCBA98)
- Step3.** Set sector erase mode (EFM\_FWMC.PEMOD=0b100).
- Step4.** Disable write protection. (EFM\_F0/1NWPRTx (x=0-3) corresponding bit is set to 1)
- Step5.** Write any 32-bit value to any address within the sector to be erased (address must be 4-aligned).
- Step6.** Wait for the Flash operation to end and for it to be in an idle state. (EFM\_FSR.OPTEND0/1=1, EFM\_FSR.RDY0/1=1)
- Step7.** Clear the erase end flag bit. (EFM\_FSR.OPTEND0/1)

Performing an erase operation on a locked OTP address will fail, the OTP area data will remain, and the flag bit EFM\_FSR.OTPWERR0 will be set.

### 8.3.4.8 Full Erase Function

EFM provides two erase methods: single block Flash full erase and simultaneous full erase of both blocks.

The steps for setting the full erase operation are as follows:

- Step1.** Remove the Flash register write protection (EFM\_FAPRT first writes 0x0123, then writes 0x3210).
- Step2.** Unlock the EFM\_KEY1 lock. (Write 0x01234567 to EFM\_KEY1 first, then write 0xFEDCBA98)
- Step3.** For single block Flash full erase, set EFM\_FWMC.PEMOD=0b101.
- Step4.** For simultaneous full erase of both blocks, set EFM\_FWMC.PEMOD[2:0]=0b110.
- Step5.** Disable write protection. (EFM\_F0/1NWPRTx (x=0-3) Flash block all bits are set to 1)
- Step6.** For single block Flash full erase: Write any 32-bit value to any address within the Flash block to be erased (address must be 4-aligned).
- Step7.** For simultaneous full erase of both blocks, write any 32-bit value to any address within Flash block 0 or block 1 (address must be 4-aligned).  
The full erase action will be automatically synchronized to the other block of Flash.
- Step8.** Wait for the Flash operation to end and for it to be in an idle state. (EFM\_FSR.OPTEND0/1=1, EFM\_FSR.RDY0/1=1)
- Step9.** Clear the erase end flag bit. (EFM\_FSR.OPTEND0/1)

After OTP is enabled, performing a single block full erase write operation on Flash block 0 address will fail, block 0 data will remain, and the flag bit EFM\_FSR.OTPWERR0 will be set; performing a simultaneous full erase write operation on Flash block 0 address will result in block 0 data remaining and block 1 data being erased, and the flag bit EFM\_FSR.OTPWERR0 will be set.

### 8.3.4.9 Data Security Protection

This product provides four protection levels for Flash data to prevent untrusted users from reading and tampering with Flash data through debug interfaces (JTAG and SWD), ISP interfaces (In System Program), and test interfaces.

### Protection Level 0: No Protection

Debug interfaces, ISP interfaces and test interfaces can access (read and rewrite) MCU resources, including Flash data.

### Protection Level 1

After the data 0xAF180402 is programmed into the flash address 0x00000430-0x00000433 and the 96-bit password is programmed into the address 0x03004000-0x0300400B, protection level 1 is enabled.

After enabling protection level 1:

- Debug interfaces, ISP interfaces and test interfaces cannot access MCU resources.
- User programs cannot program or perform sector erases in sector 0.
- The data at addresses 0x03004000-0x0300400B cannot be read.

After activating protection level 1, it can be reset to protection level 0 through password authentication or full erase. If OTP is enabled, full erase is disabled. Please consult the sales window for password authentication and full erase methods.

### Protection Level 2

Protection level 2 is enabled by programming data 0xA85173AE into Flash addresses 0x00000434-0x00000437.

After enabling protection level 2:

- Debug interfaces, ISP interfaces and test interfaces cannot access MCU resources.
- User programs cannot program or perform sector erases in sector 0.

After activating protection level 2, it can be reset to protection level 0 through full erase. If OTP is enabled, full erase is disabled.

### Protection Level 3

Protection level 3 is enabled by programming data 0x42545048 into Flash addresses 0x00000420-0x00000423, 0x00000424-0x00000427, 0x00000428-0x0000042B.

After enabling protection level 3:

- Debug interfaces, ISP interfaces and test interfaces cannot rewrite Flash data.
- User programs cannot program or perform sector erases in sector 0.

After activating protection level 3, it can be reset to protection level 0 through full erase. If OTP is enabled, full erase is disabled.

Protection levels 1, 2, and 3 can be enabled individually or simultaneously. When enabled simultaneously, the protection measures are cumulative.

#### 8.3.4.10 D-BUS Read Protection Function

D-BUS read protection is provided for the address space 0x00000000-0x0001FFFF (128KBytes). To enable D-BUS protection, set the ICG3.DBUSPRT to 0x4450. After a reset, the function is enabled. When the CPU-PC pointer is within this 128KB space (i.e., the CPU is executing a program in this region) : The CPU can access normally. When the CPU-PC pointer is outside this 128KB space (i.e., the CPU is executing a program outside this region) : D-BUS read access to the 128KB space by the CPU is disabled, and a bus error is returned. D-BUS is the CPU's data access bus, including operand access, stack data access, general data access, etc.

### 8.3.4.11 BGO Function

When the target addresses for Flash programming and erasing are not in the same Flash block as the address of the executing program, or when the program is running in RAM, EFM\_FWMC.BUSHLCTL can be set to 1 (bus released during programming and erasing) to allow Flash programming or erasing to occur while the program is executing.

The following are the steps for setting BGO functions:

- Step1.** Disable the Flash register write protection. (Write 0x0123 to EFM\_FAPRT, then write 0x3210)
- Step2.** Unlock the EFM\_KEY1 lock. (Write 0x01234567 to EFM\_KEY1 first, then write 0xFEDCBA98)
- Step3.** Set the programming, sector erase or full erase mode. (EFM\_FWMC.PE-MOD=0b001/0b010/0b011/0b100/0b101)
- Step4.** Disable write protection. (Set the corresponding sector bits to 1 in EFM\_F0/1NWPRTx (x=0-3) for the target address)
- Step5.** Set the bus release during programming, sector erase or full erase (EFM\_FWMC.BUSHLCTL=1)
- Step6.** 128 bits of data are written to the programming address, and 32 bits of data (a total of 128 bits) are written to the programming addresses whose last addresses are 0x0, 0x4, 0x8, and 0xC respectively.  
(Programming address and executing program should be in different Flash blocks)  
If the executing program is in RAM, both blocks of Flash can be programmed or (full) erased simultaneously.
- Step7.** Wait for the erase/write end flag bit to be set. (EFM\_FSR.OPTEND0/1=1)

### 8.3.5 One-Time Programmable Byte (OTP)

This MCU provides a maximum of 134KBytes of OTP area, and the address distribution is as follows.

Divided into 16x8KBytes, 1x2KBytes, 1x1KBytes, 1x512Bytes, 4x256Bytes, 96x16Bytes.

Table 8-3 OTP Address Distribution Table

| Sector | OTP Block Data Address | Capacity | OTP Block Lock Address |
|--------|------------------------|----------|------------------------|
| 0      | 0x00000000-0x00001FFF  | 8KB      | 0x03001800-0x0300180F  |
| 1      | 0x00002000-0x00003FFF  | 8KB      | 0x03001810-0x0300181F  |
| ...    | ...                    | ...      | ...                    |
| 14     | 0x0001C000-0x0001DFFF  | 8KB      | 0x030018E0-0x030018EF  |
| 15     | 0x0001E000-0x0001FFFF  | 8KB      | 0x030018F0-0x030018FF  |
| 16     | 0x03000000-0x0300000F  | 16B      | 0x03001900-0x0300190F  |
| ...    | ...                    | ...      | ...                    |
| 111    | 0x030005F0-0x030005FF  | 16B      | 0x03001EF0-0x03001EFF  |
| 112    | 0x03000600-0x030007FF  | 512B     | 0x03001F00-0x03001F0F  |
| 113    | 0x03000800-0x03000FFF  | 2KB      | 0x03001F10-0x03001F1F  |
| 114    | 0x03001000-0x030010FF  | 256B     | 0x03001F20-0x03001F2F  |
| 115    | 0x03001100-0x030011FF  | 256B     | 0x03001F30-0x03001F3F  |
| 116    | 0x03001200-0x030012FF  | 256B     | 0x03001F40-0x03001F4F  |
| 117    | 0x03001300-0x030013FF  | 256B     | 0x03001F50-0x03001F5F  |

| Sector | OTP Block Data Address | Capacity | OTP Block Lock Address |
|--------|------------------------|----------|------------------------|
| 118    | 0x03001400-0x030017FF  | 1KB      | 0x03001F60-0x03001F6F  |

To enable OTP functionality, 4x32-bit data containing zeros (recommended to write 128-bit all zeros) need to be programmed into the OTP enable addresses (0x03001FF0-0x03001FFF).

When the OTP block latch addresses contain 4x32-bit data with zeros, the corresponding block space cannot be programmed, erased, or full erased again.

OTP sectors 0-15 share 128KBytes of physical space with Flash block 0 addresses 0x00000000-0x0001FFFF; if the latch address data for this space is all 1s, it is a regular Flash space that can be programmed and erased multiple times; if the latch address data contains zeros, the corresponding sectors are read-only, and Flash block 0 cannot be bulk erased.

To program OTP latch addresses, unlock the EFM\_KEY2 register (EFM\_KEY2 is first written with 0x10325476, then with 0xEFCDAB89). If an incorrect sequence is written to the EFM\_KEY2 register, a bus error will occur, and the EFM\_KEY2 register will lock until reset.

Please follow the steps below to operate the OTP area:

- Step1.** Disable the Flash register write protection. (EFM\_FAPRT: First write 0x0123, then write 0x3210).
- Step2.** Unlock the EFM\_KEY1 lock. (EFM\_KEY1: First write 0x01234567, then write 0xFEDCBA98).
- Step3.** Set programming mode. (EFM\_FWMC.PEMOD=0b001/0b010/0b011).
- Step4.** Write data to OTP addresses as needed.
- Step5.** Unlock the EFM\_KEY2 lock. (EFM\_KEY2: First write 0x10325476, then write 0xEFCDAB89).
- Step6.** Write 0 to any bit of each of the 4x32-bit OTP enable addresses (0x03001FF0-0x03001FFF). (Recommended to write 128-bit all 0)
- Step7.** Write 0 to any bit of each of the 4 32-bit OTP latch addresses. (It is recommended to write 128-bit all 0)

After chip reset, the hardware will automatically load OTP latch address data into the circuit; if the OTP enable addresses (0x03001FF0-0x03001FF) have already been programmed with 0, step 6 can be omitted.

Programming, erasing, or full erasing the OTP data area that has been latched will generate an OTP-WERR0 error flag.

After writing to OTP latch addresses, to return to the EFM\_KEY2 locked state, write 1 to EFM\_FWMC.KEY2LOCK.

### 8.3.6 Boot Swap

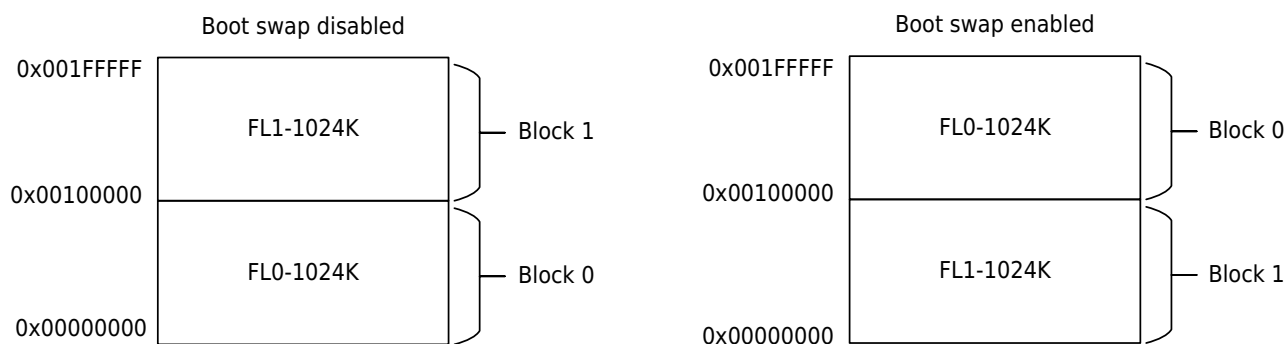
Programming data 0x5A5A5A into the boot swap enable address (0x03002000) enables the boot swap function

When OTP functionality is not enabled (0x03001FF0-0x03001FFF are all 1), after programming the boot swap enable address with a reset value, the addresses of Flash block 0 and Flash block 1 are swapped entirely; at this time, the OTP enable addresses cannot be programmed. The boot swap status can be queried through the EFM\_FSWP.FSWP bit.

When OTP functionality is enabled (the 4x32-bit data at 0x03001FF0-0x03001FFF contain zeros), after programming the boot swap enable address with a reset value, the addresses of Flash block 0 and Flash block 1 are partially swapped. At this time, EFM\_FSWP.FSWP is 0, and the address swap function can be judged by reading the data at address 0x03002000.

The swap rules are shown in the figure below:

When the OTP function of 2MBytes product is not enabled, the Flash address distribution



When the OTP function of 2MBytes product is enabled, the Flash address distribution

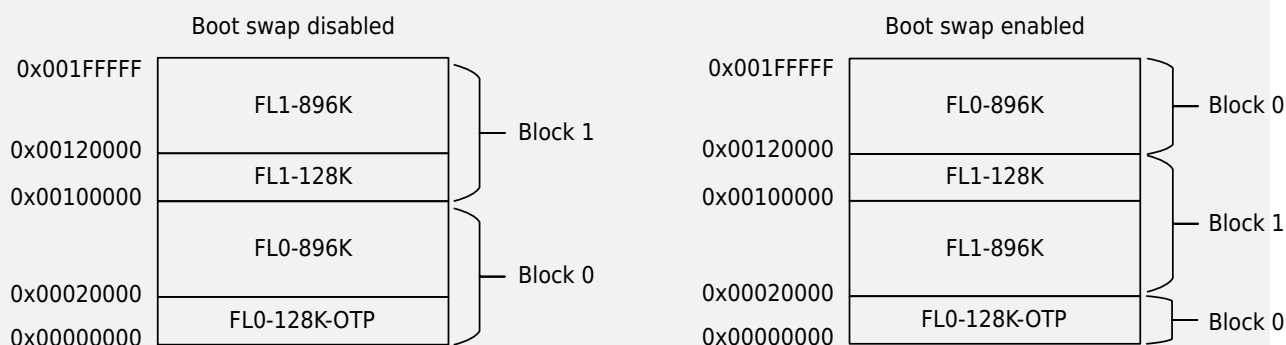


Figure 8-2 Flash Address Distribution for 2MBytes Products

Taking boot swap with OTP disabled as an example, the usage of this function is introduced.

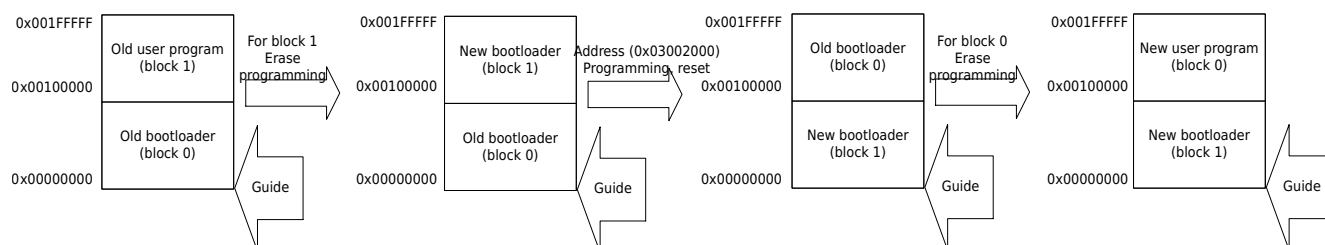


Figure 8-3 Boot Swap Activation 1

When users need to upgrade the boot program again, since address 0x03002000 has already been programmed, erasing it is required for another upgrade. The operating procedures are as follows:

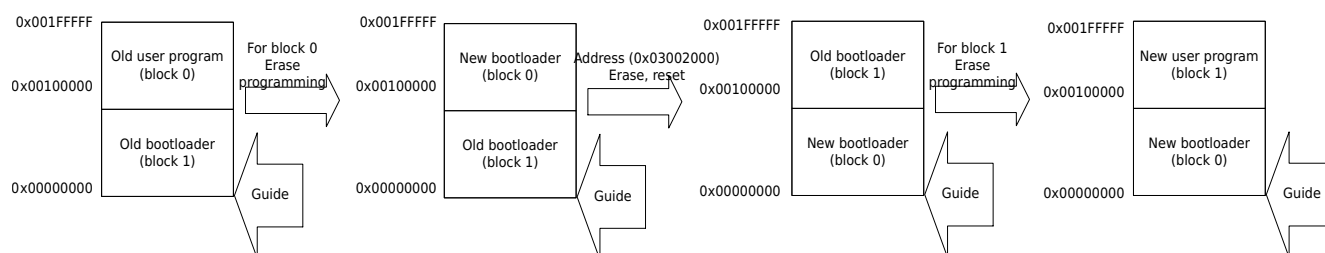


Figure 8-4 Boot Swap Activation 2

### 8.3.7 Data Reliability and Integrity

Programmed data is stored in Flash with error correction code (ECC) to prevent data loss.

#### 8.3.7.1 ECC in Flash

The 9-bit ECC in Flash is calculated based on 128-bit data and 20-bit address-related information.

The ECC in Flash has the following functions:

- Correcting one-bit errors in 128-bit data and 9-bit ECC.
- Detecting one-bit errors in 20-bit address-related information.
- Detecting two-bit errors in 128-bit data with 9-bit ECC and 20-bit address-related information.

The correction and detection of ECC in Flash are verified and recorded.

By default, ECC is automatically generated during programming. Additionally, by setting EFM\_F0ECCCR/F1ECCCR.GDIS=1, ECC uses the data in EFM\_F0ECCDR/F1ECCDR.DWD during programming.

By default, ECC is verified by hardware during read periods. Additionally, by setting EFM\_F0ECCCR/F1ECCCR.VDIS=1, ECC correction and detection are disabled, and the read ECC is sent to the register EFM\_F0ECCDR/F1ECCDR.DRD.

20-bit Address-Related Information:

- Bit [19]: Reserved bit, fixed to 0.
- Bit [18]: 1 for 0x0300 xxxx space, 0 for block 0 and block 1 space.
- Bit [17]: 1 for individual sectors of block 0 and block 1 using relief sectors, 0 for others.
- Bit [16]: Reserved bit, fixed to 0.
- Bits [15:0]: Address [19:4].

Relief Sectors: Block 0 and block 1 each have 2 additional sectors used to replace defective sectors found during mass production testing.

#### 8.3.7.2 ECC Error

Block 0 and block 1 each have 2 ECC error record registers and 1 ECC error overflow flag bit.

If an ECC error occurs, it will first set ERR0. If ERR0 is not cleared and a new ECC error occurs, it will set ERR1. If both ERR0 and ERR1 are active and a new ECC error occurs, EFM\_FSR.ECEROF0/1 will be set. Therefore, customers need to clear the error record registers in a timely manner.

#### 8.3.7.3 Error Injection Read

Customers can intentionally set data errors by setting the error injection read mode, which can be used for ECC logic self-test.

Error injection read mode setup steps are as follows:

1. Disable the Flash register write protection. (Write 0x0123 to EFM\_FAPRT, then write 0x3210)
2. Set ECC verification enabled. (Example: EFM\_CKPR=0x77, EFM\_CKCR=0x03000300)
3. Set error injection read mode. (EFM\_EIEN.F0\_EIEN/F1\_EIEN=1)
4. Write data to EFM\_F0EIR/F1EIRn (n=0-4).
  - In the above registers, only 1 bit can be set to "1" to check for one-bit ECC errors.
  - 2 bits can be set to "1" to check for two-bit ECC errors.
5. Read the programmed Flash address.
6. Check the EFM\_CKSR, EFM\_FSR, EFM\_F0ERRn, EFM\_F1ERRn (n=0/1) registers.
7. Set back to Normal read mode. (EFM\_EIEN.F0\_EIEN/F1\_EIEN=0)

The above procedure should be executed on a different Flash block from the check address, or on other memory (e.g., RAM).

### 8.3.8 Interrupt

The EFM module has 3 interrupts: PE (program/erase) error interrupt, bus conflict interrupt, and operation end interrupt, which can be queried by reading the EFM\_FSR register status flag bits.

#### 1. PE error interrupt EFM\_PEERR:

- **Position:**

- ▶ When a programming, erasing, or full erasing operation is performed on a latched OTP address (EFM\_FSR.OTPWERR0=1).
- ▶ After the OTP function is enabled, the space from 0x03000000 to 0x03001FFF is erased, with the full erase operation (EFM\_FSR.OTPWERR0=1).
- ▶ Issue program, erase, and full erase operations to write-protected sectors (EFM\_FSR.PRTWERR0/1=1).
- ▶ The programming address is not 4-byte aligned or the data size is not 32 bits (EFM\_FSR.PGSZERR0/1=1).  
If the programming addresses with the last bit of 0x0, 0x4, 0x8, 0xC are not written consecutively (EFM\_FSR.PGSZERR0/1=1).
- ▶ In single programming Read-Back mode, the hardware automatically reads a value from the programming address that is inconsistent with the written value (EFM\_FSR.MISMTCH0/1=1).

- **Clear:**

Write 1 to the corresponding flag clear bit in the EFM\_FSCLR register to clear the status bit.

#### 2. Flash read and write conflict interrupt EFM\_COLERR:

- **Position:**

- ▶ During the Flash busy period (EFM\_FSR.RDY0=0 or EFM\_FSR.RDY1=0), read and write operations are performed on the Flash block. (Continuous programming mode, except for continuous writing to Flash)

- **Clear:**

Write 1 to the corresponding flag clear bit in the EFM\_FSCLR register to clear the status bit.

#### 3. End of operation interrupt EFM\_OPTEND:

- **Position:**

- ▶ Programming mode: End of programming for a single address (EFM\_FSR.OPTEND0/1=1).
- ▶ Erase mode: Sector erase, full erase end (EFM\_FSR.OPTEND0/1=1).

- **Clear:**

Write 1 to the corresponding flag clear bit in the EFM\_FSCLR register to clear the status bit.



## 8.4 Register Description

### 8.4.1 Register List

Base Address: 0x40010400

Table 8-4 Embedded Flash Memory (EFM) Register Offset

| Offset | Register     | Description                        | Width |
|--------|--------------|------------------------------------|-------|
| 0x0000 | EFM_FAPRT    | Flash Access Protect Register      | 32    |
| 0x0004 | EFM_KEY1     | Flash Key 1 Register               | 32    |
| 0x0008 | EFM_KEY2     | Flash Key 2 Register               | 32    |
| 0x0014 | EFM_FSTP     | Flash Stop Register                | 32    |
| 0x0018 | EFM_FRMC     | Flash Read Mode Register           | 32    |
| 0x001C | EFM_FWMC     | Flash Write Mode Register          | 32    |
| 0x0020 | EFM_FSR      | Flash State Register               | 32    |
| 0x0024 | EFM_FSCLR    | Flash State Clear Register         | 32    |
| 0x0028 | EFM_FITE     | Flash Interrupt Enable Register    | 32    |
| 0x002C | EFM_FSWP     | Flash Boot Swap Status Register    | 32    |
| 0x0030 | EFM_FRMC2    | Flash Read Mode Register 2         | 32    |
| 0x0040 | EFM_CHIPID   | Chip Identification Register       | 32    |
| 0x0050 | FEM_UQID0    | UNIQUE ID Register 0               | 32    |
| 0x0054 | EFM_UQID1    | UNIQUE ID Register 1               | 32    |
| 0x0058 | EFM_UQID2    | UNIQUE ID Register 2               | 32    |
| 0x0180 | EFM_WLOCK    | Flash Write Lock Register          | 32    |
| 0x0190 | EFM_FONWPRT0 | Flash0 Write Protection Register 0 | 32    |
| 0x0194 | EFM_FONWPRT1 | Flash0 Write Protection Register 1 | 32    |
| 0x0198 | EFM_FONWPRT2 | Flash0 Write Protection Register 2 | 32    |
| 0x019C | EFM_FONWPRT3 | Flash0 Write Protection Register 3 | 32    |
| 0x01A0 | EFM_F1NWPRT0 | Flash1 Write Protection Register 0 | 32    |
| 0x01A4 | EFM_F1NWPRT1 | Flash1 Write Protection Register 1 | 32    |
| 0x01A8 | EFM_F1NWPRT2 | Flash1 Write Protection Register 2 | 32    |
| 0x01AC | EFM_F1NWPRT3 | Flash1 Write Protection Register 3 | 32    |
| 0x0350 | EFM_CKCR     | Flash Check Control Register       | 32    |
| 0x0354 | EFM_CKPR     | Flash Check Protection Register    | 32    |



| Offset        | Register                    | Description                                          | Width |
|---------------|-----------------------------|------------------------------------------------------|-------|
| 0x0358        | <a href="#">EFM_CKSR</a>    | Flash Check Status Register                          | 32    |
| 0x035C        | <a href="#">EFM_EIEN</a>    | Flash Error Injection Enable Register                | 32    |
| 0x0370        | <a href="#">EFM_F0ECCCR</a> | Flash0 ECC Control Register                          | 32    |
| 0x0374        | <a href="#">EFM_F0ECCDR</a> | Flash0 ECC Data Register                             | 32    |
| 0x0378+0x04*n | <a href="#">EFM_F0ERRn</a>  | Flash0 Error Record Register n (n=0-1)               | 32    |
| 0x0380+0x04*n | <a href="#">EFM_F0EIRn</a>  | Flash0 Error Injection Enable Register n (n=0-4)     | 32    |
| 0x03B0        | <a href="#">EFM_F1ECCCR</a> | Flash1 ECC Control Register                          | 32    |
| 0x03B4        | <a href="#">EFM_F1ECCDR</a> | Flash1 ECC Data Register                             | 32    |
| 0x03B8+0x04*n | <a href="#">EFM_F1ERRn</a>  | Flash1 Error Record Register n (n=0-1)               | 32    |
| 0x03C0+0x04*n | <a href="#">EFM_F1EIRn</a>  | Flash1 Error Injection Bit Enable Register n (n=0-4) | 32    |

### 8.4.2 Flash Access Protect Register (EFM\_FAPRT)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |       |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0x0000 | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15    | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |       |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | FAPRT |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RW    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit   | Name     | Function                      | Description                                                                                                                                                                                                                                                                                                                                                                                                                            | Access |
|-------|----------|-------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:16 | Reserved | Reserved                      | -                                                                                                                                                                                                                                                                                                                                                                                                                                      | -      |
| 15:0  | FAPRT    | EFM Register Write Protection | EFM register access write-protection register.<br>Release method: Write 16-bit data "0x0123" to FAPRT first, then write 16-bit data "0x3210".<br>In the unprotected state, write any data, and the EFM register re-enters the protected state.<br>When EFM register access protection is enabled, the read value of this register is 0x00000000.<br>When EFM register access protection is disabled, the register reads as 0x00000001. | RW     |

### 8.4.3 Flash Key 1 Register (EFM\_KEY1)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0x0004 | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | KEY1         |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access | RW           |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit  | Name | Function       | Description                                                                                                                                                                        | Access |
|------|------|----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:0 | KEY1 | Key 1 Register | EFM_FWMC write protection register.<br>Write 0x01234567 to EFM_KEY1 first, then write 0xFEDCBA98 to release the write protection of EFM_FWMC.<br>The register reads as 0x00000000. | RW     |

## 8.4.4 Flash Key 2 Register (EFM\_KEY2)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0x0008 | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | KEY2         |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access | RW           |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit  | Name | Function       | Description                                                                                                                                                                                                  | Access |
|------|------|----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:0 | KEY2 | Key 2 Register | Write protection register for OTP latch address.<br>Write 0x10325476 to EFM_KEY2 first, then write 0xEFCDAB89 to release the write protection of the OTP latch address.<br>The register reads as 0x00000000. | RW     |

## 8.4.5 Flash Stop Register (EFM\_FSTP)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |       |   |       |  |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|-------|---|-------|--|
| 0x0014 | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1     | 0 |       |  |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |       |   |       |  |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   | F1STP |   | F0STP |  |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   | RW    |   | RW    |  |

| Bit  | Name     | Function                        | Description                                                                                                                                                                                                                                                                                                                                                                | Access |
|------|----------|---------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:2 | Reserved | Reserved                        | -                                                                                                                                                                                                                                                                                                                                                                          | -      |
| 1    | F1STP    | Flash Block 1 Stop Mode Control | <ul style="list-style-type: none"> <li>0b0: Flash Block 1 active state</li> <li>0b1: Flash Block 1 in Stop mode</li> </ul> <div>  <b>Note</b><br/>           After this bit is set from 1 to 0, access Flash block 1 after confirming that the FSR.RDY1 bit is 1.         </div>        | RW     |
| 0    | F0STP    | Flash Block 0 Stop Mode Control | <ul style="list-style-type: none"> <li>0b0: Flash Block 0 active state</li> <li>0b1: Flash Block 0 in Stop mode</li> </ul> <div>  <b>Note</b><br/>           When this bit is set from 1 to 0, please confirm that the FSR.RDY0 bit is 1 before accessing Flash Block 0.         </div> | RW     |

## 8.4.6 Flash Read Mode Register (EFM\_FRMC)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |      |    |       |    |        |    |        |          |    |   |   |   |   |    |     |          |   |   |   |  |    |  |      |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|------|----|-------|----|--------|----|--------|----------|----|---|---|---|---|----|-----|----------|---|---|---|--|----|--|------|
| 0x0018 | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18   | 17 | 16    | 15 | 14     | 13 | 12     | 11       | 10 | 9 | 8 | 7 | 6 | 5  | 4   | 3        | 2 | 1 | 0 |  |    |  |      |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |      |    |       |    |        |    |        |          |    |   |   |   |   |    |     |          |   |   |   |  |    |  |      |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    | CRST |    | PREFE |    | DCACHE |    | ICACHE | Reserved |    |   |   |   |   |    | LVM | Reserved |   |   |   |  |    |  | FLWT |
| Access |              |    |    |    |    |    |    |    |    |    |    |    | RW |      | RW |       | RW |        | RW |        |          |    |   |   |   |   | RW |     |          |   |   |   |  | RW |  |      |

| Bit   | Name     | Function                       | Description                                                                                                                                                                                                                                                                                                | Access |
|-------|----------|--------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:20 | Reserved | Reserved                       | -                                                                                                                                                                                                                                                                                                          | -      |
| 19    | CRST     | Cache Reset                    | <ul style="list-style-type: none"> <li>0b0: Cache data is de-asserted from reset</li> <li>0b1: Cache data reset (buffer, cache, prefetch)</li> </ul>                                                                                                                                                       | RW     |
| 18    | PREFE    | ICODE Prefetch Permission Bit  | <ul style="list-style-type: none"> <li>0b0: Disable ICODE prefetch function.</li> <li>0b1: ICODE prefetch function enabled</li> </ul>                                                                                                                                                                      | RW     |
| 17    | DCACHE   | DCODE Cache Permission Bits    | <ul style="list-style-type: none"> <li>0b0: Disable DCODE cache function</li> <li>0b1: DCODE cache function enabled</li> </ul>                                                                                                                                                                             | RW     |
| 16    | ICACHE   | ICODE Cache Permission Bits    | <ul style="list-style-type: none"> <li>0b0: Disable ICODE cache function</li> <li>0b1: ICODE cache function enabled</li> </ul>                                                                                                                                                                             | RW     |
| 15:9  | Reserved | Reserved                       | -                                                                                                                                                                                                                                                                                                          | -      |
| 8     | LVM      | Low Voltage Read Mode          | <ul style="list-style-type: none"> <li>0b0: Normal voltage read mode</li> <li>0b1: Low voltage read mode</li> </ul> <div>  <b>Note</b><br/>           In ultra-low-speed operation mode, set LVM to "1".         </div> | RW     |
| 7:4   | Reserved | Reserved                       | -                                                                                                                                                                                                                                                                                                          | -      |
| 3:0   | FLWT     | Flash Read Inserted Wait State | <ul style="list-style-type: none"> <li>0b0000: No read inserted wait state</li> <li>0b0001: Insert 1 read wait state</li> <li>0b0010: Insert 2 read wait state</li> <li>.....</li> <li>0b1110: Insert 14 read wait state</li> <li>0b1111: Insert 15 read wait state</li> </ul>                             | RW     |

## 8.4.7 Flash Write Mode Register (EFM\_FWMC)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |          |    |          |    |          |    |    |    |   |            |    |            |   |           |    |          |   |   |  |  |       |  |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----------|----|----------|----|----------|----|----|----|---|------------|----|------------|---|-----------|----|----------|---|---|--|--|-------|--|
| 0x001C | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17       | 16 | 15       | 14 | 13       | 12 | 11 | 10 | 9 | 8          | 7  | 6          | 5 | 4         | 3  | 2        | 1 | 0 |  |  |       |  |
| Reset  | 0x00030000   |    |    |    |    |    |    |    |    |    |    |    |    |    |          |    |          |    |          |    |    |    |   |            |    |            |   |           |    |          |   |   |  |  |       |  |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    | KEY2LOCK |    | KEY1LOCK |    | Reserved |    |    |    |   | BLKECCRSEL |    | BLKECCWSEL |   | BUSHLDCTL |    | Reserved |   |   |  |  | PEMOD |  |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    | RW       | RW |          |    |          |    |    |    |   | RW         | RW | RW         |   |           | RW |          |   |   |  |  |       |  |

| Bit   | Name       | Function                                                                  | Description                                                                                                                                                                                                                                                                                                                                                                                                                    | Access |
|-------|------------|---------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:18 | Reserved   | Reserved                                                                  | -                                                                                                                                                                                                                                                                                                                                                                                                                              | -      |
| 17    | KEY2LOCK   | Flash Key 2 Protection Status Bit                                         | <p>Write 1 to lock the EFM_KEY2 register, write 0 has no effect</p> <p>Read:</p> <ul style="list-style-type: none"> <li>0b0: EFM_KEY2 sequence unlocked</li> <li>0b1: EFM_KEY2 sequence locked</li> </ul> <div>  <b>Note</b><br/>           After writing an incorrect sequence to EFM_KEY2, this bit remains 1 until reset.         </div> | RW     |
| 16    | KEY1LOCK   | Flash Key 1 Protection Status Bit                                         | <p>Write 1 to lock the EFM_KEY1 register, write 0 has no effect</p> <p>Read:</p> <ul style="list-style-type: none"> <li>0b0: EFM_KEY1 sequence unlocked</li> <li>0b1: EFM_KEY1 sequence locked</li> </ul> <div>  <b>Note</b><br/>           After writing an incorrect sequence to EFM_KEY1, this bit remains 1 until reset.         </div> | RW     |
| 15:11 | Reserved   | Reserved                                                                  | -                                                                                                                                                                                                                                                                                                                                                                                                                              | -      |
| 10    | BLKECCRSEL | ECC Selection Bit for Reading 128-bit Data and 9-bit ECC All 1 from Flash | <ul style="list-style-type: none"> <li>0b0: No ECC error</li> <li>0b1: ECC error determined by the algorithm</li> </ul>                                                                                                                                                                                                                                                                                                        | RW     |
| 9     | BLKECCWSEL | ECC Selection Bit for Flash Programming with 128-bit All 1 Data.          | <ul style="list-style-type: none"> <li>0b0: Data automatically generated during programming or data set by ECCDR</li> <li>0b1: 0x1FF</li> </ul>                                                                                                                                                                                                                                                                                | RW     |

| Bit | Name            | Function                               | Description                                                                                                                                                                                                                                                                                                                                                                                               | Access |
|-----|-----------------|----------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 8   | BUSHLDCTL       | Bus Control During Flash Program/Erase | <ul style="list-style-type: none"> <li>● 0b0: During Flash programming and erasing, the bus is occupied</li> <li>● 0b1: During Flash programming and erasing, the bus is released</li> </ul>                                                                                                                                                                                                              | RW     |
| 7:3 | <i>Reserved</i> | <i>Reserved</i>                        | -                                                                                                                                                                                                                                                                                                                                                                                                         | -      |
| 2:0 | PEMOD           | Flash Program/Erase Mode               | <ul style="list-style-type: none"> <li>● 0b000: Read-only Mode</li> <li>● 0b001: Single Programming Mode</li> <li>● 0b010: Single Programming with Readback Mode</li> <li>● 0b011: Continuous Programming Mode</li> <li>● 0b100: Sector Erase Mode</li> <li>● 0b101: Single Block Flash Full Erase Mode</li> <li>● 0b110: Two Blocks of Flash Full Erase Mode</li> <li>● 0b111: Read-only Mode</li> </ul> | RW     |

## 8.4.8 Flash State Register (EFM\_FSR)

| Offset | Bit Position |    |    |    |    |    |    |      |        |         |         |         |          |          |          |          |    |    |    |    |    |    |      |        |         |         |         |          |          |          |          |    |    |    |    |
|--------|--------------|----|----|----|----|----|----|------|--------|---------|---------|---------|----------|----------|----------|----------|----|----|----|----|----|----|------|--------|---------|---------|---------|----------|----------|----------|----------|----|----|----|----|
| 0x0020 | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24   | 23     | 22      | 21      | 20      | 19       | 18       | 17       | 16       | 15 | 14 | 13 | 12 | 11 | 10 | 9    | 8      | 7       | 6       | 5       | 4        | 3        | 2        | 1        | 0  |    |    |    |
| Reset  | 0x01000100   |    |    |    |    |    |    |      |        |         |         |         |          |          |          |          |    |    |    |    |    |    |      |        |         |         |         |          |          |          |          |    |    |    |    |
| Name   | Reserved     |    |    |    |    |    |    | RDY1 | BRMER1 | ECEROF1 | COLERR1 | OPTEND1 | MISMTCH1 | PGSZERR1 | PRTWERR1 | Reserved |    |    |    |    |    |    | RDY0 | BRMER0 | ECEROF0 | COLERR0 | OPTEND0 | MISMTCH0 | PGSZERR0 | PRTWERR0 | OTPWERR0 |    |    |    |    |
| Access |              |    |    |    |    |    |    | RO   | RO     | RO      | RO      | RO      | RO       | RO       | RO       |          |    |    |    |    |    |    | RO   | RO     | RO      | RO      | RO      | RO       | RO       | RO       | RO       | RO | RO | RO | RO |

| Bit   | Name     | Function                                                                    | Description                                                                                                                                                                                                                                                                                                                                                                                                                                             | Access |
|-------|----------|-----------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:25 | Reserved | Reserved                                                                    | -                                                                                                                                                                                                                                                                                                                                                                                                                                                       | -      |
| 24    | RDY1     | Flash Block 1 Busy/Idle State                                               | <ul style="list-style-type: none"> <li>0b0: Flash block 1 busy</li> <li>0b1: Flash block 1 idle</li> </ul>                                                                                                                                                                                                                                                                                                                                              | RO     |
| 23    | BRMER1   | Flash Block 1 Blank Read Error Flag Bit                                     | <ul style="list-style-type: none"> <li>0b0: Flash block 1 Blank read data is all "1"</li> <li>0b1: Flash block 1 Blank read data is not all "1"</li> </ul>                                                                                                                                                                                                                                                                                              | RO     |
| 22    | ECEROF1  | Flash Block 1 ECC Error Overflow Flag Bit                                   | <ul style="list-style-type: none"> <li>0b0: No ECC error overflow in Flash Block 1</li> <li>0b1: ECC error overflow in Flash Block 1</li> </ul>                                                                                                                                                                                                                                                                                                         | RO     |
| 21    | COLERR1  | Flash Block 1 Read and Write Conflict Error Flag Bit                        | <ul style="list-style-type: none"> <li>0b0: No flag generated</li> <li>0b1: Generate flag</li> </ul> Setting conditions: <ul style="list-style-type: none"> <li>Read and write operations on Flash Block 1 when RDY1 is not set</li> <li>In continuous programming mode, writing to Flash Block 1 without setting</li> </ul> Clearing Conditions: <ul style="list-style-type: none"> <li>Write 1 to the corresponding clear bit of EFM_FSCLR</li> </ul> | RO     |
| 20    | OPTEND1  | Flash Block 1 Space Programming/Erasing/Full Erasing Operation End Flag Bit | <ul style="list-style-type: none"> <li>0b0: No flag generated</li> <li>0b1: Generate flag</li> </ul> Setting conditions: <ul style="list-style-type: none"> <li>End of programming/erase/full erase operation</li> </ul> Clearing conditions: <ul style="list-style-type: none"> <li>Write 1 to the corresponding clear bit of EFM_FSCLR</li> </ul>                                                                                                     | RO     |

| Bit  | Name     | Function                                                                      | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | Access |
|------|----------|-------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 19   | MISMTCH1 | Flash Block 1 Space Single Programming Read-Back Value Inconsistency Flag Bit | <ul style="list-style-type: none"> <li>● 0b0: No flag generated</li> <li>● 0b1: Generate flag</li> </ul> Setting conditions: <ul style="list-style-type: none"> <li>● In single programming read-back mode, the read data from Flash Block 1 after programming is inconsistent with the written value</li> </ul> Clearing conditions: <ul style="list-style-type: none"> <li>● Write 1 to the corresponding clear bit of EFM_FSCLR</li> </ul>                                                                                                                                                                  | RO     |
| 18   | PGSZERR1 | Flash Block 1 Space Programming Address and Size Misalignment Flag Bit        | <ul style="list-style-type: none"> <li>● 0b0: No flag generated</li> <li>● 0b1: Generate flag</li> </ul> Setting conditions: <ul style="list-style-type: none"> <li>● The programming address is not 4-byte aligned, or a non-32-bit write operation occurs</li> <li>● The programming addresses with the last digit of 0x0, 0x4, 0x8, 0xC are not written consecutively (terminating the write process during consecutive writing dose not trigger the bit to be set)</li> </ul> Clearing conditions: <ul style="list-style-type: none"> <li>● Write 1 to the corresponding clear bit of EFM_FSCLR</li> </ul> | RO     |
| 17   | PRTWERR1 | Flash Block 1 Write Protection Error Flag Bit                                 | <ul style="list-style-type: none"> <li>● 0b0: No flag generated</li> <li>● 0b1: Generate flag</li> </ul> Setting conditions: <ul style="list-style-type: none"> <li>● Programming, erasing, and full erasing on the write-protected sector</li> </ul> Clearing conditions: <ul style="list-style-type: none"> <li>● Write 1 to the corresponding clear bit of EFM_FSCLR</li> </ul>                                                                                                                                                                                                                             | RO     |
| 16:9 | Reserved | Reserved                                                                      | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | -      |
| 8    | RDY0     | Flash Block 0 Busy/Idle State                                                 | <ul style="list-style-type: none"> <li>● 0b0: Flash block 0 busy</li> <li>● 0b1: Flash block 0 idle</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | RO     |
| 7    | BRMER0   | Flash Block 0 Blank Read Error Flag Bit                                       | <ul style="list-style-type: none"> <li>● 0b0: Flash block 1 Blank read data is all "1"</li> <li>● 0b1: Flash block 1 Blank read data is not all "1"</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                 | RO     |
| 6    | ECEROF0  | Flash Block 0 ECC Error Overflow Flag Bit                                     | <ul style="list-style-type: none"> <li>● 0b0: No ECC error overflow in Flash block 0</li> <li>● 0b1: ECC error overflow in Flash Block 0</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                            | RO     |



| Bit | Name     | Function                                                                      | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | Access |
|-----|----------|-------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 5   | COLERR0  | Flash Block 0 Read and Write Conflict Error Flag Bit                          | <ul style="list-style-type: none"> <li>● 0b0: No flag generated</li> <li>● 0b1: Generate flag</li> </ul> Setting conditions: <ul style="list-style-type: none"> <li>● Read and write operations on Flash Block 0 when RDY0 is not set</li> <li>● In continuous programming mode, writing to Flash block 0 without setting</li> </ul> Clearing conditions: <ul style="list-style-type: none"> <li>● Write 1 to the corresponding clear bit of EFM_FSCLR</li> </ul>                                                                                                                                              | RO     |
| 4   | OPTEND0  | Flash Block 0 Space Program/Erasing/Full Erasing Operation End Flag Bit       | <ul style="list-style-type: none"> <li>● 0b0: No flag generated</li> <li>● 0b1: Generate flag</li> </ul> Setting conditions: <ul style="list-style-type: none"> <li>● End of programming/erase/full erase operation</li> </ul> Clearing conditions: <ul style="list-style-type: none"> <li>● Write 1 to the corresponding clear bit of EFM_FSCLR</li> </ul>                                                                                                                                                                                                                                                    | RO     |
| 3   | MISMTCH0 | Flash Block 0 Space Single Programming Read-Back Value Inconsistency Flag Bit | <ul style="list-style-type: none"> <li>● 0b0: No flag generated</li> <li>● 0b1: Generate flag</li> </ul> Setting conditions: <ul style="list-style-type: none"> <li>● In single programming read-back mode, the read data from Flash block 0 after programming is inconsistent with the written value</li> </ul> Clearing conditions: <ul style="list-style-type: none"> <li>● Write 1 to the corresponding clear bit of EFM_FSCLR</li> </ul>                                                                                                                                                                  | RO     |
| 2   | PGSZERR0 | Flash Block 0 Space Programming Address and Size Misalignment Flag Bit        | <ul style="list-style-type: none"> <li>● 0b0: No flag generated</li> <li>● 0b1: Generate flag</li> </ul> Setting conditions: <ul style="list-style-type: none"> <li>● The programming address is not 4-byte aligned, or a non-32-bit write operation occurs</li> <li>● The programming addresses with the last digit of 0x0, 0x4, 0x8, 0xC are not written consecutively (terminating the write process during consecutive writing dose not trigger the bit to be set)</li> </ul> Clearing conditions: <ul style="list-style-type: none"> <li>● Write 1 to the corresponding clear bit of EFM_FSCLR</li> </ul> | RO     |
| 1   | PRTWERR0 | Flash Block 0 Write Protection Error Flag Bit                                 | <ul style="list-style-type: none"> <li>● 0b0: No flag generated</li> <li>● 0b1: Generate flag</li> </ul> Setting conditions: <ul style="list-style-type: none"> <li>● Programming, erasing, and full erasing on the write-protected sector</li> </ul> Clearing conditions: <ul style="list-style-type: none"> <li>● Write 1 to the corresponding clear bit of EFM_FSCLR</li> </ul>                                                                                                                                                                                                                             | RO     |

| Bit | Name     | Function                              | Description                                                                                                                                                                                                                                                                                                                                                         | Access |
|-----|----------|---------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 0   | OTPWERR0 | OTP Erase Error Flag of Flash Block 0 | <ul style="list-style-type: none"><li>● 0b0: No flag generated</li><li>● 0b1: Generate flag</li></ul> Setting conditions: <ul style="list-style-type: none"><li>● Programming, erasing, and full erasing on the OTP latch area</li></ul> Clearing conditions: <ul style="list-style-type: none"><li>● Write 1 to the corresponding clear bit of EFM_FSCLR</li></ul> | RO     |

## 8.4.9 Flash State Clear Register (EFM\_FSCLR)

| Offset | Bit Position |    |    |    |    |    |    |    |           |            |            |            |             |             |             |          |    |    |    |    |    |    |   |           |            |            |            |             |             |             |             |    |    |    |    |    |
|--------|--------------|----|----|----|----|----|----|----|-----------|------------|------------|------------|-------------|-------------|-------------|----------|----|----|----|----|----|----|---|-----------|------------|------------|------------|-------------|-------------|-------------|-------------|----|----|----|----|----|
| 0x0024 | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23        | 22         | 21         | 20         | 19          | 18          | 17          | 16       | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8         | 7          | 6          | 5          | 4           | 3           | 2           | 1           | 0  |    |    |    |    |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |           |            |            |            |             |             |             |          |    |    |    |    |    |    |   |           |            |            |            |             |             |             |             |    |    |    |    |    |
| Name   | Reserved     |    |    |    |    |    |    |    | BRMERCLR1 | ECEROFCLR1 | COLERRCLR1 | OPTENDCLR1 | MISMTCHCLR1 | PGSZERRCLR1 | PRTWERRCLR1 | Reserved |    |    |    |    |    |    |   | BRMERCLR0 | ECEROFCLR0 | COLERRCLR0 | OPTENDCLR0 | MISMTCHCLR0 | PGSZERRCLR0 | PRTWERRCLR0 | OTPVERRCLR0 |    |    |    |    |    |
| Access |              |    |    |    |    |    |    |    | RW        | RW         | RW         | RW         | RW          | RW          | RW          |          |    |    |    |    |    |    |   |           | RW         | RW         | RW         | RW          | RW          | RW          | RW          | RW | RW | RW | RW | RW |

| Bit   | Name        | Function                                                                               | Description                                                                     | Access |
|-------|-------------|----------------------------------------------------------------------------------------|---------------------------------------------------------------------------------|--------|
| 31:24 | Reserved    | Reserved                                                                               | -                                                                               | -      |
| 23    | BRMERCLR1   | Clear the Blank Read Error Flag of Flash Block 1                                       | Write 1 to clear FSR.BRMER1 bit, write 0 has no effect; read always returns 0   | RW     |
| 22    | ECEROFCLR1  | Clear the ECC Error Overflow Flag of Flash Block 1                                     | Write 1 to clear FSR.ECEROF1 bit, write 0 has no effect; read always returns 0  | RW     |
| 21    | COLERRCLR1  | Clear the Read and Write Conflict Error Flag of Flash Block 1                          | Write 1 to clear FSR.COLERR1 bit, write 0 has no effect; read always returns 0  | RW     |
| 20    | OPTENDCLR1  | Clear the Space Programming/Erasing/Full Erasing End Flag of Flash Block 1             | Write 1 to clear FSR.OPTEND1 bit, write 0 has no effect; read always returns 0  | RW     |
| 19    | MISMTCHCLR1 | Clear the Space Single Programming Read-back Value Inconsistency Flag of Flash Block 1 | Write 1 to clear FSR.MISMTCH1 bit, write 0 has no effect; read always returns 0 | RW     |
| 18    | PGSZERRCLR1 | Clear the Space Programming Address and Size Misalignment Flag of Flash Block 1        | Write 1 to clear FSR.PGSZERR1 bit, write 0 has no effect; read always returns 0 | RW     |

| Bit  | Name            | Function                                                                               | Description                                                                         | Access |
|------|-----------------|----------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|--------|
| 17   | PRTWERRCLR1     | Clear the Write Protection Error Flag of Flash Block 1                                 | Write 1 to clear FSR.PRTWERR1 bit, write 0 has no effect; read always returns 0     | RW     |
| 16:8 | <i>Reserved</i> | <i>Reserved</i>                                                                        | -                                                                                   | -      |
| 7    | BRMERCLR0       | Clear the Read Error Flag of Flash Block 0 Blank                                       | Write 1 to clear FSR.BRMER0 bit, write 0 has no effect; read always returns 0       | RW     |
| 6    | ECEROFCLR0      | Clear the ECC Error Overflow Flag of Flash Block 0                                     | Write 1 to clear FSR.ECEROF0 bit, write 0 has no effect; read always returns 0      | RW     |
| 5    | COLERRCLR0      | Clear the Read and Write Conflict Error Flag of Flash Block 0                          | Write 1 to clear the FSR.COLERR0 bit, write 0 has no effect; read always returns 0  | RW     |
| 4    | OPTENDCLR0      | Clear the Space Programming/Erasing/Full Erasing End Flag of Flash Block 0             | Write 1 to clear the FSR.OPTEND0 bit, write 0 has no effect; read always returns 0  | RW     |
| 3    | MISMTCHCLR0     | Clear the Space Single Programming Read-back Value Inconsistency Flag of Flash Block 0 | Write 1 to clear the FSR.MISMTCH0 bit, write 0 has no effect; read always returns 0 | RW     |
| 2    | PGSZERRCLR0     | Clear the Space Programming Address and Size Misalignment Flag of Flash Block 0        | Write 1 to clear the FSR.PGSZERR0 bit, write 0 has no effect; read always returns 0 | RW     |
| 1    | PRTWERRCLR0     | Clear the Write Protection Error Flag of Flash Block 0                                 | Write 1 to clear the FSR.PRTWERR0 bit, write 0 has no effect; read always returns 0 | RW     |
| 0    | OTPWERRCLR0     | Flash Block 0 OTP Erase Error Flag Cleared                                             | Write 1 to clear the FSR.OTPWERR0 bit; write 0 has no effect; read always returns 0 | RW     |

## 8.4.10 Flash Interrupt Enable Register (EFM\_FITE)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |           |           |          |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|-----------|-----------|----------|
| 0x0028 | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |           |           |          |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |           |           |          |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   | COLERRITE | OPTENDITE | PEERRITE |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   | RW        | RW        | RW       |

| Bit  | Name      | Function                                        | Description                                                                                               | Access |
|------|-----------|-------------------------------------------------|-----------------------------------------------------------------------------------------------------------|--------|
| 31:3 | Reserved  | Reserved                                        | -                                                                                                         | -      |
| 2    | COLERRITE | Read and write conflict error interrupt enabled | <ul style="list-style-type: none"> <li>0b0: Interrupt disabled</li> <li>0b1: Interrupt enabled</li> </ul> | RW     |
| 1    | OPTENDITE | Operation end interrupt enabled                 | <ul style="list-style-type: none"> <li>0b0: Interrupt disabled</li> <li>0b1: Interrupt enabled</li> </ul> | RW     |
| 0    | PEERRITE  | Programming/erasing error interrupt enabled     | <ul style="list-style-type: none"> <li>0b0: Interrupt disabled</li> <li>0b1: Interrupt enabled</li> </ul> | RW     |

## 8.4.11 Flash Boot Swap Status Register (EFM\_FSWP)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |      |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|------|
| 0x002C | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0    |
| Reset  | 0XXXXXXXXX   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |      |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   | FSWP |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   | RO   |

| Bit  | Name     | Function                             | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | Access |
|------|----------|--------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:1 | Reserved | Reserved                             | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | -      |
| 0    | FSWP     | Block 0 and block 1 address swap bit | <ul style="list-style-type: none"> <li>0b0: Flash block 0 and block 1 addresses are not swapped. After reset, the CPU starts from Flash block 0</li> <li>0b1: Flash block 0 and block 1 addresses are swapped. After reset, the CPU starts from Flash block 1</li> </ul> <div>  <b>Note</b><br/>           The initial value of the register is determined by the value at Flash address 0x03002000 to 0x03002002. If the data is 0x5A5A5A, the initial value after reset is 1; otherwise, it is 0.         </div> | RO     |

## 8.4.12 Flash Read Mode Register 2 (EFM\_FRMC2)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |      |          |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |      |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|------|----------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|------|
| 0x0030 | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16   | 15       | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |      |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |      |          |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |      |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    | BRM1 | Reserved |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   | BRM0 |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RW   |          |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   | RW   |

| Bit   | Name     | Function                         | Description                                                                                           | Access |
|-------|----------|----------------------------------|-------------------------------------------------------------------------------------------------------|--------|
| 31:17 | Reserved | Reserved                         | -                                                                                                     | -      |
| 16    | BRM1     | Flash block 1<br>Blank read mode | <ul style="list-style-type: none"> <li>0b0: Normal read mode</li> <li>0b1: Blank read mode</li> </ul> | RW     |
| 15:1  | Reserved | Reserved                         | -                                                                                                     | -      |
| 0     | BRM0     | Flash block 0<br>Blank read mode | <ul style="list-style-type: none"> <li>0b0: Normal read mode</li> <li>0b1: Blank read mode</li> </ul> | RW     |

**Note**

Blank read mode disables ECC verification. Once the Blank read is confirmed to be ended, set it back to Normal read mode.

### 8.4.13 Chip Identification Register (EFM\_CHIPID)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0x0040 | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x584804A8   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | CHIPID       |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access | RO           |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit  | Name   | Function           | Description                                                                                | Access |
|------|--------|--------------------|--------------------------------------------------------------------------------------------|--------|
| 31:0 | CHIPID | Chip Specific Flag | CHIPID[31:16]: ASCII code 0x5848 of XH<br>CHIPID[15:0]: Product model, for example, 0x04A8 | RO     |

### 8.4.14 UNIQUE ID Register (EFM\_UQID0)

| Offset | Bit Position  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|---------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0x0050 | 31            | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0xFFFFFFFF    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | LOT_ID[47:16] |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access | RO            |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit  | Name          | Function | Description         | Access |
|------|---------------|----------|---------------------|--------|
| 31:0 | LOT_ID[47:16] | LOT ID   | Chip LOT ID [47:16] | RO     |



## 8.4.15 UNIQUE ID Register (EFM\_UQID1)

| Offset | Bit Position |    |    |    |    |    |    |    |          |    |    |    |    |    |    |    |              |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----------|----|----|----|----|----|----|----|--------------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0x0054 | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23       | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15           | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0xFFFFFFFF   |    |    |    |    |    |    |    |          |    |    |    |    |    |    |    |              |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | Reserved     |    |    |    |    |    |    |    | WAFER_ID |    |    |    |    |    |    |    | LOT_ID[15:0] |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access |              |    |    |    |    |    |    |    |          |    |    |    |    |    |    |    |              |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit   | Name         | Function | Description        | Access |
|-------|--------------|----------|--------------------|--------|
| 31:24 | Reserved     | Reserved | -                  | -      |
| 23:16 | WAFER_ID     | Wafer ID | Wafer ID           | RO     |
| 15:0  | LOT_ID[15:0] | LOT ID   | Chip LOT ID [15:0] | RO     |

## 8.4.16 UNIQUE ID Register (EFM\_UQID2)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |            |    |    |    |    |    |   |   |            |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|------------|----|----|----|----|----|---|---|------------|---|---|---|---|---|---|---|
| 0x0058 | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15         | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7          | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0xFFFFFFFF   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |            |    |    |    |    |    |   |   |            |   |   |   |   |   |   |   |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | Y_LOCATION |    |    |    |    |    |   |   | X_LOCATION |   |   |   |   |   |   |   |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |            |    |    |    |    |    |   |   |            |   |   |   |   |   |   |   |

| Bit   | Name       | Function     | Description                           | Access |
|-------|------------|--------------|---------------------------------------|--------|
| 31:16 | Reserved   | Reserved     | -                                     | -      |
| 15:8  | Y_LOCATION | Y-coordinate | Y-coordinate of the chip on the Wafer | RO     |
| 7:0   | X_LOCATION | X-coordinate | X-coordinate of the chip on the Wafer | RO     |

## 8.4.17 Flash Write Lock Register (EFM\_WLOCK)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |            |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|------------|---|---|---|---|---|---|---|
| 0x0180 | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7          | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |            |   |   |   |   |   |   |   |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   | WLOCK[7:0] |   |   |   |   |   |   |   |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   | RW         |   |   |   |   |   |   |   |

| Bit  | Name     | Function                   | Description                                                                                                                                                                                                                                                                                                                                                                                                             | Access |
|------|----------|----------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:8 | Reserved | Reserved                   | -                                                                                                                                                                                                                                                                                                                                                                                                                       | -      |
| 7    | WLOCK[7] | F1NWPRT3 Register Lock Bit | <ul style="list-style-type: none"> <li>0b0: The sector write disable controlled by the F1NWPRT3 register is not locked up</li> <li>0b1: The sector write disable controlled by the F1NWPRT3 register is locked up</li> </ul> <div>  <b>Note</b><br/> Once this bit is set to 1, it can only be restored to 0 through a reset. </div>   | RW     |
| 6    | WLOCK[6] | F1NWPRT2 Register Lock Bit | <ul style="list-style-type: none"> <li>0b0: The sector write disable controlled by the F1NWPRT2 register is not locked up</li> <li>0b1: The sector write disable controlled by the F1NWPRT2 register is locked up</li> </ul> <div>  <b>Note</b><br/> Once this bit is set to 1, it can only be restored to 0 through a reset. </div> | RW     |
| 5    | WLOCK[5] | F1NWPRT1 Register Lock Bit | <ul style="list-style-type: none"> <li>0b0: The sector write disable controlled by the F1NWPRT1 register is not locked up</li> <li>0b1: The sector write disable controlled by the F1NWPRT1 register is locked up</li> </ul> <div>  <b>Note</b><br/> Once this bit is set to 1, it can only be restored to 0 through a reset. </div> | RW     |
| 4    | WLOCK[4] | F1NWPRT0 Register Lock Bit | <ul style="list-style-type: none"> <li>0b0: The sector write disable controlled by the F1NWPRT0 register is not locked up</li> <li>0b1: The sector write disable controlled by the F1NWPRT0 register is locked up</li> </ul> <div>  <b>Note</b><br/> Once this bit is set to 1, it can only be restored to 0 through a reset. </div> | RW     |

| Bit | Name     | Function                   | Description                                                                                                                                                                                                                                                                                                                                                                                                             | Access |
|-----|----------|----------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 3   | WLOCK[3] | F0NWPRT3 Register Lock Bit | <ul style="list-style-type: none"> <li>0b0: The sector write disable controlled by the F0NWPRT3 register is not locked up</li> <li>0b1: The sector write disable controlled by the F0NWPRT3 register is locked up</li> </ul> <div>  <b>Note</b><br/> Once this bit is set to 1, it can only be restored to 0 through a reset. </div>   | RW     |
| 2   | WLOCK[2] | F0NWPRT2 Register Lock Bit | <ul style="list-style-type: none"> <li>0b0: The sector write disable controlled by the F0NWPRT2 register is not locked up</li> <li>0b1: The sector write disable controlled by the F0NWPRT2 register is locked up</li> </ul> <div>  <b>Note</b><br/> Once this bit is set to 1, it can only be restored to 0 through a reset. </div>   | RW     |
| 1   | WLOCK[1] | F0NWPRT1 Register Lock Bit | <ul style="list-style-type: none"> <li>0b0: The sector write disable controlled by the F0NWPRT1 register is not locked up</li> <li>0b1: The sector write disable controlled by the F0NWPRT1 register is locked up</li> </ul> <div>  <b>Note</b><br/> Once this bit is set to 1, it can only be restored to 0 through a reset. </div>  | RW     |
| 0   | WLOCK[0] | F0NWPRT0 Register Lock Bit | <ul style="list-style-type: none"> <li>0b0: The sector write disable controlled by the F0NWPRT0 register is not locked up</li> <li>0b1: The sector write disable controlled by the F0NWPRT0 register is locked up</li> </ul> <div>  <b>Note</b><br/> Once this bit is set to 1, it can only be restored to 0 through a reset. </div> | RW     |

## 8.4.18 Flash0 Write Protection Register0(EFM\_F0NWPRT0)

| Offset | Bit Position |           |           |           |           |           |           |           |           |           |           |           |           |           |           |           |           |           |           |           |           |           |          |          |          |          |          |          |          |          |          |          |    |
|--------|--------------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----|
| 0x0190 | 31           | 30        | 29        | 28        | 27        | 26        | 25        | 24        | 23        | 22        | 21        | 20        | 19        | 18        | 17        | 16        | 15        | 14        | 13        | 12        | 11        | 10        | 9        | 8        | 7        | 6        | 5        | 4        | 3        | 2        | 1        | 0        |    |
| Reset  | 0x00000000   |           |           |           |           |           |           |           |           |           |           |           |           |           |           |           |           |           |           |           |           |           |          |          |          |          |          |          |          |          |          |          |    |
| Name   | FONWPRT31    | FONWPRT30 | FONWPRT29 | FONWPRT28 | FONWPRT27 | FONWPRT26 | FONWPRT25 | FONWPRT24 | FONWPRT23 | FONWPRT22 | FONWPRT21 | FONWPRT20 | FONWPRT19 | FONWPRT18 | FONWPRT17 | FONWPRT16 | FONWPRT15 | FONWPRT14 | FONWPRT13 | FONWPRT12 | FONWPRT11 | FONWPRT10 | FONWPRT9 | FONWPRT8 | FONWPRT7 | FONWPRT6 | FONWPRT5 | FONWPRT4 | FONWPRT3 | FONWPRT2 | FONWPRT1 | FONWPRT0 |    |
| Access | RW           | RW        | RW        | RW        | RW        | RW        | RW        | RW        | RW        | RW        | RW        | RW        | RW        | RW        | RW        | RW        | RW        | RW        | RW        | RW        | RW        | RW        | RW       | RW       | RW       | RW       | RW       | RW       | RW       | RW       | RW       | RW       | RW |

| Bit  | Name                 | Function                                    | Description                                                                                                                                                                                                                                                                                                                                                                                                                              | Access |
|------|----------------------|---------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:0 | F0NWPRTx<br>(x=31-0) | Write enable bit for Flash block 0 sector x | <ul style="list-style-type: none"> <li>0b0: Write disable for Flash block 0 sector x</li> <li>0b1: Write enable for Flash block 0 sector x</li> </ul> <div>  <b>Note</b><br/>           When WLOCK[0]=0, this bit supports writing 0 or 1.<br/><br/>           When WLOCK[0]=1, this bit only supports writing 0, write 1 has no effect.         </div> | RW     |

## 8.4.19 Flash0 Write Protection Register 1 (EFM\_F0NWPRT1)

| Offset | Bit Position |           |           |           |           |           |           |           |           |           |           |           |           |           |           |           |           |           |           |           |           |           |           |           |           |           |           |           |           |           |           |           |    |
|--------|--------------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|----|
| 0x0194 | 31           | 30        | 29        | 28        | 27        | 26        | 25        | 24        | 23        | 22        | 21        | 20        | 19        | 18        | 17        | 16        | 15        | 14        | 13        | 12        | 11        | 10        | 9         | 8         | 7         | 6         | 5         | 4         | 3         | 2         | 1         | 0         |    |
| Reset  | 0x00000000   |           |           |           |           |           |           |           |           |           |           |           |           |           |           |           |           |           |           |           |           |           |           |           |           |           |           |           |           |           |           |           |    |
| Name   | F0NWPRT63    | F0NWPRT62 | F0NWPRT61 | F0NWPRT60 | F0NWPRT59 | F0NWPRT58 | F0NWPRT57 | F0NWPRT56 | F0NWPRT55 | F0NWPRT54 | F0NWPRT53 | F0NWPRT52 | F0NWPRT51 | F0NWPRT50 | F0NWPRT49 | F0NWPRT48 | F0NWPRT47 | F0NWPRT46 | F0NWPRT45 | F0NWPRT44 | F0NWPRT43 | F0NWPRT42 | F0NWPRT41 | F0NWPRT40 | F0NWPRT39 | F0NWPRT38 | F0NWPRT37 | F0NWPRT36 | F0NWPRT35 | F0NWPRT34 | F0NWPRT33 | F0NWPRT32 |    |
| Access | RW           | RW        | RW        | RW        | RW        | RW        | RW        | RW        | RW        | RW        | RW        | RW        | RW        | RW        | RW        | RW        | RW        | RW        | RW        | RW        | RW        | RW        | RW        | RW        | RW        | RW        | RW        | RW        | RW        | RW        | RW        | RW        | RW |

| Bit  | Name                  | Function                                    | Description                                                                                                                                                                                                                                                                                                                                                                                                                              | Access |
|------|-----------------------|---------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:0 | F0NWPRTx<br>(x=63-32) | Write enable bit for Flash block 0 sector x | <ul style="list-style-type: none"> <li>0b0: Write disable for Flash block 0 sector x</li> <li>0b1: Write enable for Flash block 0 sector x</li> </ul> <div>  <b>Note</b><br/>           When WLOCK[1]=0, this bit supports writing 0 or 1.<br/><br/>           When WLOCK[1]=1, this bit only supports writing 0, write 1 is disabled.         </div> | RW     |

## 8.4.20 Flash0 Write Protection Register 2 (EFM\_FONWPRT2)

| Offset | Bit Position |           |           |           |           |           |           |           |           |           |           |           |           |           |           |           |           |           |           |           |           |           |           |           |           |           |           |           |           |           |           |           |
|--------|--------------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|
| 0x0198 | 31           | 30        | 29        | 28        | 27        | 26        | 25        | 24        | 23        | 22        | 21        | 20        | 19        | 18        | 17        | 16        | 15        | 14        | 13        | 12        | 11        | 10        | 9         | 8         | 7         | 6         | 5         | 4         | 3         | 2         | 1         | 0         |
| Reset  | 0x00000000   |           |           |           |           |           |           |           |           |           |           |           |           |           |           |           |           |           |           |           |           |           |           |           |           |           |           |           |           |           |           |           |
| Name   | FONWPRT95    | FONWPRT94 | FONWPRT93 | FONWPRT92 | FONWPRT91 | FONWPRT90 | FONWPRT89 | FONWPRT88 | FONWPRT87 | FONWPRT86 | FONWPRT85 | FONWPRT84 | FONWPRT83 | FONWPRT82 | FONWPRT81 | FONWPRT80 | FONWPRT79 | FONWPRT78 | FONWPRT77 | FONWPRT76 | FONWPRT75 | FONWPRT74 | FONWPRT73 | FONWPRT72 | FONWPRT71 | FONWPRT70 | FONWPRT69 | FONWPRT68 | FONWPRT67 | FONWPRT66 | FONWPRT65 | FONWPRT64 |
| Access | RW           | RW        | RW        | RW        | RW        | RW        | RW        | RW        | RW        | RW        | RW        | RW        | RW        | RW        | RW        | RW        | RW        | RW        | RW        | RW        | RW        | RW        | RW        | RW        | RW        | RW        | RW        | RW        | RW        | RW        | RW        | RW        |

| Bit  | Name                  | Function                                    | Description                                                                                                                                                                                                                                                                                                                                                                                                                              | Access |
|------|-----------------------|---------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:0 | FONWPRTx<br>(x=95-64) | Write enable bit for Flash block 0 sector x | <ul style="list-style-type: none"> <li>0b0: Write disable for Flash block 0 sector x</li> <li>0b1: Write enable for Flash block 0 sector x</li> </ul> <div>  <b>Note</b><br/>           When WLOCK[2]=0, this bit supports writing 0 or 1.<br/><br/>           When WLOCK[2]=1, this bit only supports writing 0, write 1 has no effect.         </div> | RW     |

## 8.4.21 Flash0 Write Protection Register 3 (EFM\_FONWPRT3)

| Offset | Bit Position |            |            |            |            |            |            |            |            |            |            |            |            |            |            |            |            |            |            |            |            |            |            |            |            |            |            |            |           |           |           |           |
|--------|--------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|-----------|-----------|-----------|-----------|
| 0x019C | 31           | 30         | 29         | 28         | 27         | 26         | 25         | 24         | 23         | 22         | 21         | 20         | 19         | 18         | 17         | 16         | 15         | 14         | 13         | 12         | 11         | 10         | 9          | 8          | 7          | 6          | 5          | 4          | 3         | 2         | 1         | 0         |
| Reset  | 0x00000000   |            |            |            |            |            |            |            |            |            |            |            |            |            |            |            |            |            |            |            |            |            |            |            |            |            |            |            |           |           |           |           |
| Name   | FONWPRT127   | FONWPRT126 | FONWPRT125 | FONWPRT124 | FONWPRT123 | FONWPRT122 | FONWPRT121 | FONWPRT120 | FONWPRT119 | FONWPRT118 | FONWPRT117 | FONWPRT116 | FONWPRT115 | FONWPRT114 | FONWPRT113 | FONWPRT112 | FONWPRT111 | FONWPRT110 | FONWPRT109 | FONWPRT108 | FONWPRT107 | FONWPRT106 | FONWPRT105 | FONWPRT104 | FONWPRT103 | FONWPRT102 | FONWPRT101 | FONWPRT100 | FONWPRT99 | FONWPRT98 | FONWPRT97 | FONWPRT96 |
| Access | RW           | RW         | RW         | RW         | RW         | RW         | RW         | RW         | RW         | RW         | RW         | RW         | RW         | RW         | RW         | RW         | RW         | RW         | RW         | RW         | RW         | RW         | RW         | RW         | RW         | RW         | RW         | RW         | RW        | RW        | RW        | RW        |

| Bit  | Name                   | Function                                    | Description                                                                                                                                                                                                                                                                                                                                                                                                                                | Access |
|------|------------------------|---------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:0 | FONWPRTx<br>(x=127-96) | Write enable bit for Flash block 0 sector x | <ul style="list-style-type: none"> <li>0b0: Write disable for Flash block 0 sector x</li> <li>0b1: Write enable for Flash block 0 sector x</li> </ul> <div>  <b>Note</b><br/>           When WLOCK[3]=0, this bit supports writing 0 or 1.<br/><br/>           When WLOCK[3]=1, this bit only supports writing 0, write 1 has no effect.         </div> | RW     |

## 8.4.22 Flash1 Write Protection Register 0 (EFM\_F1NWPRT0)

| Offset | Bit Position |           |           |           |           |           |           |           |           |           |           |           |           |           |           |           |           |           |           |           |           |           |          |          |          |          |          |          |          |          |          |          |
|--------|--------------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|
| 0x01A0 | 31           | 30        | 29        | 28        | 27        | 26        | 25        | 24        | 23        | 22        | 21        | 20        | 19        | 18        | 17        | 16        | 15        | 14        | 13        | 12        | 11        | 10        | 9        | 8        | 7        | 6        | 5        | 4        | 3        | 2        | 1        | 0        |
| Reset  | 0x00000000   |           |           |           |           |           |           |           |           |           |           |           |           |           |           |           |           |           |           |           |           |           |          |          |          |          |          |          |          |          |          |          |
| Name   | F1NWPRT31    | F1NWPRT30 | F1NWPRT29 | F1NWPRT28 | F1NWPRT27 | F1NWPRT26 | F1NWPRT25 | F1NWPRT24 | F1NWPRT23 | F1NWPRT22 | F1NWPRT21 | F1NWPRT20 | F1NWPRT19 | F1NWPRT18 | F1NWPRT17 | F1NWPRT16 | F1NWPRT15 | F1NWPRT14 | F1NWPRT13 | F1NWPRT12 | F1NWPRT11 | F1NWPRT10 | F1NWPRT9 | F1NWPRT8 | F1NWPRT7 | F1NWPRT6 | F1NWPRT5 | F1NWPRT4 | F1NWPRT3 | F1NWPRT2 | F1NWPRT1 | F1NWPRT0 |
| Access | RW           | RW        | RW        | RW        | RW        | RW        | RW        | RW        | RW        | RW        | RW        | RW        | RW        | RW        | RW        | RW        | RW        | RW        | RW        | RW        | RW        | RW        | RW       | RW       | RW       | RW       | RW       | RW       | RW       | RW       | RW       | RW       |

| Bit  | Name                 | Function                                    | Description                                                                                                                                                                                                                                                                                                                                                                                                                              | Access |
|------|----------------------|---------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:0 | F1NWPRTx<br>(x=31-0) | Write enable bit for Flash block 1 sector x | <ul style="list-style-type: none"> <li>0b0: Write disable for Flash block 1 sector x</li> <li>0b1: Write enable for Flash block 1 sector x</li> </ul> <div>  <b>Note</b><br/>           When WLOCK[4]=0, this bit supports writing 0 or 1.<br/><br/>           When WLOCK[4]=1, this bit only supports writing 0, write 1 has no effect.         </div> | RW     |

## 8.4.23 Flash1 Write Protection Register 1 (EFM\_F1NWPRT1)

| Offset | Bit Position |           |           |           |           |           |           |           |           |           |           |           |           |           |           |           |           |           |           |           |           |           |           |           |           |           |           |           |           |           |           |           |
|--------|--------------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|
| 0x01A4 | 31           | 30        | 29        | 28        | 27        | 26        | 25        | 24        | 23        | 22        | 21        | 20        | 19        | 18        | 17        | 16        | 15        | 14        | 13        | 12        | 11        | 10        | 9         | 8         | 7         | 6         | 5         | 4         | 3         | 2         | 1         | 0         |
| Reset  | 0x00000000   |           |           |           |           |           |           |           |           |           |           |           |           |           |           |           |           |           |           |           |           |           |           |           |           |           |           |           |           |           |           |           |
| Name   | F1NWPRT63    | F1NWPRT62 | F1NWPRT61 | F1NWPRT60 | F1NWPRT59 | F1NWPRT58 | F1NWPRT57 | F1NWPRT56 | F1NWPRT55 | F1NWPRT54 | F1NWPRT53 | F1NWPRT52 | F1NWPRT51 | F1NWPRT50 | F1NWPRT49 | F1NWPRT48 | F1NWPRT47 | F1NWPRT46 | F1NWPRT45 | F1NWPRT44 | F1NWPRT43 | F1NWPRT42 | F1NWPRT41 | F1NWPRT40 | F1NWPRT39 | F1NWPRT38 | F1NWPRT37 | F1NWPRT36 | F1NWPRT35 | F1NWPRT34 | F1NWPRT33 | F1NWPRT32 |
| Access | RW           | RW        | RW        | RW        | RW        | RW        | RW        | RW        | RW        | RW        | RW        | RW        | RW        | RW        | RW        | RW        | RW        | RW        | RW        | RW        | RW        | RW        | RW        | RW        | RW        | RW        | RW        | RW        | RW        | RW        | RW        | RW        |

| Bit  | Name                  | Function                                    | Description                                                                                                                                                                                                                                                                                                                                                                                                                                | Access |
|------|-----------------------|---------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:0 | F1NWPRTx<br>(x=63-32) | Write enable bit for Flash block 1 sector x | <ul style="list-style-type: none"> <li>0b0: Write disable for Flash block 1 sector x</li> <li>0b1: Write enable for Flash block 1 sector x</li> </ul> <div>  <b>Note</b><br/>           When WLOCK[5]=0, this bit supports writing 0 or 1.<br/><br/>           When WLOCK[5]=1, this bit only supports writing 0, write 1 has no effect.         </div> | RW     |

## 8.4.24 Flash1 Write Protection Register 2 (EFM\_F1NWPRT2)

| Offset | Bit Position |           |           |           |           |           |           |           |           |           |           |           |           |           |           |           |           |           |           |           |           |           |           |           |           |           |           |           |           |           |           |           |    |
|--------|--------------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|----|
| 0x01A8 | 31           | 30        | 29        | 28        | 27        | 26        | 25        | 24        | 23        | 22        | 21        | 20        | 19        | 18        | 17        | 16        | 15        | 14        | 13        | 12        | 11        | 10        | 9         | 8         | 7         | 6         | 5         | 4         | 3         | 2         | 1         | 0         |    |
| Reset  | 0x00000000   |           |           |           |           |           |           |           |           |           |           |           |           |           |           |           |           |           |           |           |           |           |           |           |           |           |           |           |           |           |           |           |    |
| Name   | F1NWPRT95    | F1NWPRT94 | F1NWPRT93 | F1NWPRT92 | F1NWPRT91 | F1NWPRT90 | F1NWPRT89 | F1NWPRT88 | F1NWPRT87 | F1NWPRT86 | F1NWPRT85 | F1NWPRT84 | F1NWPRT83 | F1NWPRT82 | F1NWPRT81 | F1NWPRT80 | F1NWPRT79 | F1NWPRT78 | F1NWPRT77 | F1NWPRT76 | F1NWPRT75 | F1NWPRT74 | F1NWPRT73 | F1NWPRT72 | F1NWPRT71 | F1NWPRT70 | F1NWPRT69 | F1NWPRT68 | F1NWPRT67 | F1NWPRT66 | F1NWPRT65 | F1NWPRT64 |    |
| Access | RW           | RW        | RW        | RW        | RW        | RW        | RW        | RW        | RW        | RW        | RW        | RW        | RW        | RW        | RW        | RW        | RW        | RW        | RW        | RW        | RW        | RW        | RW        | RW        | RW        | RW        | RW        | RW        | RW        | RW        | RW        | RW        | RW |

| Bit  | Name                  | Function                                    | Description                                                                                                                                                                                                                                                                                                                                                                                                                              | Access |
|------|-----------------------|---------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:0 | F1NWPRTx<br>(x=95-64) | Write enable bit for Flash block 1 sector x | <ul style="list-style-type: none"> <li>0b0: Write disable for Flash block 1 sector x</li> <li>0b1: Write enable for Flash block 1 sector x</li> </ul> <div>  <b>Note</b><br/>           When WLOCK[6]=0, this bit supports writing 0 or 1.<br/><br/>           When WLOCK[6]=1, this bit only supports writing 0, write 1 has no effect.         </div> | RW     |

## 8.4.25 Flash1 Write Protection Register 3 (EFM\_F1NWPRT3)

| Offset | Bit Position |            |            |            |            |            |            |            |            |            |            |            |            |            |            |            |            |            |            |            |            |            |            |            |            |            |            |            |           |           |           |           |    |
|--------|--------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|-----------|-----------|-----------|-----------|----|
| 0x01AC | 31           | 30         | 29         | 28         | 27         | 26         | 25         | 24         | 23         | 22         | 21         | 20         | 19         | 18         | 17         | 16         | 15         | 14         | 13         | 12         | 11         | 10         | 9          | 8          | 7          | 6          | 5          | 4          | 3         | 2         | 1         | 0         |    |
| Reset  | 0x00000000   |            |            |            |            |            |            |            |            |            |            |            |            |            |            |            |            |            |            |            |            |            |            |            |            |            |            |            |           |           |           |           |    |
| Name   | F1NWPRT127   | F1NWPRT126 | F1NWPRT125 | F1NWPRT124 | F1NWPRT123 | F1NWPRT122 | F1NWPRT121 | F1NWPRT120 | F1NWPRT119 | F1NWPRT118 | F1NWPRT117 | F1NWPRT116 | F1NWPRT115 | F1NWPRT114 | F1NWPRT113 | F1NWPRT112 | F1NWPRT111 | F1NWPRT110 | F1NWPRT109 | F1NWPRT108 | F1NWPRT107 | F1NWPRT106 | F1NWPRT105 | F1NWPRT104 | F1NWPRT103 | F1NWPRT102 | F1NWPRT101 | F1NWPRT100 | F1NWPRT99 | F1NWPRT98 | F1NWPRT97 | F1NWPRT96 |    |
| Access | RW           | RW         | RW         | RW         | RW         | RW         | RW         | RW         | RW         | RW         | RW         | RW         | RW         | RW         | RW         | RW         | RW         | RW         | RW         | RW         | RW         | RW         | RW         | RW         | RW         | RW         | RW         | RW         | RW        | RW        | RW        | RW        | RW |

| Bit  | Name                   | Function                                    | Description                                                                                                                                                                                                                                                                                                                                                                                                                                | Access |
|------|------------------------|---------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:0 | F1NWPRTx<br>(x=127-96) | Write enable bit for Flash block 1 sector x | <ul style="list-style-type: none"> <li>0b0: Write disable for Flash block 1 sector x</li> <li>0b1: Write enable for Flash block 1 sector x</li> </ul> <div>  <b>Note</b><br/>           When WLOCK[7]=0, this bit supports writing 0 or 1.<br/><br/>           When WLOCK[7]=1, this bit only supports writing 0, write 1 has no effect.         </div> | RW     |

## 8.4.26 Flash Check Control Register (EFM\_CKCR)

| Offset | Bit Position |    |    |    |          |    |    |    |          |    |    |    |          |    |    |    |          |    |    |    |          |    |   |   |          |   |   |   |          |   |   |   |
|--------|--------------|----|----|----|----------|----|----|----|----------|----|----|----|----------|----|----|----|----------|----|----|----|----------|----|---|---|----------|---|---|---|----------|---|---|---|
| 0x0350 | 31           | 30 | 29 | 28 | 27       | 26 | 25 | 24 | 23       | 22 | 21 | 20 | 19       | 18 | 17 | 16 | 15       | 14 | 13 | 12 | 11       | 10 | 9 | 8 | 7        | 6 | 5 | 4 | 3        | 2 | 1 | 0 |
| Reset  | 0x02000200   |    |    |    |          |    |    |    |          |    |    |    |          |    |    |    |          |    |    |    |          |    |   |   |          |   |   |   |          |   |   |   |
| Name   | Reserved     |    |    |    | F1ECCMOD |    |    |    | Reserved |    |    |    | F1ECCOAD |    |    |    | Reserved |    |    |    | F0ECCMOD |    |   |   | Reserved |   |   |   | F0ECCOAD |   |   |   |
| Access |              |    |    |    | RW       |    |    |    |          |    |    |    | RW       |    |    |    |          |    |    |    | RW       |    |   |   |          |   |   |   | RW       |   |   |   |

| Bit   | Name     | Function                                         | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | Access |
|-------|----------|--------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:26 | Reserved | Reserved                                         | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | -      |
| 25:24 | F1ECCMOD | ECC check enable bit for Flash block 1           | <ul style="list-style-type: none"> <li>● 0b00: Disable ECC check function.</li> <li>● 0b01: <ul style="list-style-type: none"> <li>▶ If a 1-bit error occurs, ECC will correct the error without generating a 1-bit error flag or causing an interrupt/reset.</li> <li>▶ If a 2-bit error occurs, ECC will detect the error, generate a 2-bit error flag, and trigger an interrupt/reset.</li> </ul> </li> <li>● 0b10: <ul style="list-style-type: none"> <li>▶ If a 1-bit error occurs, ECC will correct the error and generate a 1-bit error flag, but without causing an interrupt/reset.</li> <li>▶ If a 2-bit error occurs, ECC will detect the error, generate a 2-bit error flag, and trigger an interrupt/reset.</li> </ul> </li> <li>● 0b11: <ul style="list-style-type: none"> <li>▶ If a 1-bit error occurs, ECC will correct the error, generate a 1-bit error flag, and trigger an interrupt/reset.</li> <li>▶ If a 2-bit error occurs, ECC will detect the error, generate a 2-bit error flag, and trigger an interrupt/reset.</li> </ul> </li> </ul> | RW     |
| 23:17 | Reserved | Reserved                                         | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | -      |
| 16    | F1ECCOAD | Operation after ECC check error in Flash block 1 | <ul style="list-style-type: none"> <li>● 0b0: Non-maskable interrupt</li> <li>● 0b1: Reset</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | RW     |
| 15:10 | Reserved | Reserved                                         | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | -      |



| Bit | Name            | Function                                         | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | Access |
|-----|-----------------|--------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 9:8 | F0ECCMOD        | ECC check enable bit for Flash block 0           | <ul style="list-style-type: none"> <li>● 0b00: Disable ECC check function.</li> <li>● 0b01: <ul style="list-style-type: none"> <li>▶ If a 1-bit error occurs, ECC will correct the error without generating a 1-bit error flag or causing an interrupt/reset.</li> <li>▶ If a 2-bit error occurs, ECC will detect the error, generate a 2-bit error flag, and trigger an interrupt/reset.</li> </ul> </li> <li>● 0b10: <ul style="list-style-type: none"> <li>▶ If a 1-bit error occurs, ECC will correct the error and generate a 1-bit error flag, but without causing an interrupt/reset.</li> <li>▶ If a 2-bit error occurs, ECC will detect the error, generate a 2-bit error flag, and trigger an interrupt/reset.</li> </ul> </li> <li>● 0b11: <ul style="list-style-type: none"> <li>▶ If a 1-bit error occurs, ECC will correct the error, generate a 1-bit error flag, and trigger an interrupt/reset.</li> <li>▶ If a 2-bit error occurs, ECC will detect the error, generate a 2-bit error flag, and trigger an interrupt/reset.</li> </ul> </li> </ul> | RW     |
| 7:1 | <i>Reserved</i> | <i>Reserved</i>                                  | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | -      |
| 0   | F0ECCOAD        | Operation after ECC check error in Flash block 0 | <ul style="list-style-type: none"> <li>● 0b0: Non-maskable interrupt</li> <li>● 0b1: Reset</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | RW     |

**Note**

A one-bit error in the related information of the 20-bit address is classified as a two-bit error in the description of this register.

## 8.4.27 Flash Check Protection Register (EFM\_CKPR)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |        |   |   |   |       |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|--------|---|---|---|-------|---|---|---|
| 0x0354 | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7      | 6 | 5 | 4 | 3     | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |        |   |   |   |       |   |   |   |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   | CKPRKW |   |   |   | CKPRC |   |   |   |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   | RW     |   |   |   | RW    |   |   |   |

| Bit  | Name     | Function                                   | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | Access |
|------|----------|--------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:8 | Reserved | Reserved                                   | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | -      |
| 7:1  | CKPRKW   | Write key code                             | When CKPRC needs to be set to 1 or cleared, 0x3B must be written to CKPRKW at the same time                                                                                                                                                                                                                                                                                                                                                                                       | RW     |
| 0    | CKPRC    | Flash check control register write enabled | <ul style="list-style-type: none"> <li>0b0: Write disable to the Flash check control register</li> <li>0b1: Write enable to the Flash check control register</li> </ul> <div>  <b>Note</b><br/>           This bit controls the write operation of the EFM_CKCR register. When CKPRC is 1, writing to EFM_CKCR is enabled; when CKPRC is 0, writing to EFM_CKCR is not disabled.         </div> | RW     |

**Note**

Writing 0x00000077 to this register sets CKPRC to 1; writing 0x00000076 clears CKPRC to 0.

## 8.4.28 Flash Check Status Register (EFM\_CKSR)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |         |         |          |    |    |    |   |   |   |   |   |   |   |   |   |   |  |  |         |         |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---------|---------|----------|----|----|----|---|---|---|---|---|---|---|---|---|---|--|--|---------|---------|
| 0x0358 | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15      | 14      | 13       | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |  |  |         |         |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |         |         |          |    |    |    |   |   |   |   |   |   |   |   |   |   |  |  |         |         |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | F1_2ERR | F1_1ERR | Reserved |    |    |    |   |   |   |   |   |   |   |   |   |   |  |  | F0_2ERR | F0_1ERR |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RW      | RW      |          |    |    |    |   |   |   |   |   |   |   |   |   |   |  |  | RW      | RW      |

| Bit   | Name     | Function                               | Description                                                                                                                                                            | Access |
|-------|----------|----------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:18 | Reserved | Reserved                               | -                                                                                                                                                                      | -      |
| 17    | F1_2ERR  | ECC 2-bit error flag for Flash block 1 | Write 1 to the flag, write 0 has no effect<br>Read:<br><ul style="list-style-type: none"> <li>0b0: No 2-bit ECC errors</li> <li>0b1: 2-bit ECC error occurs</li> </ul> | RW     |
| 16    | F1_1ERR  | ECC 1-bit error flag for Flash block 1 | Write 1 to the flag, write 0 has no effect<br>Read:<br><ul style="list-style-type: none"> <li>0b0: No 1-bit ECC errors</li> <li>0b1: 1-bit ECC error occurs</li> </ul> | RW     |
| 15:2  | Reserved | Reserved                               | -                                                                                                                                                                      | -      |
| 1     | F0_2ERR  | ECC 2-bit error flag for Flash block 0 | Write 1 to the flag, write 0 has no effect<br>Read:<br><ul style="list-style-type: none"> <li>0b0: No 2-bit ECC errors</li> <li>0b1: 2-bit ECC error occurs</li> </ul> | RW     |
| 0     | F0_1ERR  | ECC 1-bit error flag for Flash block 0 | Write 1 to the flag, write 0 has no effect<br>Read:<br><ul style="list-style-type: none"> <li>0b0: No 1-bit ECC errors</li> <li>0b1: 1-bit ECC error occurs</li> </ul> | RW     |

**Note**

When confirming and clearing EFM\_CKSR, it is recommended to confirm and clear EFM\_FSR.ECEROF0/1, EFM\_F0ERRn (n=0-1), and EFM\_F1ERRn (n=0-1) together to ensure that these registers are updated synchronously to reflect the latest results.

A 2-bit error in the related information of the 20-bit address is classified as a two-bit error in the description of this register.

## 8.4.29 Flash Error Injection Enable Register (EFM\_EIEN)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |         |          |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |         |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---------|----------|----|----|----|----|---|---|---|---|---|---|---|---|---|---|--|---------|
| 0x035C | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15      | 14       | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |  |         |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |         |          |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |         |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | F1_EIEN | Reserved |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  | F0_EIEN |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RW      |          |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  | RW      |

| Bit   | Name     | Function                                                         | Description                                                                                                                                                          | Access |
|-------|----------|------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:17 | Reserved | Reserved                                                         | -                                                                                                                                                                    | -      |
| 16    | F1_EIEN  | Flash block 1 error injection enable (error injection read mode) | <ul style="list-style-type: none"> <li>0b0: Error injection disabled (Normal read mode)</li> <li>0b1: Error injection enabled (error injection read mode)</li> </ul> | RW     |
| 15:1  | Reserved | Reserved                                                         | -                                                                                                                                                                    | -      |
| 0     | F0_EIEN  | Flash block 0 error injection enable (error injection read mode) | <ul style="list-style-type: none"> <li>0b0: Error injection disabled (Normal read mode)</li> <li>0b1: Error injection enabled (error injection read mode)</li> </ul> | RW     |

## 8.4.30 Flash0 ECC Control Register (EFM\_F0ECCCR)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |      |   |      |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|------|---|------|---|
| 0x0370 | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3    | 2 | 1    | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |      |   |      |   |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   | VDIS |   | GDIS |   |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   | RW   |   | RW   |   |

| Bit  | Name     | Function                         | Description                                                                                                                                                            | Access |
|------|----------|----------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:2 | Reserved | Reserved                         | -                                                                                                                                                                      | -      |
| 1    | VDIS     | Disable automatic ECC check      | <ul style="list-style-type: none"> <li>0b0: Automatic ECC check</li> <li>0b1: Send the ECC to ECCDR.DRD and then perform software check</li> </ul>                     | RW     |
| 0    | GDIS     | Disable automatic ECC generation | <ul style="list-style-type: none"> <li>0b0: The ECC uses data automatically generated during programming</li> <li>0b1: The ECC uses the data from ECCDR.DWD</li> </ul> | RW     |

## 8.4.31 Flash0 ECC Data Register (EFM\_F0ECCDR)

| Offset | Bit Position |    |    |    |    |    |    |    |     |    |    |    |    |    |    |    |          |    |    |    |    |    |   |   |     |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|-----|----|----|----|----|----|----|----|----------|----|----|----|----|----|---|---|-----|---|---|---|---|---|---|---|
| 0x0374 | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23  | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15       | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7   | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |     |    |    |    |    |    |    |    |          |    |    |    |    |    |   |   |     |   |   |   |   |   |   |   |
| Name   | Reserved     |    |    |    |    |    |    |    | DRD |    |    |    |    |    |    |    | Reserved |    |    |    |    |    |   |   | DWD |   |   |   |   |   |   |   |
| Access |              |    |    |    |    |    |    |    | RO  |    |    |    |    |    |    |    |          |    |    |    |    |    |   |   | RW  |   |   |   |   |   |   |   |

| Bit   | Name     | Function               | Description            | Access |
|-------|----------|------------------------|------------------------|--------|
| 31:25 | Reserved | Reserved               | -                      | -      |
| 24:16 | DRD      | Read ECC               | Read ECC               | RO     |
| 15:9  | Reserved | Reserved               | -                      | -      |
| 8:0   | DWD      | ECC during programming | ECC during programming | RW     |

### 8.4.32 Flash0 Error Record Register n (EFM\_F0ERRn) (n=0-1)

The EFM\_F0ERR0 register indicates the type of ECC error 0 and the address index (offset) of this error.

The EFM\_F0ERR1 register indicates the type of ECC error 1 and the address index (offset) of this error.

Offset: 0x0378+0x04\*n

| Offset | Bit Position |    |    |    |    |    |    |    |       |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |          |       |       |   |
|--------|--------------|----|----|----|----|----|----|----|-------|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|----------|-------|-------|---|
| -      | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23    | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3        | 2     | 1     | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |       |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |          |       |       |   |
| Name   | Reserved     |    |    |    |    |    |    |    | INDEX |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   | Reserved | FATAL | VALID |   |
| Access |              |    |    |    |    |    |    |    |       |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   | RW       | RW    | RW    |   |

| Bit   | Name     | Function                            | Description                                                                                                                                                                                                                                                                                                                                       | Access |
|-------|----------|-------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:24 | Reserved | Reserved                            | -                                                                                                                                                                                                                                                                                                                                                 | -      |
| 23:4  | INDEX    | Offset                              | INDEX[19]: Fixed as 0.<br>INDEX[18]: The address space 0x0300XXXX is 1, block 0 and block 1 are 0.<br>INDEX[17]: Individual sectors of block 0 and block 1 using the spare sector are 1, others are 0.<br>INDEX[16]: Fixed as 0.<br>INDEX[15:0]: The offset corresponding to the address of the single block of Flash is shifted right by 4 bits. | RW     |
| 3:2   | Reserved | Reserved                            | -                                                                                                                                                                                                                                                                                                                                                 | -      |
| 1     | FATAL    | The recorded error is a fatal error | <ul style="list-style-type: none"> <li>0b0: Correctable</li> <li>0b1: Fatal</li> </ul>                                                                                                                                                                                                                                                            | RW     |
| 0     | VALID    | Register content is enable          | <ul style="list-style-type: none"> <li>0b0: Disable</li> <li>0b1: Enable</li> </ul>                                                                                                                                                                                                                                                               | RW     |

- FATAL=0, VALID=1: Detect a single-bit error in 128-bit data and 9-bit ECC.
- FATAL=1, VALID=1: Detect a single-bit error in the 20-bit address-related information, and two-bit errors in 128-bit data, 9-bit ECC, and 20-bit address-related information are detected.
- EFM\_FSR.ECEROF0=1: When EFM\_F0ERR0.VALID=1 and EFM\_F0ERR1.VALID=1, another new ECC error occurs.



#### Note

To clear the fatal error, set Bit[1] to 1 and Bit[0] to 0.

To clear the correctable error, set Bit[0] to 0.

If this register contains information about correctable errors, the INDEX[19:0] bit field is to be cleared after setting Bit[0] and Bit[1] in this register to 0. If Bit[1] is set to 1 and Bit[0] is set to 0, the INDEX[19:0] bit field is always cleared.

## 8.4.33 Flash0 Error Injection Enable Register n (EFM\_F0EIRn) (n=0-4)

Offset: 0x0380+0x04\*n

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| -      | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | EIR[31:0]    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access | RW           |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit  | Name               | Function                                                                                    | Description                                                                                                                                                                                   | Access |
|------|--------------------|---------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:0 | EIR[i]<br>(i=31-0) | Error injection bit enable (inject ECC errors into Flash data in error injection read mode) | <ul style="list-style-type: none"> <li>0b0: No error is injected into Bit[n*32+i] of the Flash read data</li> <li>0b1: Errors are injected into Bit[n*32+i] of the Flash read data</li> </ul> | RW     |

**Note**

For the EFM\_F0EIR4 register:

- Bit[31:12]: Address-related bits for Flash.
- Bit[11:9]: Reserved.
- Bit[8:0]: ECC bits for Flash.

## 8.4.34 Flash1 ECC Control Register (EFM\_F1ECCCR)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |      |   |      |  |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|------|---|------|--|
| 0x03B0 | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1    | 0 |      |  |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |      |   |      |  |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   | VDIS |   | GDIS |  |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   | RW   |   | RW   |  |

| Bit  | Name     | Function                         | Description                                                                                                                                                            | Access |
|------|----------|----------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:2 | Reserved | Reserved                         | -                                                                                                                                                                      | -      |
| 1    | VDIS     | Disable automatic ECC check      | <ul style="list-style-type: none"> <li>0b0: Automatic ECC check</li> <li>0b1: Send the ECC to ECCDR.DRD and then perform software check</li> </ul>                     | RW     |
| 0    | GDIS     | Disable automatic ECC generation | <ul style="list-style-type: none"> <li>0b0: The ECC uses data automatically generated during programming</li> <li>0b1: The ECC uses the data from ECCDR.DWD</li> </ul> | RW     |

## 8.4.35 Flash1 ECC Data Register (EFM\_F1ECCDR)

| Offset | Bit Position |    |    |    |    |    |    |    |     |    |    |    |    |    |    |    |          |    |    |    |    |    |   |   |     |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|-----|----|----|----|----|----|----|----|----------|----|----|----|----|----|---|---|-----|---|---|---|---|---|---|---|
| 0x03B4 | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23  | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15       | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7   | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |     |    |    |    |    |    |    |    |          |    |    |    |    |    |   |   |     |   |   |   |   |   |   |   |
| Name   | Reserved     |    |    |    |    |    |    |    | DRD |    |    |    |    |    |    |    | Reserved |    |    |    |    |    |   |   | DWD |   |   |   |   |   |   |   |
| Access |              |    |    |    |    |    |    |    | RO  |    |    |    |    |    |    |    |          |    |    |    |    |    |   |   | RW  |   |   |   |   |   |   |   |

| Bit   | Name     | Function               | Description            | Access |
|-------|----------|------------------------|------------------------|--------|
| 31:25 | Reserved | Reserved               | -                      | -      |
| 24:16 | DRD      | Read ECC               | Read ECC               | RO     |
| 15:9  | Reserved | Reserved               | -                      | -      |
| 8:0   | DWD      | ECC during programming | ECC during programming | RW     |



### 8.4.36 Flash1 Error Log Register n (EFM\_F1ERRn) (n=0-1)

The EFM\_F1ERR0 register indicates the type of ECC error 0 and the address index (offset) of this error.

The EFM\_F1ERR1 register indicates the type of ECC error 1 and the address index (offset) of this error.

Offset: 0x03B8+0x04\*n

| Offset | Bit Position |    |    |    |    |    |    |    |       |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |          |       |       |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|-------|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|----------|-------|-------|---|---|---|---|---|
| -      | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23    | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7        | 6     | 5     | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |       |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |          |       |       |   |   |   |   |   |
| Name   | Reserved     |    |    |    |    |    |    |    | INDEX |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   | Reserved | FATAL | VALID |   |   |   |   |   |
|        |              |    |    |    |    |    |    |    |       |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |          |       |       |   |   |   |   |   |
| Access |              |    |    |    |    |    |    |    | RW    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |          | RW    | RW    |   |   |   |   |   |
|        |              |    |    |    |    |    |    |    |       |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |          |       |       |   |   |   |   |   |

| Bit   | Name     | Function                            | Description                                                                                                                                                                                                                                                                                                                                               | Access |
|-------|----------|-------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:24 | Reserved | Reserved                            | -                                                                                                                                                                                                                                                                                                                                                         | -      |
| 23:4  | INDEX    | Offset                              | INDEX[19]: Fixed as 0.<br>INDEX[18]: The address space 0x0300XXXX is 1, block 0 and block 1 are 0.<br>INDEX[17]: Individual sectors of block 0 and block 1 using the spare sector are 1, others are 0.<br>INDEX[16]: Fixed as 0.<br>INDEX[15:0]: The offset address corresponding to the address of the single block of Flash is shifted right by 4 bits. | RW     |
| 3:2   | Reserved | Reserved                            | -                                                                                                                                                                                                                                                                                                                                                         | -      |
| 1     | FATAL    | The recorded error is a fatal error | <ul style="list-style-type: none"> <li>0b0: Correctable</li> <li>0b1: Fatal</li> </ul>                                                                                                                                                                                                                                                                    | RW     |
| 0     | VALID    | Register content is enable          | <ul style="list-style-type: none"> <li>0b0: Disable</li> <li>0b1: Enable</li> </ul>                                                                                                                                                                                                                                                                       | RW     |

- FATAL=0, VALID=1: Detect a single-bit error in 128-bit data and 9-bit ECC.
- FATAL=1, VALID=1: Detect a single-bit error in the 20-bit address-related information, and two-bit errors in 128-bit data, 9-bit ECC, and 20-bit address-related information are detected.
- EFM\_FSR.ECEROF1=1: When EFM\_F1ERR0.VALID=1 and EFM\_F1ERR1.VALID=1, another new ECC error occurs.



#### Note

To clear the fatal error, set Bit[1] to 1 and Bit[0] to 0.

To clear the correctable error, set Bit[0] to 0.

If this register contains information about correctable errors, the INDEX[19:0] bit field is to be cleared after setting Bit[0] and Bit[1] in this register to 0. If Bit[1] is set to 1 and Bit[0] is set to 0, the INDEX[19:0] bit field is always cleared.

### 8.4.37 Flash1 Error Injection Bit Enable Register n (EFM\_F1EIRn) (n=0-4)

Offset: 0x03C0+0x04\*n

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| -      | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | EIR[31:0]    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access | RW           |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit  | Name               | Function                                                                                    | Description                                                                                                                                                                                   | Access |
|------|--------------------|---------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:0 | EIR[i]<br>(i=31-0) | Error injection bit enable (inject ECC errors into Flash data in error injection read mode) | <ul style="list-style-type: none"> <li>0b0: No error is injected into Bit[n*32+i] of the Flash read data</li> <li>0b1: Errors are injected into Bit[n*32+i] of the Flash read data</li> </ul> | RW     |



#### Note

For the EFM\_F1EIR4 register:

- Bit[31:12]: Address-related bits for Flash.
- Bit[11:9]: Reserved.
- Bit[8:0]: ECC bits for Flash.

## 8.5 Precautions

1. When erasing and writing to Flash, resetting or erasing/writing operation can forcibly stop the action and the Flash data cannot be guaranteed. Users can perform only after disabling the reset and erasing the address.
2. After the erase/write operation is completed, the hardware circuit automatically reset the cache loop.
3. Repeated programming on the same address cannot ensure the correctness of the data.
4. In continuous programming mode, the Flash analog circuit is in a high-voltage state. A long-term high-voltage state can affect the Flash features. Once continuous programming is completed, exit immediately. MCU is not disabled to enter low power mode (Sleep mode, Stop mode, Power-down mode) in continuous programming mode.
5. When using the read acceleration cache, ensure that the cache RAM does not power down.  
(PWC\_PRAMLPC.PRAMPDC2=0)
6. When the bus is in release state (EFM\_FWMC.BUSHLDCTL=1) during programming and erasing, set the vectors and subroutines to RAM or Flash blocks other than those being programmed or erased.
7. When ECC check is enabled and the unprogrammed addresses with 128-bit data and 9-bit ECC are all "1", ECC correction and detection are not guaranteed if errors occur.

## 9 Built-in SRAM (SRAM)

### 9.1 Introduction

This product has 512KB system SRAM (SRAMH/SRAM1/SRAM2/SRAM3/SRAM4) and 4KB backup SRAM (Ret SRAM).

Each SRAM can be accessed as a byte, half-word (16 bits), or full-word (32 bits). High-speed SRAM (SRAMH) read and write operations can be performed at the fastest CPU speed (240MHz) with 0 wait (i.e. 1 period), and wait periods can also be inserted. The wait period for read and write accesses to each SRAM is set by the SRAM Wait Control Register (SRAM\_WTCR).

Ret SRAM can provide 4KB of data retention space in Power-down mode.

SRAM1, SRAM2, SRAM3, SRAM4, SRAMH and Ret SRAM have ECC check. The ECC check is a one-correction-two-check code, which can correct one-bit error in the read data and address, or check 2-bit errors.

## 9.2 Functional Description

### 9.2.1 Waiting Period

The relationship between the waiting period for read and write access of each SRAM and the CPU clock frequency is as shown in the following table:

**Table 9-1 Relationship Between Waiting Period Setting of SRAM Read and Write Access and CPU Clock Frequency**

| Waiting Periods (CPU Access Periods) | Access to High-speed SRAM (SRAMH) Allowed CPU Clock Frequency Range | CPU Clock Frequency Range Allowed for Accessing the Power-down Mode Backup SRAM (Ret SRAM) | Access to Other SRAM (SRAM1, 2, 3, 4) Allowed CPU Clock Frequency Range |
|--------------------------------------|---------------------------------------------------------------------|--------------------------------------------------------------------------------------------|-------------------------------------------------------------------------|
| 0wait (1 CPU access periods)         | 0-240MHz                                                            | 0-120MHz                                                                                   | 0-200MHz                                                                |
| 1wait (2 CPU access periods)         | 0-240MHz                                                            | 0-240MHz                                                                                   | 0-240MHz                                                                |
| 2wait (3 CPU access periods)         | 0-240MHz                                                            | 0-240MHz                                                                                   | 0-240MHz                                                                |
| 3wait (4 CPU access periods)         | 0-240MHz                                                            | 0-240MHz                                                                                   | 0-240MHz                                                                |
| 4wait (5 CPU access periods)         | 0-240MHz                                                            | 0-240MHz                                                                                   | 0-240MHz                                                                |
| 5wait (6 CPU access periods)         | 0-240MHz                                                            | 0-240MHz                                                                                   | 0-240MHz                                                                |
| 6wait (7 CPU access periods)         | 0-240MHz                                                            | 0-240MHz                                                                                   | 0-240MHz                                                                |
| 7wait (8 CPU access periods)         | 0-240MHz                                                            | 0-240MHz                                                                                   | 0-240MHz                                                                |

## 9.2.2 SRAM Space Allocation

Table 9-2 SRAM Space Allocation

| Name     | Capacity | Address Range         | Check Mode |
|----------|----------|-----------------------|------------|
| SRAM1    | 128KB    | 0x20000000-0x2001FFFF | ECC check  |
| SRAM2    | 128KB    | 0x20020000-0x2003FFFF | ECC check  |
| SRAM3    | 96KB     | 0x20040000-0x20057FFF | ECC check  |
| SRAM4    | 32KB     | 0x20058000-0x2005FFFF | ECC check  |
| Ret SRAM | 4KB      | 0x200F0000-0x200FFFFF | ECC check  |
| SRAMH    | 128KB    | 0x1FFE0000-0x1FFFFFFF | ECC check  |

## 9.2.3 ECC Check Error Injection

The ECC in SRAM includes an error injection feature that can inject errors into 32-bit data and 7-bit redundancy code. Based on the configuration of the SRAM error injection enable register and the SRAM error injection bit enable register, the hardware circuit will invert the SRAM read data bit by bit to achieve error injection. Through the SRAM error injection bit enable register configuration, both 1-bit and multi-bit errors can be injected. Since the ECC circuit can only detect errors up to 2-bit, errors involving 3-bit or more may not be detected.

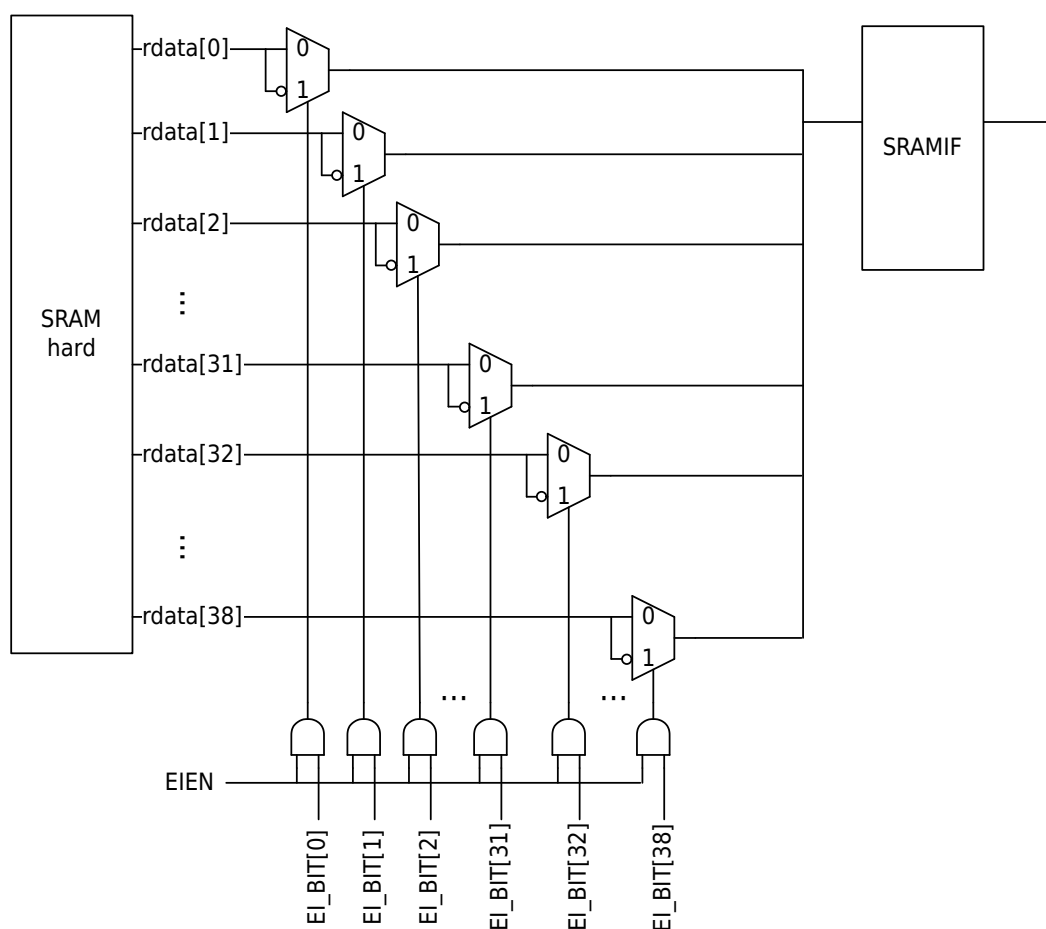


Figure 9-1 Error Injection Diagram for SRAM 39-bit

**Note**

0x20000000-0x20000002 does not support error injection.

The ECC in the internal CACHE RAM also includes an error injection feature that can inject errors into 128-bit data and 9-bit redundancy code. According to the configuration of the RAM error injection enable register and SRAM error injection bit enable register, the hardware circuit will invert the RAM read data bit by bit to perform error injection. Through the RAM error injection bit enable register configuration, both 1-bit and multi-bit errors can be injected. Since the ECC circuit can only detect errors up to 2-bit, errors involving 3-bit or more may not be detected.

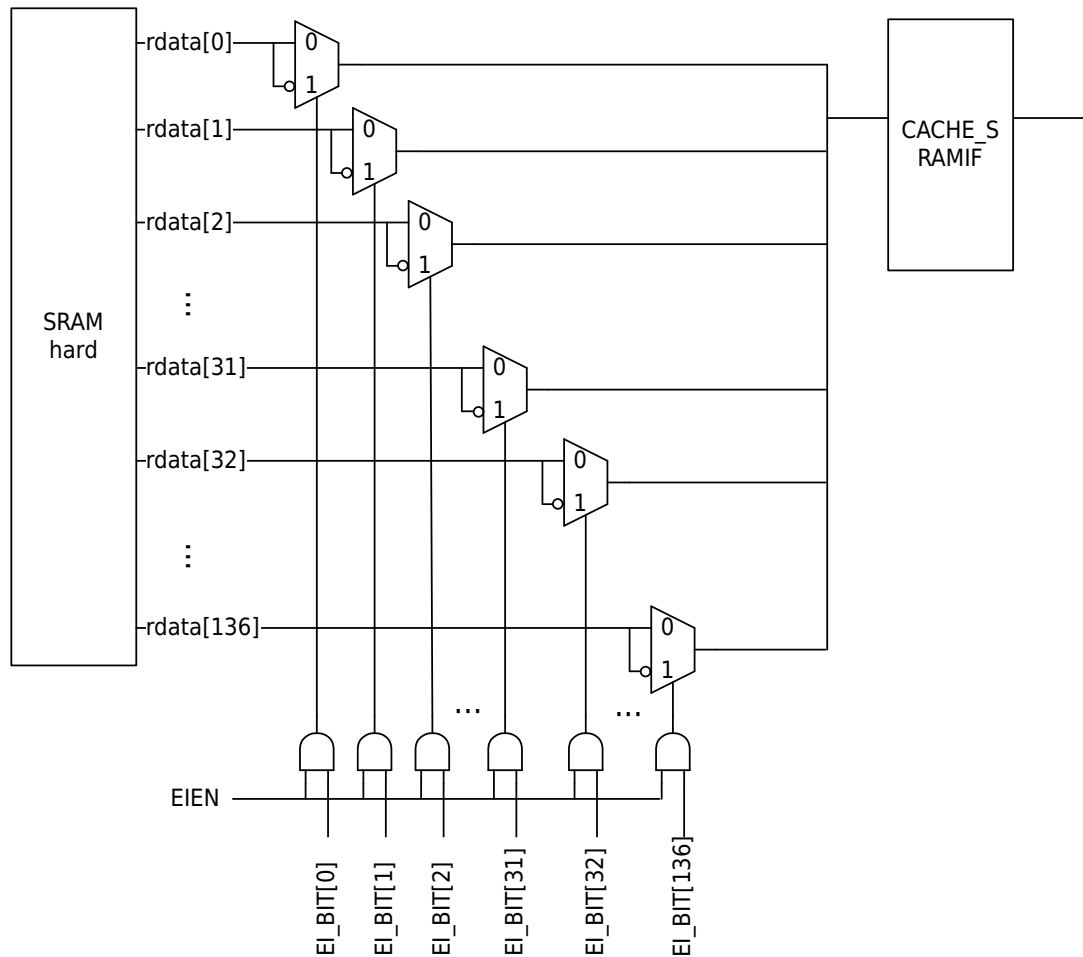


Figure 9-2 Error Injection Diagram for CACHE RAM 137-bit

## 9.3 Register Description

### 9.3.1 Register List

Base address: 0x40050800

Table 9-3 Built-in SRAM (SRAM) Register Offset

| Offset | Register         | Description                                 | Width |
|--------|------------------|---------------------------------------------|-------|
| 0x00   | SRAMC_WTCR       | SRAM Wait Time Control Register             | 32    |
| 0x04   | SRAMC_WTPR       | SRAM Wait Time Protect Register             | 32    |
| 0x08   | SRAMC_CKCR       | SRAM Check Control Register                 | 32    |
| 0x0C   | SRAMC_CKPR       | SRAM Check Protect Register                 | 32    |
| 0x10   | SRAMC_CKSR       | SRAM Check Status Register                  | 32    |
| 0x14   | SRAM1_EIEN       | SRAM1 Error Injection Enable Register       | 32    |
| 0x18   | SRAM1_EIBIT0     | SRAM1 Error Injection Bit Enable Register 0 | 32    |
| 0x1C   | SRAM1_EIBIT1     | SRAM1 Error Injection Bit Enable Register 1 | 32    |
| 0x20   | SRAM1_ECCERRADDR | SRAM1 Error Address Register                | 32    |
| 0x24   | SRAM2_EIEN       | SRAM2 Error Injection Enable Register       | 32    |
| 0x28   | SRAM2_EIBIT0     | SRAM2 Error Injection Bit Enable Register 0 | 32    |
| 0x2C   | SRAM2_EIBIT1     | SRAM2 Error Injection Bit Enable Register 1 | 32    |
| 0x30   | SRAM2_ECCERRADDR | SRAM2 Error Address Register                | 32    |
| 0x34   | SRAM3_EIEN       | SRAM3 Error Injection Enable Register       | 32    |
| 0x38   | SRAM3_EIBIT0     | SRAM3 Error Injection Bit Enable Register 0 | 32    |
| 0x3C   | SRAM3_EIBIT1     | SRAM3 Error Injection Bit Enable Register 1 | 32    |
| 0x40   | SRAM3_ECCERRADDR | SRAM3 Error Address Register                | 32    |
| 0x44   | SRAM4_EIEN       | SRAM4 Error Injection Enable Register       | 32    |
| 0x48   | SRAM4_EIBIT0     | SRAM4 Error Injection Bit Enable Register 0 | 32    |
| 0x4C   | SRAM4_EIBIT1     | SRAM4 Error Injection Bit Enable Register 1 | 32    |
| 0x50   | SRAM4_ECCERRADDR | SRAM4 Error Address Register                | 32    |
| 0x54   | SRAMH_EIEN       | SRAMH Error Injection Enable Register       | 32    |
| 0x58   | SRAMH_EIBIT0     | SRAMH Error Injection Bit Enable Register 0 | 32    |
| 0x5C   | SRAMH_EIBIT1     | SRAMH Error Injection Bit Enable Register 1 | 32    |
| 0x60   | SRAMH_ECCERRADDR | SRAMH Error Address Register                | 32    |
| 0x64   | SRAMB_EIEN       | Ret SRAM Error Injection Enable Register    | 32    |

| Offset | Register                         | Description                                    | Width |
|--------|----------------------------------|------------------------------------------------|-------|
| 0x68   | <a href="#">SRAMB_EIBIT0</a>     | Ret SRAM Error Injection Bit Enable Register 0 | 32    |
| 0x6C   | <a href="#">SRAMB_EIBIT1</a>     | Ret SRAM Error Injection Bit Enable Register 1 | 32    |
| 0x70   | <a href="#">SRAMB_ECCERRADDR</a> | Ret SRAM Error Address Register                | 32    |
| 0x74   | <a href="#">CACHE_EIEN</a>       | CACHERAM Error Injection Enable Register       | 32    |
| 0x78   | <a href="#">CACHE_EIBIT0</a>     | CACHERAM Error Injection Bit Enable Register 0 | 32    |
| 0x7C   | <a href="#">CACHE_EIBIT1</a>     | CACHERAM Error Injection Bit Enable Register 1 | 32    |
| 0x80   | <a href="#">CACHE_EIBIT2</a>     | CACHERAM Error Injection Bit Enable Register 2 | 32    |
| 0x84   | <a href="#">CACHE_EIBIT3</a>     | CACHERAM Error Injection Bit Enable Register 3 | 32    |
| 0x88   | <a href="#">CACHE_EIBIT4</a>     | CACHERAM Error Injection Bit Enable Register 4 | 32    |
| 0x8C   | <a href="#">CACHE_ECCERRADDR</a> | CACHERAM Error Address Register                | 32    |



## 9.3.2 SRAM Wait Time Control Register (SRAMC\_WTCR)

| Offset | Bit Position |          |    |    |          |          |    |    |          |          |    |    |          |          |    |    |          |          |    |    |          |          |   |   |          |            |   |   |          |            |   |   |
|--------|--------------|----------|----|----|----------|----------|----|----|----------|----------|----|----|----------|----------|----|----|----------|----------|----|----|----------|----------|---|---|----------|------------|---|---|----------|------------|---|---|
| 0x00   | 31           | 30       | 29 | 28 | 27       | 26       | 25 | 24 | 23       | 22       | 21 | 20 | 19       | 18       | 17 | 16 | 15       | 14       | 13 | 12 | 11       | 10       | 9 | 8 | 7        | 6          | 5 | 4 | 3        | 2          | 1 | 0 |
| Reset  | 0x00000000   |          |    |    |          |          |    |    |          |          |    |    |          |          |    |    |          |          |    |    |          |          |   |   |          |            |   |   |          |            |   |   |
| Name   | Reserved     | SRAMBWWT |    |    | Reserved | SRAMBRWT |    |    | Reserved | SRAMHWWT |    |    | Reserved | SRAMHRWT |    |    | Reserved | SRAM4WWT |    |    | Reserved | SRAM4RWT |   |   | Reserved | SRAM123WWT |   |   | Reserved | SRAM123RWT |   |   |
| Access |              | RW       |    |    |          | RW       |    |    |          | RW       |    |    |          | RW       |    |    |          | RW       |    |    |          | RW       |   |   |          | RW         |   |   |          | RW         |   |   |

| Bit   | Name     | Function                        | Description                                                                                                                                                  | Access |
|-------|----------|---------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31    | Reserved | Reserved                        | -                                                                                                                                                            | -      |
| 30:28 | SRAMBWWT | Ret SRAM write period selection | <ul style="list-style-type: none"> <li>● 0b000: 1 period write</li> <li>● 0b001: 2 period write</li> <li>● .....</li> <li>● 0b111: 8 period write</li> </ul> | RW     |
| 27    | Reserved | Reserved                        | -                                                                                                                                                            | -      |
| 26:24 | SRAMBRWT | Ret SRAM read period selection  | <ul style="list-style-type: none"> <li>● 0b000: 1 period read</li> <li>● 0b001: 2 period read</li> <li>● .....</li> <li>● 0b111: 8 period read</li> </ul>    | RW     |
| 23    | Reserved | Reserved                        | -                                                                                                                                                            | -      |
| 22:20 | SRAMHWWT | SRAMH write period selection    | <ul style="list-style-type: none"> <li>● 0b000: 1 period write</li> <li>● 0b001: 2 period write</li> <li>● .....</li> <li>● 0b111: 8 period write</li> </ul> | RW     |
| 19    | Reserved | Reserved                        | -                                                                                                                                                            | -      |
| 18:16 | SRAMHRWT | SRAMH read period selection     | <ul style="list-style-type: none"> <li>● 0b000: 1 period read</li> <li>● 0b001: 2 period read</li> <li>● .....</li> <li>● 0b111: 8 period read</li> </ul>    | RW     |
| 15    | Reserved | Reserved                        | -                                                                                                                                                            | -      |
| 14:12 | SRAM4WWT | SRAM4 write period selection    | <ul style="list-style-type: none"> <li>● 0b000: 1 period write</li> <li>● 0b001: 2 period write</li> <li>● .....</li> <li>● 0b111: 8 period write</li> </ul> | RW     |
| 11    | Reserved | Reserved                        | -                                                                                                                                                            | -      |

| Bit  | Name            | Function                                      | Description                                                                                                                                                  | Access |
|------|-----------------|-----------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 10:8 | SRAM4RWT        | SRAM4 read period selection                   | <ul style="list-style-type: none"> <li>● 0b000: 1 period read</li> <li>● 0b001: 2 period read</li> <li>● .....</li> <li>● 0b111: 8 period read</li> </ul>    | RW     |
| 7    | <i>Reserved</i> | <i>Reserved</i>                               | -                                                                                                                                                            | -      |
| 6:4  | SRAM123WWT      | SRAM1, SRAM2 and SRAM3 write period selection | <ul style="list-style-type: none"> <li>● 0b000: 1 period write</li> <li>● 0b001: 2 period write</li> <li>● .....</li> <li>● 0b111: 8 period write</li> </ul> | RW     |
| 3    | <i>Reserved</i> | <i>Reserved</i>                               | -                                                                                                                                                            | -      |
| 2:0  | SRAM123RWT      | SRAM1, SRAM2 and SRAM3 read period selection  | <ul style="list-style-type: none"> <li>● 0b000: 1 period read</li> <li>● 0b001: 2 period read</li> <li>● .....</li> <li>● 0b111: 8 period read</li> </ul>    | RW     |

## 9.3.3 SRAM Wait Time Protect Register (SRAMC\_WTPR)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |        |   |   |   |       |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|--------|---|---|---|-------|---|---|---|
| 0x04   | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7      | 6 | 5 | 4 | 3     | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |        |   |   |   |       |   |   |   |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   | WTPRKW |   |   |   | WTPRC |   |   |   |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   | RW     |   |   |   | RW    |   |   |   |

| Bit  | Name     | Function                                 | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                            | Access |
|------|----------|------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:8 | Reserved | Reserved                                 | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | -      |
| 7:1  | WTPRKW   | Write Key                                | When WTPRC needs to be set to 1 or cleared to 0, 0x3B must be written to WTPRKW at the same time                                                                                                                                                                                                                                                                                                                                                                       | RW     |
| 0    | WTPRC    | SRAM Wait Control Register Write Control | <ul style="list-style-type: none"> <li>0b0: SRAM wait control register write disabled</li> <li>0b1: SRAM wait control register write enabled</li> </ul> <div>  <b>Note</b><br/>           This bit controls the write operation of the SRAMC_WTCR register. When WTPRC is 1, writing to SRAMC_WTCR is allowed; when WTPRC is 0, writing to SRAMC_WTCR is prohibited.         </div> | RW     |

**Note**

Writing 0x00000077 to this register can set WTPRC to 1; writing 0x00000076 can clear WTPRC to 0.

## 9.3.4 SRAM Check Control Register (SRAMC\_CKCR)

**Note**

In the case where RAM ECC checking is enabled, when accessing data, the used RAM space must be initialized on a word basis.

| Offset | Bit Position |              |    |              |    |              |    |              |    |              |    |              |    |              |    |              |    |              |    |              |    |              |    |          |   |    |   |   |   |   |   |   |
|--------|--------------|--------------|----|--------------|----|--------------|----|--------------|----|--------------|----|--------------|----|--------------|----|--------------|----|--------------|----|--------------|----|--------------|----|----------|---|----|---|---|---|---|---|---|
| 0x08   | 31           | 30           | 29 | 28           | 27 | 26           | 25 | 24           | 23 | 22           | 21 | 20           | 19 | 18           | 17 | 16           | 15 | 14           | 13 | 12           | 11 | 10           | 9  | 8        | 7 | 6  | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |              |    |              |    |              |    |              |    |              |    |              |    |              |    |              |    |              |    |              |    |              |    |          |   |    |   |   |   |   |   |   |
| Name   | CACHE_ECCMOD | SRAMH_ECCMOD |    | SRAMB_ECCMOD |    | SRAM4_ECCMOD |    | SRAM3_ECCMOD |    | SRAM2_ECCMOD |    | SRAM1_ECCMOD |    | SRAMB_ECCMOD |    | SRAM4_ECCMOD |    | SRAM3_ECCMOD |    | SRAM2_ECCMOD |    | SRAM1_ECCMOD |    | Reserved |   |    |   |   |   |   |   |   |
| Access | RW           | RW           | RW | RW           | RW | RW           | RW | RW           | RW | RW           | RW | RW           | RW | RW           | RW | RW           | RW | RW           | RW | RW           | RW | RW           | RW |          |   | RW |   |   |   |   |   |   |

| Bit   | Name         | Function                         | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | Access |
|-------|--------------|----------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:30 | CACHE_ECCMOD | ECC Check Enable Bit of CACHERAM | <ul style="list-style-type: none"> <li>0b00: Disable ECC check function.</li> <li>0b01: <ul style="list-style-type: none"> <li>If a 1-bit error occurs, ECC will correct the error without generating a 1-bit error flag or causing an interrupt/reset.</li> <li>If a 2-bit error occurs, ECC will detect the error, generate a 2-bit error flag, and trigger an interrupt/reset.</li> </ul> </li> <li>0b10: <ul style="list-style-type: none"> <li>If a 1-bit error occurs, ECC will correct the error and generate a 1-bit error flag, but without causing an interrupt/reset.</li> <li>If a 2-bit error occurs, ECC will detect the error, generate a 2-bit error flag, and trigger an interrupt/reset.</li> </ul> </li> <li>0b11: <ul style="list-style-type: none"> <li>If a 1-bit error occurs, ECC will correct the error, generate a 1-bit error flag, and trigger an interrupt/reset.</li> <li>If a 2-bit error occurs, ECC will detect the error, generate a 2-bit error flag, and trigger an interrupt/reset.</li> </ul> </li> </ul> | RW     |

| Bit   | Name         | Function                      | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | Access |
|-------|--------------|-------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 29:28 | SRAMH_ECCMOD | SRAMH ECC Check Enable Bit    | <ul style="list-style-type: none"> <li>● 0b00: Disable ECC check function.</li> <li>● 0b01: <ul style="list-style-type: none"> <li>▶ If a 1-bit error occurs, ECC will correct the error without generating a 1-bit error flag or causing an interrupt/reset.</li> <li>▶ If a 2-bit error occurs, ECC will detect the error, generate a 2-bit error flag, and trigger an interrupt/reset.</li> </ul> </li> <li>● 0b10: <ul style="list-style-type: none"> <li>▶ If a 1-bit error occurs, ECC will correct the error and generate a 1-bit error flag, but without causing an interrupt/reset.</li> <li>▶ If a 2-bit error occurs, ECC will detect the error, generate a 2-bit error flag, and trigger an interrupt/reset.</li> </ul> </li> <li>● 0b11: <ul style="list-style-type: none"> <li>▶ If a 1-bit error occurs, ECC will correct the error, generate a 1-bit error flag, and trigger an interrupt/reset.</li> <li>▶ If a 2-bit error occurs, ECC will detect the error, generate a 2-bit error flag, and trigger an interrupt/reset.</li> </ul> </li> </ul> | RW     |
| 27:26 | SRAMB_ECCMOD | Ret SRAM ECC Check Enable Bit | <ul style="list-style-type: none"> <li>● 0b00: Disable ECC check function.</li> <li>● 0b01: <ul style="list-style-type: none"> <li>▶ If a 1-bit error occurs, ECC will correct the error without generating a 1-bit error flag or causing an interrupt/reset.</li> <li>▶ If a 2-bit error occurs, ECC will detect the error, generate a 2-bit error flag, and trigger an interrupt/reset.</li> </ul> </li> <li>● 0b10: <ul style="list-style-type: none"> <li>▶ If a 1-bit error occurs, ECC will correct the error and generate a 1-bit error flag, but without causing an interrupt/reset.</li> <li>▶ If a 2-bit error occurs, ECC will detect the error, generate a 2-bit error flag, and trigger an interrupt/reset.</li> </ul> </li> <li>● 0b11: <ul style="list-style-type: none"> <li>▶ If a 1-bit error occurs, ECC will correct the error, generate a 1-bit error flag, and trigger an interrupt/reset.</li> <li>▶ If a 2-bit error occurs, ECC will detect the error, generate a 2-bit error flag, and trigger an interrupt/reset.</li> </ul> </li> </ul> | RW     |
| 25:24 | SRAM4_ECCMOD | SRAM4 ECC Check Enable Bit    | <ul style="list-style-type: none"> <li>● 0b00: Disable ECC check function.</li> <li>● 0b01: <ul style="list-style-type: none"> <li>▶ If a 1-bit error occurs, ECC will correct the error without generating a 1-bit</li> </ul> </li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | RW     |

| Bit   | Name         | Function                   | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | Access |
|-------|--------------|----------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 25:24 | SRAM4_ECCMOD | SRAM4 ECC Check Enable Bit | <p>error flag or causing an interrupt/reset.</p> <ul style="list-style-type: none"> <li>▶ If a 2-bit error occurs, ECC will detect the error, generate a 2-bit error flag, and trigger an interrupt/reset.</li> <li>● 0b10: <ul style="list-style-type: none"> <li>▶ If a 1-bit error occurs, ECC will correct the error and generate a 1-bit error flag, but without causing an interrupt/reset.</li> <li>▶ If a 2-bit error occurs, ECC will detect the error, generate a 2-bit error flag, and trigger an interrupt/reset.</li> </ul> </li> <li>● 0b11: <ul style="list-style-type: none"> <li>▶ If a 1-bit error occurs, ECC will correct the error, generate a 1-bit error flag, and trigger an interrupt/reset.</li> <li>▶ If a 2-bit error occurs, ECC will detect the error, generate a 2-bit error flag, and trigger an interrupt/reset.</li> </ul> </li> </ul>                                                                                                                                                                                            | RW     |
| 23:22 | SRAM3_ECCMOD | SRAM3 ECC Check Enable Bit | <ul style="list-style-type: none"> <li>● 0b00: Disable ECC check function.</li> <li>● 0b01: <ul style="list-style-type: none"> <li>▶ If a 1-bit error occurs, ECC will correct the error without generating a 1-bit error flag or causing an interrupt/reset.</li> <li>▶ If a 2-bit error occurs, ECC will detect the error, generate a 2-bit error flag, and trigger an interrupt/reset.</li> </ul> </li> <li>● 0b10: <ul style="list-style-type: none"> <li>▶ If a 1-bit error occurs, ECC will correct the error and generate a 1-bit error flag, but without causing an interrupt/reset.</li> <li>▶ If a 2-bit error occurs, ECC will detect the error, generate a 2-bit error flag, and trigger an interrupt/reset.</li> </ul> </li> <li>● 0b11: <ul style="list-style-type: none"> <li>▶ If a 1-bit error occurs, ECC will correct the error, generate a 1-bit error flag, and trigger an interrupt/reset.</li> <li>▶ If a 2-bit error occurs, ECC will detect the error, generate a 2-bit error flag, and trigger an interrupt/reset.</li> </ul> </li> </ul> | RW     |

| Bit   | Name         | Function                           | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | Access |
|-------|--------------|------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 21:20 | SRAM2_ECCMOD | SRAM2 ECC Check Enable Bit         | <ul style="list-style-type: none"> <li>● 0b00: Disable ECC check function.</li> <li>● 0b01: <ul style="list-style-type: none"> <li>▶ If a 1-bit error occurs, ECC will correct the error without generating a 1-bit error flag or causing an interrupt/reset.</li> <li>▶ If a 2-bit error occurs, ECC will detect the error, generate a 2-bit error flag, and trigger an interrupt/reset.</li> </ul> </li> <li>● 0b10: <ul style="list-style-type: none"> <li>▶ If a 1-bit error occurs, ECC will correct the error and generate a 1-bit error flag, but without causing an interrupt/reset.</li> <li>▶ If a 2-bit error occurs, ECC will detect the error, generate a 2-bit error flag, and trigger an interrupt/reset.</li> </ul> </li> <li>● 0b11: <ul style="list-style-type: none"> <li>▶ If a 1-bit error occurs, ECC will correct the error, generate a 1-bit error flag, and trigger an interrupt/reset.</li> <li>▶ If a 2-bit error occurs, ECC will detect the error, generate a 2-bit error flag, and trigger an interrupt/reset.</li> </ul> </li> </ul> | RW     |
| 19:18 | SRAM1_ECCMOD | SRAM1 ECC Check Enable Bit         | <ul style="list-style-type: none"> <li>● 0b00: Disable ECC check function.</li> <li>● 0b01: <ul style="list-style-type: none"> <li>▶ If a 1-bit error occurs, ECC will correct the error without generating a 1-bit error flag or causing an interrupt/reset.</li> <li>▶ If a 2-bit error occurs, ECC will detect the error, generate a 2-bit error flag, and trigger an interrupt/reset.</li> </ul> </li> <li>● 0b10: <ul style="list-style-type: none"> <li>▶ If a 1-bit error occurs, ECC will correct the error and generate a 1-bit error flag, but without causing an interrupt/reset.</li> <li>▶ If a 2-bit error occurs, ECC will detect the error, generate a 2-bit error flag, and trigger an interrupt/reset.</li> </ul> </li> <li>● 0b11: <ul style="list-style-type: none"> <li>▶ If a 1-bit error occurs, ECC will correct the error, generate a 1-bit error flag, and trigger an interrupt/reset.</li> <li>▶ If a 2-bit error occurs, ECC will detect the error, generate a 2-bit error flag, and trigger an interrupt/reset.</li> </ul> </li> </ul> | RW     |
| 17    | SRAMB_ECCOAD | Ret SRAM ECC Check Error Operation | <ul style="list-style-type: none"> <li>● 0b0: Non-maskable interrupt</li> <li>● 0b1: Reset</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | RW     |

| Bit  | Name         | Function                                 | Description                                                                                           | Access |
|------|--------------|------------------------------------------|-------------------------------------------------------------------------------------------------------|--------|
| 16   | SRAM4_ECCOAD | Operation After SRAM4 ECC Check Error    | <ul style="list-style-type: none"> <li>● 0b0: Non-maskable interrupt</li> <li>● 0b1: Reset</li> </ul> | RW     |
| 15   | SRAM3_ECCOAD | Operation After SRAM3 ECC Check Error    | <ul style="list-style-type: none"> <li>● 0b0: Non-maskable interrupt</li> <li>● 0b1: Reset</li> </ul> | RW     |
| 14   | SRAM2_ECCOAD | Operation After SRAM2 ECC Check Error    | <ul style="list-style-type: none"> <li>● 0b0: Non-maskable interrupt</li> <li>● 0b1: Reset</li> </ul> | RW     |
| 13   | SRAM1_ECCOAD | Operation After SRAM1 ECC Check Error    | <ul style="list-style-type: none"> <li>● 0b0: Non-maskable interrupt</li> <li>● 0b1: Reset</li> </ul> | RW     |
| 12   | SRAMH_ECCOAD | Operation After SRAMH ECC Check Error    | <ul style="list-style-type: none"> <li>● 0b0: Non-maskable interrupt</li> <li>● 0b1: Reset</li> </ul> | RW     |
| 11   | CACHE_ECCOAD | Operation after CACHERAM ECC Check Error | <ul style="list-style-type: none"> <li>● 0b0: Non-maskable interrupt</li> <li>● 0b1: Reset</li> </ul> | RW     |
| 10:0 | Reserved     | Reserved                                 | -                                                                                                     | -      |



## 9.3.5 SRAM Check Protect Register (SRAMC\_CKPR)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |        |   |   |   |       |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|--------|---|---|---|-------|---|---|---|
| 0x0C   | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7      | 6 | 5 | 4 | 3     | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |        |   |   |   |       |   |   |   |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   | CKPRKW |   |   |   | CKPRC |   |   |   |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   | RW     |   |   |   | RW    |   |   |   |

| Bit  | Name     | Function                                 | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                          | Access |
|------|----------|------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:8 | Reserved | Reserved                                 | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | -      |
| 7:1  | CKPRKW   | Write Key                                | When CKPRC needs to be set to 1 or cleared to 0, 0x3B must be written to CKPRKW at the same time                                                                                                                                                                                                                                                                                                                                                                     | RW     |
| 0    | CKPRC    | SRAM Check Control Register Write Enable | <ul style="list-style-type: none"> <li>0b0: SRAM check control register write disabled</li> <li>0b1: SRAM check control register write enabled</li> </ul> <div>  <b>Note</b><br/>           This bit controls the write operation of the SRAM_CKCR register. When CKPRC is 1, writing to SRAM_CKCR is allowed; when CKPRC is 0, writing to SRAM_CKCR is prohibited.         </div> | RW     |

**Note**

Writing 0x00000077 to this register sets CKPRC to 1; writing 0x00000076 clears CKPRC to 0.

## 9.3.6 SRAM Check Status Register (SRAMC\_CKSR)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |            |            |            |            |            |            |            |            |            |            |          |            |            |            |            |          |    |    |   |  |  |  |  |  |  |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|----------|------------|------------|------------|------------|----------|----|----|---|--|--|--|--|--|--|
| 0x10   | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18         | 17         | 16         | 15         | 14         | 13         | 12         | 11         | 10         | 9          | 8        | 7          | 6          | 5          | 4          | 3        | 2  | 1  | 0 |  |  |  |  |  |  |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |            |            |            |            |            |            |            |            |            |            |          |            |            |            |            |          |    |    |   |  |  |  |  |  |  |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    | CACHE_2ERR | CACHE_1ERR | SRAMH_2ERR | SRAMH_1ERR | SRAM3_2ERR | SRAM3_1ERR | SRAM2_2ERR | SRAM2_1ERR | SRAM1_2ERR | SRAM1_1ERR | Reserved | SRAMB_2ERR | SRAMB_1ERR | SRAM4_2ERR | SRAM4_1ERR | Reserved |    |    |   |  |  |  |  |  |  |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    | RW         | RW         | RW         | RW         | RW         | RW         | RW         | RW         | RW         | RW         | RW       | RW         |            |            | RW         | RW       | RW | RW |   |  |  |  |  |  |  |

| Bit   | Name       | Function                               | Description                                                                                                                                                                  | Access |
|-------|------------|----------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:19 | Reserved   | Reserved                               | -                                                                                                                                                                            | -      |
| 18    | CACHE_2ERR | CACHERAM occurred ECC 2-bit error flag | Write 1 to clear the flag, write 0 has no effect<br>Read:<br><ul style="list-style-type: none"> <li>0b0: No 2-bit ECC errors</li> <li>0b1: 2-bit ECC error occurs</li> </ul> | RW     |
| 17    | CACHE_1ERR | CACHERAM occurred ECC 1-bit error flag | Write 1 to clear the flag, write 0 has no effect<br>Read:<br><ul style="list-style-type: none"> <li>0b0: No 1-bit ECC errors</li> <li>0b1: 1-bit ECC error occurs</li> </ul> | RW     |
| 16    | SRAMH_2ERR | SRAMH occurred ECC 2-bit error flag    | Write 1 to clear the flag, write 0 has no effect<br>Read:<br><ul style="list-style-type: none"> <li>0b0: No 2-bit ECC errors</li> <li>0b1: 2-bit ECC error occurs</li> </ul> | RW     |
| 15    | SRAMH_1ERR | SRAMH occurred ECC 1-bit error flag    | Write 1 to clear the flag, write 0 has no effect<br>Read:<br><ul style="list-style-type: none"> <li>0b0: No 1-bit ECC errors</li> <li>0b1: 1-bit ECC error occurs</li> </ul> | RW     |
| 14    | SRAM3_2ERR | SRAM3 occurred ECC 2-bit error flag    | Write 1 to clear the flag, write 0 has no effect<br>Read:<br><ul style="list-style-type: none"> <li>0b0: No 2-bit ECC errors</li> <li>0b1: 2-bit ECC error occurs</li> </ul> | RW     |
| 13    | SRAM3_1ERR | SRAM3 occurred ECC 1-bit error flag    | Write 1 to clear the flag, write 0 has no effect<br>Read:<br><ul style="list-style-type: none"> <li>0b0: No 1-bit ECC errors</li> <li>0b1: 1-bit ECC error occurs</li> </ul> | RW     |
| 12    | SRAM2_2ERR | SRAM2 occurred ECC 2-bit error flag    | Write 1 to clear the flag, write 0 has no effect<br>Read:<br><ul style="list-style-type: none"> <li>0b0: No 2-bit ECC errors</li> <li>0b1: 2-bit ECC error occurs</li> </ul> | RW     |

| Bit | Name            | Function                               | Description                                                                                                                                                                      | Access |
|-----|-----------------|----------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 11  | SRAM2_1ERR      | SRAM2 occurred ECC 1-bit error flag    | Write 1 to clear the flag, write 0 has no effect<br>Read:<br><ul style="list-style-type: none"> <li>● 0b0: No 1-bit ECC errors</li> <li>● 0b1: 1-bit ECC error occurs</li> </ul> | RW     |
| 10  | SRAM1_2ERR      | SRAM1 occurred ECC 2-bit error flag    | Write 1 to clear the flag, write 0 has no effect<br>Read:<br><ul style="list-style-type: none"> <li>● 0b0: No 2-bit ECC errors</li> <li>● 0b1: 2-bit ECC error occurs</li> </ul> | RW     |
| 9   | SRAM1_1ERR      | SRAM1 occurred ECC 1-bit error flag    | Write 1 to clear the flag, write 0 has no effect<br>Read:<br><ul style="list-style-type: none"> <li>● 0b0: No 1-bit ECC errors</li> <li>● 0b1: 1-bit ECC error occurs</li> </ul> | RW     |
| 8   | <i>Reserved</i> | <i>Reserved</i>                        | -                                                                                                                                                                                | -      |
| 7   | SRAMB_2ERR      | Ret SRAM occurred ECC 2-bit error flag | Write 1 to clear the flag, write 0 has no effect<br>Read:<br><ul style="list-style-type: none"> <li>● 0b0: No 2-bit ECC errors</li> <li>● 0b1: 2-bit ECC error occurs</li> </ul> | RW     |
| 6   | SRAMB_1ERR      | Ret SRAM occurred ECC 1-bit error flag | Write 1 to clear the flag, write 0 has no effect<br>Read:<br><ul style="list-style-type: none"> <li>● 0b0: No 1-bit ECC errors</li> <li>● 0b1: 1-bit ECC error occurs</li> </ul> | RW     |
| 5   | SRAM4_2ERR      | SRAM4 occurred ECC 2-bit error flag    | Write 1 to clear the flag, write 0 has no effect<br>Read:<br><ul style="list-style-type: none"> <li>● 0b0: No 2-bit ECC errors</li> <li>● 0b1: 2-bit ECC error occurs</li> </ul> | RW     |
| 4   | SRAM4_1ERR      | SRAM4 occurred ECC 1-bit error flag    | Write 1 to clear the flag, write 0 has no effect<br>Read:<br><ul style="list-style-type: none"> <li>● 0b0: No 1-bit ECC errors</li> <li>● 0b1: 1-bit ECC error occurs</li> </ul> | RW     |
| 3:0 | <i>Reserved</i> | <i>Reserved</i>                        | -                                                                                                                                                                                | -      |

## 9.3.7 SRAM1 Error Injection Enable Register (SRAM1\_EIEN)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |      |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|------|
| 0x14   | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0    |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |      |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   | EIEN |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   | RW   |

| Bit  | Name     | Function                | Description                                                                                                           | Access |
|------|----------|-------------------------|-----------------------------------------------------------------------------------------------------------------------|--------|
| 31:1 | Reserved | Reserved                | -                                                                                                                     | -      |
| 0    | EIEN     | Error injection enabled | <ul style="list-style-type: none"> <li>0b0: Error injection disabled</li> <li>0b1: Error injection enabled</li> </ul> | RW     |

## 9.3.8 SRAM1 Error Injection Bit Enable Register 0 (SRAM1\_EIBIT0)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0x18   | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | EIBIT[31:0]  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access | RW           |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit  | Name        | Function                    | Description                                                                                                                                                                                                                                                                                                                                                                                                     | Access |
|------|-------------|-----------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:0 | EIBIT[31:0] | Error Injection Bit Enabled | Control whether bits 31-0 of the SRAM1 output data are inverted: <ul style="list-style-type: none"> <li>0b0: No inversion, no ECC error injection for this bit.</li> <li>0b1: Invert, inject ECC error for this bit.</li> </ul> <div>  <b>Note</b><br/>           Each bit corresponds to an output data bit.         </div> | RW     |

## 9.3.9 SRAM1 Error Injection Bit Enable Register 1 (SRAM1\_EIBIT1)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |              |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|--------------|---|---|---|---|---|---|---|
| 0x1C   | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7            | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |              |   |   |   |   |   |   |   |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   | EIBIT[38:32] |   |   |   |   |   |   |   |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   | RW           |   |   |   |   |   |   |   |

| Bit  | Name         | Function                    | Description                                                                                                                                                                                                                                                                                                                                                                                                     | Access |
|------|--------------|-----------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:7 | Reserved     | Reserved                    | -                                                                                                                                                                                                                                                                                                                                                                                                               | -      |
| 6:0  | EIBIT[38:32] | Error Injection Bit Enabled | Control whether bits 38-32 of the SRAM1 output data are inverted: <ul style="list-style-type: none"> <li>0b0: No inversion, no ECC error injection for this bit.</li> <li>0b1: Invert, inject ECC error for this bit.</li> </ul> <div>  <b>Note</b><br/>           Each bit corresponds to an output data bit.         </div> | RW     |

**Note**

Bits 38-32 of the SRAM1 output data are ECC bits.

## 9.3.10 SRAM1 Error Address Register (SRAM1\_ECCERRADDR)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |            |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|------------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0x20   | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15         | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |            |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | ECCERRADDR |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RO         |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit   | Name       | Function            | Description                                                       | Access |
|-------|------------|---------------------|-------------------------------------------------------------------|--------|
| 31:17 | Reserved   | Reserved            | -                                                                 | -      |
| 16:0  | ECCERRADDR | Check Error Address | Record the offset when SRAM1 generates a 1-bit or 2-bit ECC error | RO     |

## 9.3.11 SRAM2 Error Injection Enable Register (SRAM2\_EIEN)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |      |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|------|
| 0x24   | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0    |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |      |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   | EIEN |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   | RW   |

| Bit  | Name     | Function                | Description                                                                                                           | Access |
|------|----------|-------------------------|-----------------------------------------------------------------------------------------------------------------------|--------|
| 31:1 | Reserved | Reserved                | -                                                                                                                     | -      |
| 0    | EIEN     | Error Injection Enabled | <ul style="list-style-type: none"> <li>0b0: Error injection disabled</li> <li>0b1: Error injection enabled</li> </ul> | RW     |

## 9.3.12 SRAM2 Error Injection Bit Enable Register 0 (SRAM2\_EIBIT0)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0x28   | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | EIBIT[31:0]  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access | RW           |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit  | Name        | Function                    | Description                                                                                                                                                                                                                                                                                                                                                                                                     | Access |
|------|-------------|-----------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:0 | EIBIT[31:0] | Error Injection Bit Enabled | Control whether bits 31-0 of the SRAM2 output data are inverted: <ul style="list-style-type: none"> <li>0b0: No inversion, no ECC error injection for this bit.</li> <li>0b1: Invert, inject ECC error for this bit.</li> </ul> <div>  <b>Note</b><br/>           Each bit corresponds to an output data bit.         </div> | RW     |

## 9.3.13 SRAM2 Error Injection Bit Enable Register 1 (SRAM2\_EIBIT1)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |              |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|--------------|---|---|---|---|---|---|---|
| 0x2C   | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7            | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |              |   |   |   |   |   |   |   |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   | EIBIT[38:32] |   |   |   |   |   |   |   |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   | RW           |   |   |   |   |   |   |   |

| Bit  | Name         | Function                    | Description                                                                                                                                                                                                                                                                                                                                                                                  | Access |
|------|--------------|-----------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:7 | Reserved     | Reserved                    | -                                                                                                                                                                                                                                                                                                                                                                                            | -      |
| 6:0  | EIBIT[38:32] | Error Injection Bit Enabled | Control whether bits 38-32 of the SRAM2 output data are inverted: <ul style="list-style-type: none"> <li>0b0: No inversion, no ECC error injection for this bit.</li> <li>0b1: Invert, inject ECC error for this bit.</li> </ul> <div>  <b>Note</b><br/>Each bit corresponds to an output data bit. </div> | RW     |

**Note**

Bits 38-32 of the SRAM2 output data are ECC bits.

## 9.3.14 SRAM2 Error Address Register (SRAM2\_ECCERRADDR)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |            |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|------------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0x30   | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15         | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |            |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | ECCERRADDR |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RO         |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit   | Name       | Function            | Description                                                       | Access |
|-------|------------|---------------------|-------------------------------------------------------------------|--------|
| 31:17 | Reserved   | Reserved            | -                                                                 | -      |
| 16:0  | ECCERRADDR | Check Error Address | Record the offset when SRAM2 generates a 1-bit or 2-bit ECC error | RO     |

## 9.3.15 SRAM3 Error Injection Enable Register (SRAM3\_EIEN)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |      |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|------|
| 0x34   | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |      |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |      |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   | EIEN |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   | RW   |

| Bit  | Name     | Function                | Description                                                                                                           | Access |
|------|----------|-------------------------|-----------------------------------------------------------------------------------------------------------------------|--------|
| 31:1 | Reserved | Reserved                | -                                                                                                                     | -      |
| 0    | EIEN     | Error Injection Enabled | <ul style="list-style-type: none"> <li>0b0: Error injection disabled</li> <li>0b1: Error injection enabled</li> </ul> | RW     |

## 9.3.16 SRAM3 Error Injection Bit Enable Register 0 (SRAM3\_EIBIT0)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0x38   | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | EIBIT[31:0]  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access | RW           |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit  | Name        | Function                    | Description                                                                                                                                                                                                                                                                                                                                                                                                     | Access |
|------|-------------|-----------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:0 | EIBIT[31:0] | Error Injection Bit Enabled | Control whether bits 31-0 of the SRAM3 output data are inverted: <ul style="list-style-type: none"> <li>0b0: No inversion, no ECC error injection for this bit.</li> <li>0b1: Invert, inject ECC error for this bit.</li> </ul> <div>  <b>Note</b><br/>           Each bit corresponds to an output data bit.         </div> | RW     |



## 9.3.17 SRAM3 Error Injection Bit Enable Register 1 (SRAM3\_EIBIT1)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |              |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|--------------|---|---|---|---|---|---|---|
| 0x3C   | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7            | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |              |   |   |   |   |   |   |   |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   | EIBIT[38:32] |   |   |   |   |   |   |   |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   | RW           |   |   |   |   |   |   |   |

| Bit  | Name         | Function                    | Description                                                                                                                                                                                                                                                                                                                                                                                  | Access |
|------|--------------|-----------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:7 | Reserved     | Reserved                    | -                                                                                                                                                                                                                                                                                                                                                                                            | -      |
| 6:0  | EIBIT[38:32] | Error Injection Bit Enabled | Control whether bits 38-32 of the SRAM3 output data are inverted: <ul style="list-style-type: none"> <li>0b0: No inversion, no ECC error injection for this bit.</li> <li>0b1: Invert, inject ECC error for this bit.</li> </ul> <div>  <b>Note</b><br/>Each bit corresponds to an output data bit. </div> | RW     |

**Note**

Bits 38-32 of the SRAM3 output data are ECC bits.

## 9.3.18 SRAM3 Error Address Register (SRAM3\_ECCERRADDR)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |            |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|------------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0x40   | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15         | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |            |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | ECCERRADDR |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RO         |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit   | Name       | Function            | Description                                                       | Access |
|-------|------------|---------------------|-------------------------------------------------------------------|--------|
| 31:17 | Reserved   | Reserved            | -                                                                 | -      |
| 16:0  | ECCERRADDR | Check Error Address | Record the offset when SRAM3 generates a 1-bit or 2-bit ECC error | RO     |

## 9.3.19 SRAM4 Error Injection Enable Register (SRAM4\_EIEN)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |      |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|------|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0x44   | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14   | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |      |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | EIEN |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RW   |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit  | Name     | Function                | Description                                                                                                           | Access |
|------|----------|-------------------------|-----------------------------------------------------------------------------------------------------------------------|--------|
| 31:1 | Reserved | Reserved                | -                                                                                                                     | -      |
| 0    | EIEN     | Error Injection Enabled | <ul style="list-style-type: none"> <li>0b0: Error injection disabled</li> <li>0b1: Error injection enabled</li> </ul> | RW     |

## 9.3.20 SRAM4 Error Injection Bit Enable Register 0 (SRAM4\_EIBIT0)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0x48   | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | EIBIT[31:0]  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access | RW           |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit  | Name        | Function                    | Description                                                                                                                                                                                                                                                                                                                                                                                                     | Access |
|------|-------------|-----------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:0 | EIBIT[31:0] | Error Injection Bit Enabled | Control whether bits 31-0 of the SRAM4 output data are inverted: <ul style="list-style-type: none"> <li>0b0: No inversion, no ECC error injection for this bit.</li> <li>0b1: Invert, inject ECC error for this bit.</li> </ul> <div>  <b>Note</b><br/>           Each bit corresponds to an output data bit.         </div> | RW     |

## 9.3.21 SRAM4 Error Injection Bit Enable Register 1(SRAM4\_EIBIT1)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |              |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|--------------|---|---|---|---|---|---|---|
| 0x4C   | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7            | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |              |   |   |   |   |   |   |   |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   | EIBIT[38:32] |   |   |   |   |   |   |   |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   | RW           |   |   |   |   |   |   |   |

| Bit  | Name         | Function                    | Description                                                                                                                                                                                                                                                                                                                                                                                      | Access |
|------|--------------|-----------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:7 | Reserved     | Reserved                    | -                                                                                                                                                                                                                                                                                                                                                                                                | -      |
| 6:0  | EIBIT[38:32] | Error Injection Bit Enabled | Control whether bits 38-32 of the SRAM4 output data are inverted:<br><ul style="list-style-type: none"> <li>0b0: No inversion, no ECC error injection for this bit.</li> <li>0b1: Invert, inject ECC error for this bit.</li> </ul> <div>  <b>Note</b><br/> Each bit corresponds to an output data bit. </div> | RW     |

**Note**

Bits 38-32 of the SRAM4 output data are ECC bits.

## 9.3.22 SRAM4 Error Address Register (SRAM4\_ECCERRADDR)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |            |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|------------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0x50   | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15         | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |            |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | ECCERRADDR |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RO         |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit   | Name       | Function            | Description                                                       | Access |
|-------|------------|---------------------|-------------------------------------------------------------------|--------|
| 31:15 | Reserved   | Reserved            | -                                                                 | -      |
| 14:0  | ECCERRADDR | Check Error Address | Record the offset when SRAM4 generates a 1-bit or 2-bit ECC error | RO     |

## 9.3.23 SRAMH Error Injection Enable Register (SRAMH\_EIEN)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |  |  |  |  |      |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|--|--|--|--|--|------|
| 0x54   | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |  |  |  |  |  |      |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |  |  |  |  |      |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |  |  |  |  | EIEN |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |  |  |  |  | RW   |

| Bit  | Name     | Function                | Description                                                                                                           | Access |
|------|----------|-------------------------|-----------------------------------------------------------------------------------------------------------------------|--------|
| 31:1 | Reserved | Reserved                | -                                                                                                                     | -      |
| 0    | EIEN     | Error Injection Enabled | <ul style="list-style-type: none"> <li>0b0: Error injection disabled</li> <li>0b1: Error injection enabled</li> </ul> | RW     |

## 9.3.24 SRAMH Error Injection Bit Enable Register 0 (SRAMH\_EIBIT0)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |  |  |  |  |  |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|--|--|--|--|--|--|
| 0x58   | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |  |  |  |  |  |  |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |  |  |  |  |  |
| Name   | EIBIT[31:0]  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |  |  |  |  |  |
| Access | RW           |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |  |  |  |  |  |

| Bit  | Name        | Function                    | Description                                                                                                                                                                                                                                                                                                                                                                                                     | Access |
|------|-------------|-----------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:0 | EIBIT[31:0] | Error Injection Bit Enabled | Control whether bits 31-0 of the SRAMH output data are inverted: <ul style="list-style-type: none"> <li>0b0: No inversion, no ECC error injection for this bit.</li> <li>0b1: Invert, inject ECC error for this bit.</li> </ul> <div>  <b>Note</b><br/>           Each bit corresponds to an output data bit.         </div> | RW     |

## 9.3.25 SRAMH Error Injection Bit Enable Register 1 (SRAMH\_EIBIT0)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |              |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|--------------|---|---|---|---|---|---|---|
| 0x5C   | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7            | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |              |   |   |   |   |   |   |   |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   | EIBIT[38:32] |   |   |   |   |   |   |   |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   | RW           |   |   |   |   |   |   |   |

| Bit  | Name         | Function                    | Description                                                                                                                                                                                                                                                                                                                                                                                                     | Access |
|------|--------------|-----------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:7 | Reserved     | Reserved                    | -                                                                                                                                                                                                                                                                                                                                                                                                               | -      |
| 6:0  | EIBIT[38:32] | Error Injection Bit Enabled | Control whether bits 38-32 of the SRAMH output data are inverted: <ul style="list-style-type: none"> <li>0b0: No inversion, no ECC error injection for this bit.</li> <li>0b1: Invert, inject ECC error for this bit.</li> </ul> <div>  <b>Note</b><br/>           Each bit corresponds to an output data bit.         </div> | RW     |

**Note**

Bits 38-32 of the SRAMH output data are ECC bits.

## 9.3.26 SRAMH Error Address Register (SRAMH\_ECCERRADDR)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |            |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|------------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0x60   | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15         | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |            |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | ECCERRADDR |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|        |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |            |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RO         |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|        |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |            |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit   | Name       | Function            | Description                                                       | Access |
|-------|------------|---------------------|-------------------------------------------------------------------|--------|
| 31:17 | Reserved   | Reserved            | -                                                                 | -      |
| 16:0  | ECCERRADDR | Check Error Address | Record the offset when SRAMH generates a 1-bit or 2-bit ECC error | RO     |

## 9.3.27 Ret SRAM Error Injection Enable Register (SRAMB\_EIEN)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |      |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|------|
| 0x64   | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |      |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |      |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   | EIEN |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   | RW   |

| Bit  | Name     | Function                | Description                                                                                                           | Access |
|------|----------|-------------------------|-----------------------------------------------------------------------------------------------------------------------|--------|
| 31:1 | Reserved | Reserved                | -                                                                                                                     | -      |
| 0    | EIEN     | Error Injection Enabled | <ul style="list-style-type: none"> <li>0b0: Error injection disabled</li> <li>0b1: Error injection enabled</li> </ul> | RW     |

## 9.3.28 Ret SRAM Error Injection Bit Enable Register 0 (SRAMB\_EIBIT0)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0x68   | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | EIBIT[31:0]  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access | RW           |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit  | Name        | Function                    | Description                                                                                                                                                                                                                                                                                                                                                                                                        | Access |
|------|-------------|-----------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:0 | EIBIT[31:0] | Error Injection Bit Enabled | Control whether bits 31-0 of the Ret SRAM output data are inverted: <ul style="list-style-type: none"> <li>0b0: No inversion, no ECC error injection for this bit.</li> <li>0b1: Invert, inject ECC error for this bit.</li> </ul> <div>  <b>Note</b><br/>           Each bit corresponds to an output data bit.         </div> | RW     |

## 9.3.29 Ret SRAM Error Injection Bit Enable Register 1 (SRAMB\_EIBIT1)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |              |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|--------------|---|---|---|---|---|---|---|
| 0x6C   | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7            | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |              |   |   |   |   |   |   |   |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   | EIBIT[38:32] |   |   |   |   |   |   |   |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   | RW           |   |   |   |   |   |   |   |

| Bit  | Name         | Function                    | Description                                                                                                                                                                                                                                                                                                                                                                                     | Access |
|------|--------------|-----------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:7 | Reserved     | Reserved                    | -                                                                                                                                                                                                                                                                                                                                                                                               | -      |
| 6:0  | EIBIT[38:32] | Error Injection Bit Enabled | Control whether bits 38-32 of the Ret SRAM output data are inverted: <ul style="list-style-type: none"> <li>0b0: No inversion, no ECC error injection for this bit.</li> <li>0b1: Invert, inject ECC error for this bit.</li> </ul> <div>  <b>Note</b><br/>Each bit corresponds to an output data bit. </div> | RW     |

**Note**

Bits 38-32 of the Ret SRAM output data are ECC bits.

## 9.3.30 Ret SRAM Error Address Register (SRAMB\_ECCERRADDR)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |            |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|------------|----|---|---|---|---|---|---|---|---|---|---|
| 0x70   | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11         | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |            |    |   |   |   |   |   |   |   |   |   |   |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | ECCERRADDR |    |   |   |   |   |   |   |   |   |   |   |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RO         |    |   |   |   |   |   |   |   |   |   |   |

| Bit   | Name       | Function            | Description                                                          | Access |
|-------|------------|---------------------|----------------------------------------------------------------------|--------|
| 31:12 | Reserved   | Reserved            | -                                                                    | -      |
| 11:0  | ECCERRADDR | Check Error Address | Record the offset when Ret SRAM generates a 1-bit or 2-bit ECC error | RO     |

## 9.3.31 CACHERAM Error Injection Enable Register (CACHE\_EIEN)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |      |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|------|
| 0x74   | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |      |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |      |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   | EIEN |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   | RW   |

| Bit  | Name     | Function                | Description                                                                                                           | Access |
|------|----------|-------------------------|-----------------------------------------------------------------------------------------------------------------------|--------|
| 31:1 | Reserved | Reserved                | -                                                                                                                     | -      |
| 0    | EIEN     | Error Injection Enabled | <ul style="list-style-type: none"> <li>0b0: Error injection disabled</li> <li>0b1: Error injection enabled</li> </ul> | RW     |

## 9.3.32 CACHERAM Error Injection Bit Enable Register 0 (CACHE\_EIBIT0)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0x78   | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | EIBIT[31:0]  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access | RW           |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit  | Name        | Function                    | Description                                                                                                                                                                                                                                                                                                                                                                                                        | Access |
|------|-------------|-----------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:0 | EIBIT[31:0] | Error Injection Bit Enabled | Control whether bits 31-0 of the CACHERAM output data are inverted: <ul style="list-style-type: none"> <li>0b0: No inversion, no ECC error injection for this bit.</li> <li>0b1: Invert, inject ECC error for this bit.</li> </ul> <div>  <b>Note</b><br/>           Each bit corresponds to an output data bit.         </div> | RW     |



## 9.3.33 CACHERAM Error Injection Bit Enable Register 1 (CACHE\_EIBIT1)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0x7C   | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | EIBIT[63:32] |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access | RW           |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit  | Name         | Function                    | Description                                                                                                                                                                                                                                                                                                                                                                                                       | Access |
|------|--------------|-----------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:0 | EIBIT[63:32] | Error Injection Bit Enabled | Control whether bits 63-32 of the CACHERAM output data are inverted: <ul style="list-style-type: none"> <li>0b0: No inversion, no ECC error injection for this bit.</li> <li>0b1: Invert, inject ECC error for this bit.</li> </ul> <div>  <b>Note</b><br/>           Each bit corresponds to an output data bit.         </div> | RW     |

## 9.3.34 CACHERAM Error Injection Bit Enable Register 2 (CACHE\_EIBIT2)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0x80   | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | EIBIT[95:64] |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access | RW           |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit  | Name         | Function                    | Description                                                                                                                                                                                                                                                                                                                                                                                                         | Access |
|------|--------------|-----------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:0 | EIBIT[95:64] | Error Injection Bit Enabled | Control whether bits 95-64 of the CACHERAM output data are inverted: <ul style="list-style-type: none"> <li>0b0: No inversion, no ECC error injection for this bit.</li> <li>0b1: Invert, inject ECC error for this bit.</li> </ul> <div>  <b>Note</b><br/>           Each bit corresponds to an output data bit.         </div> | RW     |

## 9.3.35 CACHERAM Error Injection Bit Enable Register 3 (CACHE\_EIBIT3)

| Offset | Bit Position  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|---------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0x84   | 31            | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | EIBIT[127:96] |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access | RW            |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit  | Name          | Function                    | Description                                                                                                                                                                                                                                                                                                                                                                                                        | Access |
|------|---------------|-----------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:0 | EIBIT[127:96] | Error Injection Bit Enabled | Control whether bits 127-96 of the CACHERAM output data are inverted: <ul style="list-style-type: none"> <li>0b0: No inversion, no ECC error injection for this bit.</li> <li>0b1: Invert, inject ECC error for this bit.</li> </ul> <div>  <b>Note</b><br/>           Each bit corresponds to an output data bit.         </div> | RW     |

## 9.3.36 CACHERAM Error Injection Bit Enable Register 4 (CACHE\_EIBIT4)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |                |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----------------|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0x88   | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19             | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |                |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    | EIBIT[136:128] |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access |              |    |    |    |    |    |    |    |    |    |    |    | RW             |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit  | Name           | Function                    | Description                                                                                                                                                                                                                                                                                                                                                                                                           | Access |
|------|----------------|-----------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:9 | Reserved       | Reserved                    | -                                                                                                                                                                                                                                                                                                                                                                                                                     | -      |
| 8:0  | EIBIT[136:128] | Error Injection Bit Enabled | Control whether bits 136-128 of the CACHERAM output data are inverted: <ul style="list-style-type: none"> <li>0b0: No inversion, no ECC error injection for this bit.</li> <li>0b1: Invert, inject ECC error for this bit.</li> </ul> <div>  <b>Note</b><br/>           Each bit corresponds to an output data bit.         </div> | RW     |

**Note**

Bits 136-128 of the CACHERAM output data are ECC bits.

### 9.3.37 CACHERAM Error Address Register (CACHE\_ECCERRADDR)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |            |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|------------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0x8C   | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15         | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |            |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | ECCERRADDR |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |            |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit   | Name       | Function            | Description                                                          | Access |
|-------|------------|---------------------|----------------------------------------------------------------------|--------|
| 31:13 | Reserved   | Reserved            | -                                                                    | -      |
| 12:0  | ECCERRADDR | Check Error Address | Record the offset when CACHERAM generates a 1-bit or 2-bit ECC error | RO     |

## 9.4 Precautions

1. In the case where the RAM ECC check error is enabled to occur, the RAM space used must be initialized in units of words when accessing data. When executing instructions from the SRAMH space, the area of the used RAM space+3 words must be initialized in word units.
2. The VBAT domain needs to be initialized before using the backup SRAM. For the initialization method, refer to the relevant description in the "Battery Backup Power Domain" section of "Power Control (PWC)".
3. SRAM1 addresses 0x20000000, 0x20000001, and 0x20000002 do not support ECC checking. If these three addresses are accessed across regions, address 0x20000003 will not be accessed. In ECC mode, if these three addresses are accessed without cross-region access, address 0x20000003 will be accessed.
4. When the ECC mode is enabled, ECC calculation is performed on a word basis, so when accessing a byte or half-word, the SRAM controller will automatically access the corresponding word, and then read and write the byte or half-word through internal processing.
5. Because the CPU handles the CPU-I/CPU-D bus differently from the CPU-S bus, when a NOP instruction is inserted, the CPU-I/CPU-D bus and the CPU-S bus will produce different idle beats, and it is also related to the clock configuration. Normally, more than one NOP instruction must be inserted for the CPU-S bus to produce an idle beat, but only one NOP instruction needs to be inserted for the CPU-I/CPU-D bus to produce an idle beat. When error injection is enabled, when writing and reading the same address back-to-back, the write and read operations must be separated by at least 2 HCLK periods. The software can insert the interval by methods such as inserting NOP instructions. If the interval time is not enough, ECC errors may not be generated during the read operation.

## 10 General-Purpose I/O (GPIO)

### 10.1 Introduction

Some of the abbreviations used in this chapter are:

- Px (x=A to I) represents a group of ports, such as PA represents a group of 16 I/O ports PA0-PA15.
- Pxy (x=A-H, y=0-15; x=I, y=0-13, the same below) represents a single I/O port. such as the PB10 port represents the 11th I/O in the PB group.
- General Purpose Input Output (GPIO).
- NOD (Nmos Open Drain) NMOS open drain output mode.

### 10.2 Features

- Each port group has 16 I/O pins, which may be less than 16 depending on actual configuration
- Support pull-up input
- Support push-pull, open-drain output mode
- Support high, medium and low drive modes
- Supports free switching between CMOS/Schmitt input modes
- Support for external interrupt input
- Support I/O pin peripheral function multiplexing, one I/O pin can have up to 64 optional multiplexing functions
- Individual I/O pins can be programmed independently
- Each I/O pin can select 2 functions to be enable at the same time (does not support 2 output functions to be enable at the same time)

## 10.3 Functional Description

### 10.3.1 Function Block Diagram

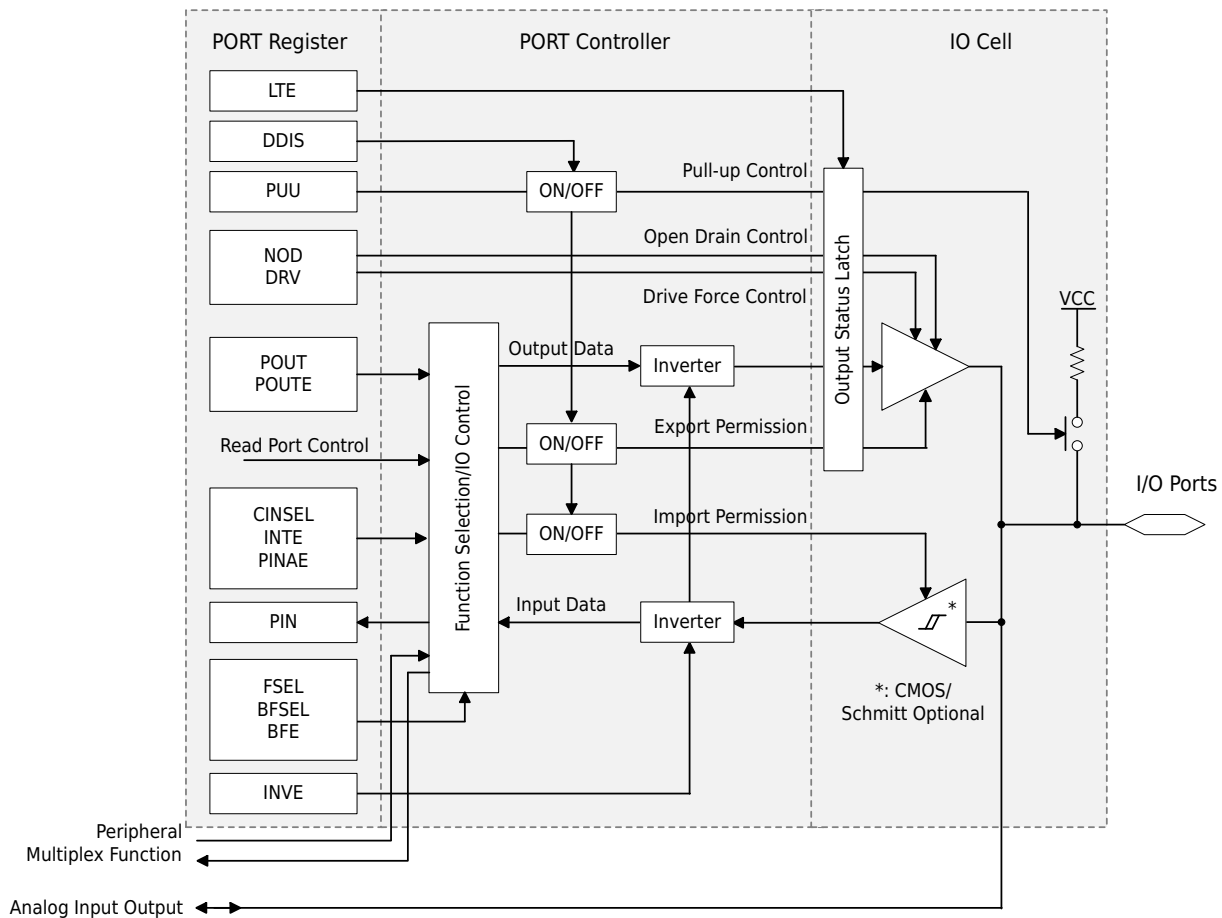


Figure 10-1 Port Basic Structure Diagram

For detailed GPIO port number, 5V tolerant, and drive capability configuration, please refer to the "Pin Definitions" chapter in the "Datasheet".

### 10.3.2 General Input/Output GPIO Function

#### General Input Function GPI

Each I/O has a general-purpose input GPI function, and when the digital function disable bit PCRxy.DDIS is 0, the GPI function is always enable, regardless of the setting value of PFSRxy.FSEL. The status of the current port can be obtained by accessing port input data register PIDRx. The corresponding single I/O port status can also be queried through the register PCRxy.PIN, and the PIDRx.PINy register bit is equivalent to the PCRxy.PIN bit.

By default, I/O input MOS is turned off to reduce power consumption. Only when PIDRx is read, PCRxy register is opened. If required, you can also set the register PINAER.PINAEx to 1 to keep the I/O input MOS in the open state.

When the system runs at a high-speed clock and PINAER is set to 0, the input status value may not be read correctly in a single period due to the delay of the I/O input. At this time, it is necessary to set the register PCCR.RDWT and insert several wait periods. For details, refer to the description of the PCCR register.

## General Output Function GPO

Except for input-specific ports, other I/O ports have general output GPO capabilities. The GPO function can be enabled by setting the port function selection register PFSRxy.FSEL to 0x0.

When the GPO function is active, the output value can be controlled by setting the generic output permission register POERx to enable or disable the output of I/O and the generic output data register PODRx. The output values of I/O can also be controlled using the following three registers: PORRx register, POSRx register, and POTRx register. Writing 1 to the corresponding bits in the register can make the corresponding I/O output 0, 1, invert. I/O output status does not change at write 0.

The registers above are operated by 16 PORT groups. For easy control of a single I/O, I/O output can also be allowed or disabled by setting PCRxy.POUTE, where the PCRxy.POUTE register bit is equivalent to POERx.POUTE<sub>y</sub>. The output value of I/O can be controlled by setting PCRxy.POUT, where the PCRxy.POUT register bit is equivalent to PODRx.POUT<sub>y</sub>. PCRxy is suitable for controlling a single PORT and POERx/PODRx is suitable for controlling an 16-bit set of PORTs.

After the system is reset, except for the JTAG multiplexing ports PA13, PA14, PA15, PB3, PB4, and the sub-oscillator multiplexing ports PC14 and PC15, the initial functions of other ports are GPO (PFSRxy.FSEL=0x0) and are in high-impedance state (output disabled POUTE=0).



### Note

Port PI13 is multiplexed with MD. When the reset port NRST is released, PI13/MD must be fixed to a low level so that the system can enter user mode. After the mode is established, PI13 can be used as a normal IO.

## 10.3.3 Peripheral Function

Through the register PFSRxy.FSEL, each port can configure up to 64 functions. These include the generic output GPO function corresponding to FSEL=0x0. For the specific functions of each port configuration, please refer to the "Pin Definitions-Pin Function Description" in the "Datasheet".

JTAG/SWD debug function, use the register PSPCR to select. PSPCR.SPFE[z], when z=0-4 is 1, the PFSRxy.FSEL bit of the corresponding port is disabled, that is, the priority of SPFE is higher than that of FSEL. The initial value of the PSPCR register is 0x1F, and the JTAG/SWD function is enable. If you want to set these ports as functions other than JTAG/SWD, you need to write 0 to the corresponding SPFE[z] bit first.

## 10.3.4 Dual Peripheral Function

In some applications, it is necessary to set a port to two functions at the same time. In this case, you can first select a function by PFSRxy.FSEL, then select the second function by setting the common control register PCCR.BFSEL, and set PFSRxy.BFE to 1. For example: set PFSRxy.FSEL=0x2, PCCR.BFSEL=0x5, PFSRxy.BFE=0x1, then function 2 and function 5 on Pxy will be enable at the same time. It is forbidden to enable 2 output functions on the same port at the same time.

## 10.3.5 Event Port Input and Output Functions

Support 4 groups of Event Ports, each with 16 ports. Event Port1 includes EVNTP100-EVNTP115, Event Port2 includes EVNTP200-EVNTP215, and so on. EVNTPmn (m=1-4, n=0-15) The port can be used as a trigger source to trigger other peripheral devices (such as TIMER, ADC, DMA, etc.) to start specific actions according to the event generated by the port input. It can also be used as a triggered object to accept events and automatically input or output.

As a trigger source, set PEVNTRISRm, PEVNTFALRm, PEVNTNFCR to select rising edge or falling edge detection, and digital filter function, and set the function selection register PFSRxy to select EVNTPmn function. When the selected edge is entered from the port, the event EVENT\_PORTm is generated and output to other peripheral devices to trigger its start action.

As the triggered object, set AOS Modules PEVNT\_TRGSEL12/PEVNT\_TRGSEL34 to select the trigger event source, and set PEVNTDIRRm to select the output or input function. When the output function is selected, EVNTPmn outputs the specified level or flips according to the setting values of PEVNTODRm, PEVNTORRm, and PEVNTOSRm when the selected event occurs. When the input function is selected, the EVNTPmn input state is saved into the register PEVNTIDRm when the selected event occurs.

When using the Event Port function, it is necessary to first set the automatic operation system AOS function enabled bit of the function clock control 0 register (PWC\_FCG0) to active.

### 10.3.6 External Interrupt EIRQ Input Function

Except for the PI13/MD pin, each I/O port has an external interrupt input function. When the PCRxy.INTE bit is set to 1, this I/O will be enabled as the external interrupt source EIRQy input (for example: PA0 corresponds to EIRQ0, PA2 corresponds to EIRQ2). Each EIRQy can be configured with more than one I/O. Please do not allow multiple I/O inputs simultaneously for each EIRQy. The EIRQy input function and the peripheral functions selected by PFSRxy.FSEL (including GPIO) can be enable at the same time.

When I/O port is used as external interrupt EIRQ, I/O port needs to combine with interrupt controller INTC, set filter, interrupt trigger edge, interrupt number and so on. For details, please refer to the relevant description in the "Interrupt Controller (INTC)" chapter.

### 10.3.7 Analog Function

Some I/O ports have analog input and output functions (including primary and secondary oscillators). When used as an analog function, please write 1 to the register PCRxy.DDIS to disable the digital function of the current port.

### 10.3.8 General Control

The general control functions described below are independent of the specific function selected for the port, i.e., the setting of PFSRxy.FSEL, unless otherwise specified.

#### Pull Up Resistor

Each I/O port has an internal pull-up resistor. The PCRxy.PUU bit of the register can be set to allow this function. When there is no input to the I/O port, the internal part is in a weak pull-up state. Pull-up is automatically disable when the I/O port is in the output state.



#### Note

When the I/O port selects the I2Cx\_SCL/I2Cx\_SDA function, the setting of the register PUU will be ignored, and the internal pull-up function will be forced to be disable.

The PA11/USBFS\_DM, PA12/USBFS\_DP, PB14/USBHS\_DM and PB15/USBHS\_DP pins are multiplexed with the USB port. All four pins have built-in pull-down resistors of approximately 400kΩ and are always enable.

#### Drive Capability Control

Each I/O port has 3 levels of high, medium and low drive capability adjustable, and the register PCRxy.DRV can be set as required.



This feature is only enable if the port is in the output state.

### Open-drain Output Mode

Set the PCRxy.NOD bit to set the I/O port to NMOS open-drain output mode. When NOD is active, the corresponding port can output 0 normally, and the port will be in high resistance state when output 1.

When the I/O port selects the I2Cx\_SCL/I2Cx\_SDA function, the setting of the register NOD will be ignored, and the open-drain output mode is forced to be enable.

### CMOS/Schmitt Input Mode

Each I/O port supports two input modes: CMOS and Schmitt. You can set the register PCRxy.CINSEL bit option, setting it to 0 means Schmitt input, and setting it to 1 means CMOS input. The default is Schmitt input.

When using the I/O input function, please set the input type according to actual needs.

## 10.4 Register Description

Except for Event Port related registers which only support 32-bit access, other registers support 16-bit and 32-bit access but not 8-bit access. These registers are combined as follows for 32-bit access:

Table 10-1 PORT Register List for 32-bit Access

| Address                  | Bit31 ... Bit16 | Bit15 ... Bit0 |
|--------------------------|-----------------|----------------|
| 0x40053800+0x10*n        | Reserved        | PIDRx          |
| 0x40053804+0x10*n        | POERx           | PODRx          |
| 0x40053808+0x10*n        | PORRx           | POSRx          |
| 0x4005380C+0x10*n        | Reserved        | POTRx          |
| 0x40053BF4               | Reserved        | PSPCR          |
| 0x40053BF8               | PINAER          | PCCR           |
| 0x40053BFC               | Reserved        | PWPR           |
| 0x40053C00+0x40*n+0x04*y | PFSRxy          | PCRxy          |



### Note

Unless otherwise specified, all register offset address calculation formulas in this section are:

- x=A-H corresponds to n=0-7, y=0-15.
- x=I corresponds to n=8, y=0-13.

### 10.4.1 Register List

Base address: 0x40053800

Table 10-2 General-Purpose IO (GPIO) Register Offset

| Offset             | Register | Description                               | Width |
|--------------------|----------|-------------------------------------------|-------|
| 0x00+0x10*n        | PIDRx    | General Input Data Register (x=A to I)    | 16    |
| 0x04+0x10*n        | PODRx    | General Output Data Register (x=A to I)   | 16    |
| 0x06+0x10*n        | POERx    | General Output Enable Register (x=A to I) | 16    |
| 0x08+0x10*n        | POSRx    | General Output Set Register (x=A to I)    | 16    |
| 0x0A+0x10*n        | PORRx    | General Output Reset Register (x=A to I)  | 16    |
| 0x0C+0x10*n        | POTRx    | General Output Toggle Register (x=A to I) | 16    |
| 0x3F4              | PSPCR    | Special Control Register                  | 16    |
| 0x3F8              | PCCR     | Common Control Register                   | 16    |
| 0x3FA              | PINAER   | Input Control Register                    | 16    |
| 0x3FC              | PWPR     | Write Protection Register                 | 16    |
| 0x400+0x40*n+0x4*y | PCRxy    | General Control Register (x=A to I)       | 16    |
| 0x402+0x40*n+0x4*y | PFSRxy   | Function Select Register (x=A to I)       | 16    |

Event Port base address: 0x40010800

Table 10-3 Event Port Register Offset

| Offset           | Register   | Description                                   | Width |
|------------------|------------|-----------------------------------------------|-------|
| 0x100+0x1C*(m-1) | PEVNTDIRRm | Event Port Direction Selection Register       | 32    |
| 0x104+0x1C*(m-1) | PEVNTIDRm  | Event Port Input Data Register                | 32    |
| 0x108+0x1C*(m-1) | PEVNTODRm  | Event Port Output Data Register               | 32    |
| 0x10C+0x1C*(m-1) | PEVNTORRm  | Event Port Output Data Reset Register         | 32    |
| 0x110+0x1C*(m-1) | PEVNTOSRm  | Event Port Output Data Set Register           | 32    |
| 0x114+0x1C*(m-1) | PEVNTRISRm | Event Port Rising Edge Input Enable Register  | 32    |
| 0x118+0x1C*(m-1) | PEVNTFALRm | Event Port Falling Edge Input Enable Register | 32    |
| 0x170            | PEVNTNFCR  | Event Port Input Filter Control Register      | 32    |



#### Note

In the offset of the Event Port register, m=1-4.

### 10.4.2 General Input Data Register (PIDRx)

Offset: 0x00+0x10\*n

| Offset | Bit Position |       |       |       |       |       |      |      |      |      |      |      |      |      |      |      |
|--------|--------------|-------|-------|-------|-------|-------|------|------|------|------|------|------|------|------|------|------|
| -      | 15           | 14    | 13    | 12    | 11    | 10    | 9    | 8    | 7    | 6    | 5    | 4    | 3    | 2    | 1    | 0    |
| Reset  | 0xFFFF       |       |       |       |       |       |      |      |      |      |      |      |      |      |      |      |
| Name   | PIN15        | PIN14 | PIN13 | PIN12 | PIN11 | PIN10 | PIN9 | PIN8 | PIN7 | PIN6 | PIN5 | PIN4 | PIN3 | PIN2 | PIN1 | PIN0 |
| Access | RO           | RO    | RO    | RO    | RO    | RO    | RO   | RO   | RO   | RO   | RO   | RO   | RO   | RO   | RO   | RO   |

| Bit  | Name             | Function     | Description                                                                                                                                                                                | Access |
|------|------------------|--------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 15:0 | PINy<br>(y=15-0) | Input status | Corresponding to Px15-Px0 ports respectively:<br><ul style="list-style-type: none"> <li>0b0: I/O port input state is low level</li> <li>0b1: I/O port input state is high level</li> </ul> | RO     |



#### Note

When the port digital function is not disabled (PCRxy.DDIS=0), the input status of the port can be obtained by reading this register, regardless of the setting value of the function selection register PFSRxy.FSEL.

There is no variable read value for the corresponding bit of the port.

When the port digital function is disabled (PCRxy.DDIS=1), since the I/O input MOS is in the off state, the corresponding PIN bit read value is a fixed value 0x1.

### 10.4.3 General Output Data Register (PODRx)

Offset: 0x04+0x10\*n

| Offset | Bit Position |        |        |        |        |        |       |       |       |       |       |       |       |       |       |       |
|--------|--------------|--------|--------|--------|--------|--------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| -      | 15           | 14     | 13     | 12     | 11     | 10     | 9     | 8     | 7     | 6     | 5     | 4     | 3     | 2     | 1     | 0     |
| Reset  | 0x0000       |        |        |        |        |        |       |       |       |       |       |       |       |       |       |       |
| Name   | POUT15       | POUT14 | POUT13 | POUT12 | POUT11 | POUT10 | POUT9 | POUT8 | POUT7 | POUT6 | POUT5 | POUT4 | POUT3 | POUT2 | POUT1 | POUT0 |
| Access | RW           | RW     | RW     | RW     | RW     | RW     | RW    | RW    | RW    | RW    | RW    | RW    | RW    | RW    | RW    | RW    |

| Bit  | Name              | Function    | Description                                                                                                                                              | Access |
|------|-------------------|-------------|----------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 15:0 | POUTy<br>(y=15-0) | Output Data | Corresponding to Px15-Px0 ports respectively:<br><ul style="list-style-type: none"> <li>0b0: Output low level</li> <li>0b1: Output high level</li> </ul> | RW     |



#### Note

Override this register to change the output status of the corresponding port when the I/O port is set to GPO.

### 10.4.4 General Output Enable Register (POERx)

Offset: 0x06+0x10\*n

| Offset | Bit Position |         |         |         |         |         |        |        |        |        |        |        |        |        |        |        |
|--------|--------------|---------|---------|---------|---------|---------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|
| -      | 15           | 14      | 13      | 12      | 11      | 10      | 9      | 8      | 7      | 6      | 5      | 4      | 3      | 2      | 1      | 0      |
| Reset  | 0x0000       |         |         |         |         |         |        |        |        |        |        |        |        |        |        |        |
| Name   | POUTE15      | POUTE14 | POUTE13 | POUTE12 | POUTE11 | POUTE10 | POUTE9 | POUTE8 | POUTE7 | POUTE6 | POUTE5 | POUTE4 | POUTE3 | POUTE2 | POUTE1 | POUTE0 |
| Access | RW           | RW      | RW      | RW      | RW      | RW      | RW     | RW     | RW     | RW     | RW     | RW     | RW     | RW     | RW     | RW     |

| Bit  | Name                           | Function      | Description                                                                                                                                          | Access |
|------|--------------------------------|---------------|------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 15:0 | POUTE <sub>y</sub><br>(y=15-0) | Output enable | Corresponding to Px15-Px0 ports respectively:<br><ul style="list-style-type: none"> <li>0b0: Output disabled</li> <li>0b1: Output enabled</li> </ul> | RW     |



#### Note

When the I/O port is set to GPO and this register is set to 1, the PODRx setting is output to the corresponding I/O port. When this register is set to 0, the output is closed and the port is in high resistance state.

It is forbidden to set the corresponding bit of a non-existent port to 1.

### 10.4.5 General Output Set Register (POSRx)

Offset: 0x08+0x10\*n

| Offset | Bit Position |       |       |       |       |       |      |      |      |      |      |      |      |      |      |      |
|--------|--------------|-------|-------|-------|-------|-------|------|------|------|------|------|------|------|------|------|------|
| -      | 15           | 14    | 13    | 12    | 11    | 10    | 9    | 8    | 7    | 6    | 5    | 4    | 3    | 2    | 1    | 0    |
| Reset  | 0x0000       |       |       |       |       |       |      |      |      |      |      |      |      |      |      |      |
| Name   | POS15        | POS14 | POS13 | POS12 | POS11 | POS10 | POS9 | POS8 | POS7 | POS6 | POS5 | POS4 | POS3 | POS2 | POS1 | POS0 |
| Access | WO           | WO    | WO    | WO    | WO    | WO    | WO   | WO   | WO   | WO   | WO   | WO   | WO   | WO   | WO   | WO   |

| Bit  | Name                         | Function      | Description                                                                                                                                                                                                                         | Access |
|------|------------------------------|---------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 15:0 | POS <sub>y</sub><br>(y=15-0) | Output Height | Corresponding to Px15-Px0 ports respectively:<br><ul style="list-style-type: none"> <li>0b0: The corresponding PODRx.POUT<sub>y</sub> does not change</li> <li>0b1: The corresponding PODRx.POUT<sub>y</sub> is set to 1</li> </ul> | WO     |



#### Note

The read value for this register is always 0. In 32-bit access, when POR<sub>y</sub> and POS<sub>y</sub> of the same I/O are written to 1 at the same time, POR<sub>y</sub> has a higher priority, that is, the corresponding POUT<sub>y</sub> is cleared.

### 10.4.6 General Output Reset Register (PORRx)

Offset: 0x0A+0x10\*n

| Offset | Bit Position |       |       |       |       |       |      |      |      |      |      |      |      |      |      |      |
|--------|--------------|-------|-------|-------|-------|-------|------|------|------|------|------|------|------|------|------|------|
| -      | 15           | 14    | 13    | 12    | 11    | 10    | 9    | 8    | 7    | 6    | 5    | 4    | 3    | 2    | 1    | 0    |
| Reset  | 0x0000       |       |       |       |       |       |      |      |      |      |      |      |      |      |      |      |
| Name   | POR15        | POR14 | POR13 | POR12 | POR11 | POR10 | POR9 | POR8 | POR7 | POR6 | POR5 | POR4 | POR3 | POR2 | POR1 | POR0 |
| Access | WO           | WO    | WO    | WO    | WO    | WO    | WO   | WO   | WO   | WO   | WO   | WO   | WO   | WO   | WO   | WO   |

| Bit  | Name             | Function   | Description                                                                                                                                                                                               | Access |
|------|------------------|------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 15:0 | PORy<br>(y=15-0) | Output Low | Corresponding to Px15-Px0 ports respectively: <ul style="list-style-type: none"> <li>0b0: The corresponding PODRx.POUTy does not change</li> <li>0b1: The corresponding PODRx.POUTy is cleared</li> </ul> | WO     |



#### Note

The read value for this register is always 0.

### 10.4.7 General Output Toggle Register (POTRx)

Offset: 0x0C+0x10\*n

| Offset | Bit Position |       |       |       |       |       |      |      |      |      |      |      |      |      |      |      |
|--------|--------------|-------|-------|-------|-------|-------|------|------|------|------|------|------|------|------|------|------|
| -      | 15           | 14    | 13    | 12    | 11    | 10    | 9    | 8    | 7    | 6    | 5    | 4    | 3    | 2    | 1    | 0    |
| Reset  | 0x0000       |       |       |       |       |       |      |      |      |      |      |      |      |      |      |      |
| Name   | POT15        | POT14 | POT13 | POT12 | POT11 | POT10 | POT9 | POT8 | POT7 | POT6 | POT5 | POT4 | POT3 | POT2 | POT1 | POT0 |
| Access | WO           | WO    | WO    | WO    | WO    | WO    | WO   | WO   | WO   | WO   | WO   | WO   | WO   | WO   | WO   | WO   |

| Bit  | Name             | Function         | Description                                                                                                                                                                                                | Access |
|------|------------------|------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 15:0 | POTy<br>(y=15-0) | Output Inversion | Corresponding to Px15-Px0 ports respectively: <ul style="list-style-type: none"> <li>0b0: The corresponding PODRx.POUTy does not change</li> <li>0b1: The corresponding PODRx.POUTy is inverted</li> </ul> | WO     |



#### Note

The read value for this register is always 0.

## 10.4.8 Special Control Register (PSPCR)

| Offset | Bit Position |    |    |    |    |    |   |   |   |   |   |           |   |   |   |   |
|--------|--------------|----|----|----|----|----|---|---|---|---|---|-----------|---|---|---|---|
| 0x3F4  | 15           | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4         | 3 | 2 | 1 | 0 |
| Reset  | 0x001F       |    |    |    |    |    |   |   |   |   |   |           |   |   |   |   |
| Name   | Reserved     |    |    |    |    |    |   |   |   |   |   | SPFE[4:0] |   |   |   |   |
| Access |              |    |    |    |    |    |   |   |   |   |   | RW        |   |   |   |   |

| Bit  | Name     | Function                   | Description                                                                                                                         | Access |
|------|----------|----------------------------|-------------------------------------------------------------------------------------------------------------------------------------|--------|
| 15:5 | Reserved | Reserved                   | -                                                                                                                                   | -      |
| 4    | SPFE[4]  | Special function selection | <ul style="list-style-type: none"> <li>0b0: NJTRST function disabled</li> <li>0b1: NJTRST function enabled</li> </ul>               | RW     |
| 3    | SPFE[3]  | Special function selection | <ul style="list-style-type: none"> <li>0b0: JTDI function disabled</li> <li>0b1: JTDI function enabled</li> </ul>                   | RW     |
| 2    | SPFE[2]  | Special function selection | <ul style="list-style-type: none"> <li>0b0: JTDO_TRACESWO function disabled</li> <li>0b1: JTDO_TRACESWO function enabled</li> </ul> | RW     |
| 1    | SPFE[1]  | Special function selection | <ul style="list-style-type: none"> <li>0b0: JTMS_SWDIO function disabled</li> <li>0b1: JTMS_SWDIO function enabled</li> </ul>       | RW     |
| 0    | SPFE[0]  | Special function selection | <ul style="list-style-type: none"> <li>0b0: JTCK_SWCLK function disabled</li> <li>0b1: JTCK_SWCLK function enabled</li> </ul>       | RW     |

**Note**

The PSPCR.SPFE have higher priority than the PFSRxy.FSEL.

## 10.4.9 Common Control Register (PCCR)

| Offset | Bit Position |      |    |    |    |          |   |   |   |   |   |       |   |   |   |   |
|--------|--------------|------|----|----|----|----------|---|---|---|---|---|-------|---|---|---|---|
| 0x3F8  | 15           | 14   | 13 | 12 | 11 | 10       | 9 | 8 | 7 | 6 | 5 | 4     | 3 | 2 | 1 | 0 |
| Reset  | 0x1000       |      |    |    |    |          |   |   |   |   |   |       |   |   |   |   |
| Name   | Reserved     | RDWT |    |    |    | Reserved |   |   |   |   |   | BFSEL |   |   |   |   |
| Access |              | RW   |    |    |    |          |   |   |   |   |   | RW    |   |   |   |   |

| Bit                 | Name                   | Function                     | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | Access              |                        |               |              |          |     |                   |          |     |                    |           |     |                    |           |     |                    |           |     |    |
|---------------------|------------------------|------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------|------------------------|---------------|--------------|----------|-----|-------------------|----------|-----|--------------------|-----------|-----|--------------------|-----------|-----|--------------------|-----------|-----|----|
| 15                  | Reserved               | Reserved                     | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | -                   |                        |               |              |          |     |                   |          |     |                    |           |     |                    |           |     |                    |           |     |    |
| 14:12               | RDWT                   | Read Port Waiting            | <div>Set the number of wait periods inserted when reading registers PIDRx and PCRxy</div> <table><tr><th>Operating Frequency</th><th>Minimum Waiting Period</th><th>Setting Value</th></tr><tr><td>0&lt;HCLK≤50MHz</td><td>0 period</td><td>0-7</td></tr><tr><td>50MHz&lt;HCLK≤100MHz</td><td>1 period</td><td>1-7</td></tr><tr><td>100MHz&lt;HCLK≤150MHz</td><td>2 periods</td><td>2-7</td></tr><tr><td>150MHz&lt;HCLK≤200MHz</td><td>3 periods</td><td>3-7</td></tr><tr><td>200MHz&lt;HCLK≤240MHz</td><td>4 periods</td><td>4-7</td></tr></table> | Operating Frequency | Minimum Waiting Period | Setting Value | 0<HCLK≤50MHz | 0 period | 0-7 | 50MHz<HCLK≤100MHz | 1 period | 1-7 | 100MHz<HCLK≤150MHz | 2 periods | 2-7 | 150MHz<HCLK≤200MHz | 3 periods | 3-7 | 200MHz<HCLK≤240MHz | 4 periods | 4-7 | RW |
| Operating Frequency | Minimum Waiting Period | Setting Value                |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |                     |                        |               |              |          |     |                   |          |     |                    |           |     |                    |           |     |                    |           |     |    |
| 0<HCLK≤50MHz        | 0 period               | 0-7                          |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |                     |                        |               |              |          |     |                   |          |     |                    |           |     |                    |           |     |                    |           |     |    |
| 50MHz<HCLK≤100MHz   | 1 period               | 1-7                          |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |                     |                        |               |              |          |     |                   |          |     |                    |           |     |                    |           |     |                    |           |     |    |
| 100MHz<HCLK≤150MHz  | 2 periods              | 2-7                          |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |                     |                        |               |              |          |     |                   |          |     |                    |           |     |                    |           |     |                    |           |     |    |
| 150MHz<HCLK≤200MHz  | 3 periods              | 3-7                          |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |                     |                        |               |              |          |     |                   |          |     |                    |           |     |                    |           |     |                    |           |     |    |
| 200MHz<HCLK≤240MHz  | 4 periods              | 4-7                          |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |                     |                        |               |              |          |     |                   |          |     |                    |           |     |                    |           |     |                    |           |     |    |
| 11:6                | Reserved               | Reserved                     | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | -                   |                        |               |              |          |     |                   |          |     |                    |           |     |                    |           |     |                    |           |     |    |
| 5:0                 | BFSEL                  | Secondary Function Selection | For the function configuration of each port, refer to the "Pin Definitions-Pin Function Description" section in the Datasheet.                                                                                                                                                                                                                                                                                                                                                                                                                      | RW                  |                        |               |              |          |     |                   |          |     |                    |           |     |                    |           |     |                    |           |     |    |



## 10.4.10 Input Control Register (PINAER)

| Offset | Bit Position |    |    |    |    |    |   |   |        |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|---|---|--------|---|---|---|---|---|---|---|
| 0x3FA  | 15           | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7      | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x0000       |    |    |    |    |    |   |   |        |   |   |   |   |   |   |   |
| Name   | Reserved     |    |    |    |    |    |   |   | PINAER |   |   |   |   |   |   |   |
| Access |              |    |    |    |    |    |   |   | RW     |   |   |   |   |   |   |   |

| Bit  | Name     | Function            | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | Access |
|------|----------|---------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 15:9 | Reserved | Reserved            | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | -      |
| 8:0  | PINAER   | Input Normally-Open | <ul style="list-style-type: none"> <li>0b0: Input MOS normally-open disabled</li> <li>0b1: Input MOS normally-open enabled</li> </ul> <p>The correspondence between each bit of PINAER and the I/O port is:</p> <ul style="list-style-type: none"> <li>PINAER[0] Control PA0-PA15</li> <li>PINAER[1] Control PB0-PB15</li> <li>PINAER[2] Control PC0-PC15</li> <li>PINAER[3] Control PD0-PD15</li> <li>PINAER[4] Control PE0-PE15</li> <li>PINAER[5] Control PF0-PF15</li> <li>PINAER[6] Control PG0-PG15</li> <li>PINAER[7] Control PH0-PH15</li> <li>PINAER[8] Control PI0-PI13</li> </ul> | RW     |

## 10.4.11 Write Protection Register (PWPR)

| Offset | Bit Position |    |    |    |    |    |   |   |          |   |   |   |   |   |   |   |    |
|--------|--------------|----|----|----|----|----|---|---|----------|---|---|---|---|---|---|---|----|
| 0x3FC  | 15           | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7        | 6 | 5 | 4 | 3 | 2 | 1 | 0 |    |
| Reset  | 0x0000       |    |    |    |    |    |   |   |          |   |   |   |   |   |   |   |    |
| Name   | WP           |    |    |    |    |    |   |   | Reserved |   |   |   |   |   |   |   | WE |
| Access | WO           |    |    |    |    |    |   |   |          |   |   |   |   |   |   |   | RW |

| Bit  | Name            | Function              | Description                                                                                                                                                               | Access |
|------|-----------------|-----------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 15:8 | WP              | Write protection code | When WE needs to be set to 1, 0xA5 must be written to WP at the same time; read always returns 0                                                                          | WO     |
| 7:1  | <i>Reserved</i> | <i>Reserved</i>       | -                                                                                                                                                                         | -      |
| 0    | WE              | Write enable          | <ul style="list-style-type: none"> <li>0b0: PSPCR, PCCR, PCRxy, PFSRxy register write disabled</li> <li>0b1: PSPCR, PCCR, PCRxy, PFSRxy register write enabled</li> </ul> | RW     |

**Note**

Writing 0xA501 to this register will set WE to 1; writing other values will clear WE to 0.

## 10.4.12 General Control Register (PCRxy)

Offset: 0x400+0x40\*n+0x4\*y

| Offset | Bit Position          |     |          |      |          |        |      |     |          |     |     |          |     |       |      |    |
|--------|-----------------------|-----|----------|------|----------|--------|------|-----|----------|-----|-----|----------|-----|-------|------|----|
| -      | 15                    | 14  | 13       | 12   | 11       | 10     | 9    | 8   | 7        | 6   | 5   | 4        | 3   | 2     | 1    | 0  |
| Reset  | 0x0X00 <sup>(1)</sup> |     |          |      |          |        |      |     |          |     |     |          |     |       |      |    |
| Name   | DDIS                  | LTE | Reserved | INTE | Reserved | CINSEL | INVE | PIN | Reserved | PUU | DRV | Reserved | NOD | POUTE | POUT |    |
| Access | RW                    | RW  |          | RW   |          | RW     | RW   | RO  |          | RW  | RW  |          | RW  | RW    | RW   | RW |

| Bit | Name     | Function                   | Description                                                                                                                                                                                              | Access |
|-----|----------|----------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 15  | DDIS     | Digital function disabled  | <ul style="list-style-type: none"> <li>0b0: Digital function enabled</li> <li>0b1: Digital function disabled</li> </ul>                                                                                  | RW     |
| 14  | LTE      | Output status latch        | <ul style="list-style-type: none"> <li>0b0: Output latch disabled</li> <li>0b1: Output latch enabled</li> </ul>                                                                                          | RW     |
| 13  | Reserved | Reserved                   | -                                                                                                                                                                                                        | -      |
| 12  | INTE     | External interrupt enabled | <ul style="list-style-type: none"> <li>0b0: External interrupt input disabled</li> <li>0b1: External interrupt input enabled</li> </ul>                                                                  | RW     |
| 11  | Reserved | Reserved                   | -                                                                                                                                                                                                        | -      |
| 10  | CINSEL   | Input modes election       | <ul style="list-style-type: none"> <li>0b0: Schmitt input</li> <li>0b1: CMOS input</li> </ul>                                                                                                            | RW     |
| 9   | INVE     | Inverse enabled            | <ul style="list-style-type: none"> <li>0b0: Input and output data not inverted</li> <li>0b1: Input and output data inverted</li> </ul>                                                                   | RW     |
| 8   | PIN      | Input status               | <ul style="list-style-type: none"> <li>0b0: I/O port input state is low level</li> <li>0b1: I/O port input state is high level</li> </ul> <p>The function is consistent with the register PIDRx.PINy</p> | RO     |
| 7   | Reserved | Reserved                   | -                                                                                                                                                                                                        | -      |
| 6   | PUU      | Pull-up enabled            | <ul style="list-style-type: none"> <li>0b0: Internal pullup resistor disabled</li> <li>0b1: Internal pullup resistor enabled</li> </ul>                                                                  | RW     |
| 5:4 | DRV      | Drive mode selection       | <ul style="list-style-type: none"> <li>0b00: Low drive force mode</li> <li>0b01: Medium drive force mode</li> <li>0b10/0b11: High drive force mode</li> </ul>                                            | RW     |
| 3   | Reserved | Reserved                   | -                                                                                                                                                                                                        | -      |
| 2   | NOD      | NMOS open-drain            | <ul style="list-style-type: none"> <li>0b0: Normal CMOS output mode</li> <li>0b1: NMOS open-drain output mode</li> </ul>                                                                                 | RW     |

| Bit | Name  | Function          | Description                                                                                                                                                             | Access |
|-----|-------|-------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 1   | POUTE | Output permission | <ul style="list-style-type: none"> <li>● 0b0: Output disabled</li> <li>● 0b1: Output enabled</li> </ul> The function is consistent with the register PO-<br>ERx.POUTEy  | RW     |
| 0   | POUT  | Output data       | <ul style="list-style-type: none"> <li>● 0b0: Output low level</li> <li>● 0b1: Output high level</li> </ul> The function is consistent with the register<br>PODRx.POUTy | RW     |

**Note**

1. The reset value of the external low-speed oscillator XTAL32 (XTAL32\_IN, XTAL32\_OUT) multiplexing ports PCRC14 and PCRC15 is 0x8100.
2. When DDIS is set to 1, all digital functions of the corresponding port are forcibly disabled, including general-purpose input and output, peripheral digital input and output, pull-up/pull-down functions, and external interrupt input functions. When the port is used as an analog input, please set the DDIS bit to 1.
3. When LTE is set to 1 Output Lock active, the current output status of the port remains until LTE is written to 0. This function is mainly used when the port function is switched. To avoid system malfunction caused by unexpected burrs in port output during function switching, before the function switching, the output status of the port is latched in LTE write 1, and then the register switch function is selected. Finally, the LTE write 0 is unlatched and the port status is updated to a new function.
4. When INVE is set to 1, the input and output data of the port will be inverted, including the GPIO function and other peripheral input and output functions.
5. CINSEL is used to set the I/O input mode and affects all digital input functions such as GPI, EIRQ and peripheral inputs.

### 10.4.13 Function Select Register (PFSRxy)

Offset: 0x402+0x40\*n+0x4\*y

| Offset | Bit Position |    |    |    |    |    |   |     |          |      |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|---|-----|----------|------|---|---|---|---|---|---|
| -      | 15           | 14 | 13 | 12 | 11 | 10 | 9 | 8   | 7        | 6    | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x0000       |    |    |    |    |    |   |     |          |      |   |   |   |   |   |   |
| Name   | Reserved     |    |    |    |    |    |   | BFE | Reserved | FSEL |   |   |   |   |   |   |
| Access |              |    |    |    |    |    |   | RW  |          | RW   |   |   |   |   |   |   |

| Bit  | Name     | Function                   | Description                                                                                                                                                                                          | Access |
|------|----------|----------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 15:9 | Reserved | Reserved                   | -                                                                                                                                                                                                    | -      |
| 8    | BFE      | Secondary function enabled | Control whether the secondary function selected by PCCR.BFSEL is enabled <ul style="list-style-type: none"> <li>0b0: Secondary function disabled</li> <li>0b1: Secondary function enabled</li> </ul> | RW     |
| 7:6  | Reserved | Reserved                   | -                                                                                                                                                                                                    | -      |
| 5:0  | FSEL     | Function selection         | For the function configuration of each port, refer to the "Pin Definitions-Pin Function Description" in Datasheet.                                                                                   | RW     |



#### Note

- I/O port can select one of several features configured on the port via FSEL. Setting FSEL to 0 means selecting Func0, setting it to 1 means selecting Func1, and so on. Setting it to 15 means selecting Func15. The general output function GPO corresponding to Func0.
- The initial state of PA13, PA14, PA15, PB3, and PB4 ports is that the JTAG/SWD function is enable after reset. When configuring the FSEL selection function, it is necessary to write 0 to the corresponding bit of the register PSPCR to inactiveate the JTAG/SWD function. After reset, the PC14 and PC15 ports serve as interfaces for the external low-speed oscillator, and they are initially in a state where digital functions are disabled. To select digital functions, first, set the register CMU\_XTAL32CR.XTAL32STP to 1 to turn off the external low-speed oscillator, and then write 0 to the corresponding register PCRxy.DDIS bit to enable the digital functions.

### 10.4.14 Event Port Direction Selection Register (PEVNTDIRRm)

Offset:  $0x100+0x1C*(m-1)$

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |            |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|------------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| -      | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15         | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |            |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | PDIR[15:0] |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RW         |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit   | Name       | Function            | Description                                                                                                               | Access |
|-------|------------|---------------------|---------------------------------------------------------------------------------------------------------------------------|--------|
| 31:16 | Reserved   | Reserved            | -                                                                                                                         | -      |
| 15:0  | PDIR[15:0] | Direction Selection | <ul style="list-style-type: none"> <li>0b0: Event Port Input Function</li> <li>0b1: Event Port Output Function</li> </ul> | RW     |

### 10.4.15 Event Port Input Data Register (PEVNTIDRm)

Offset:  $0x104+0x1C*(m-1)$

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |           |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-----------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| -      | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15        | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |           |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | PIN[15:0] |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RO        |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit   | Name      | Function         | Description                                                                                                                                                                                           | Access |
|-------|-----------|------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:16 | Reserved  | Reserved         | -                                                                                                                                                                                                     | -      |
| 15:0  | PIN[15:0] | Port Input State | <ul style="list-style-type: none"> <li>0b0: Event Port input state is low level when the event is triggered</li> <li>0b1: Event Port input state is high level when the event is triggered</li> </ul> | RO     |



#### Note

When the direction of the Event Port is set to the input state, when the set event is triggered, the input state of the corresponding I/O port is saved to this register.

### 10.4.16 Event Port Output Data Register (PEVNTODRm)

Offset: 0x108+0x1C\*(m-1)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |            |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |  |  |  |  |  |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|------------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|--|--|--|--|--|--|
| -      | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15         | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |  |  |  |  |  |  |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |            |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |  |  |  |  |  |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | POUT[15:0] |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |  |  |  |  |  |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RW         |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |  |  |  |  |  |

| Bit   | Name       | Function          | Description                                                                                                                   | Access |
|-------|------------|-------------------|-------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:16 | Reserved   | Reserved          | -                                                                                                                             | -      |
| 15:0  | POUT[15:0] | Port Output Value | <ul style="list-style-type: none"> <li>0b0: Event Port output low level</li> <li>0b1: Event Port output high level</li> </ul> | RW     |



#### Note

When the direction of the Event Port is set to output, the Event Port output value is the write value of the corresponding POUT bit before the set event is triggered. When the set event is triggered, according to the set values of PEVNTORRm and PEVNTOSRm, the corresponding bit of PEVNTODRm.POUT is cleared to 0, set to 1, or flips, and output to the EVNTPmn port at the same time.

### 10.4.17 Event Port Output Data Reset Register (PEVNTORRm)

Offset: 0x10C+0x1C\*(m-1)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |           |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |  |  |  |  |  |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-----------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|--|--|--|--|--|--|
| -      | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15        | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |  |  |  |  |  |  |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |           |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |  |  |  |  |  |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | POR[15:0] |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |  |  |  |  |  |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RW        |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |  |  |  |  |  |

| Bit   | Name      | Function           | Description                                                                                                                                                                                                             | Access |
|-------|-----------|--------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:16 | Reserved  | Reserved           | -                                                                                                                                                                                                                       | -      |
| 15:0  | POR[15:0] | Output Value Reset | <ul style="list-style-type: none"> <li>0b0: The corresponding PEVNTODRm.POUT does not change when the event is triggered</li> <li>0b1: The corresponding PEVNTODRm.POUT is reset when the event is triggered</li> </ul> | RW     |

**Note**

When PEVNTORRm.POR and PEVNTOSRm.POS are both set to 1, the corresponding PEVNTODRm.POUT flips when the event is triggered.



### 10.4.18 Event Port Output Data Set Register (PEVNTOSRm)

Offset: 0x110+0x1C\*(m-1)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |           |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-----------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| '      | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15        | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |           |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | POS[15:0] |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RW        |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit   | Name      | Function         | Description                                                                                                                                                                                                            | Access |
|-------|-----------|------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:16 | Reserved  | Reserved         | -                                                                                                                                                                                                                      | -      |
| 15:0  | POS[15:0] | Output Value Set | <ul style="list-style-type: none"> <li>0b0: The corresponding PEVNTODRm.POUT does not change when the event is triggered</li> <li>0b1: When the event is triggered, the corresponding PEVNTODRm.POUT is set</li> </ul> | RW     |



#### Note

When PEVNTORRm.POR and PEVNTOSRm.POS are both set to 1, the corresponding PEVNTODRm.POUT flips when the event is triggered.

10.4.19 Event Port Rising Edge Input Enable Register (PEVNTRISRm)

Offset: 0x114+0x1C\*(m-1)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |           |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-----------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| '      | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15        | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |           |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RIS[15:0] |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RW        |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit   | Name      | Function                          | Description                                                                                                                                                                                         | Access |
|-------|-----------|-----------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:16 | Reserved  | Reserved                          | -                                                                                                                                                                                                   | -      |
| 15:0  | RIS[15:0] | Rising Edge Detection Per-mission | <ul style="list-style-type: none"><li>0b0: EVNTPmn rising edge event detection disabled</li><li>0b1: EVNTPmn rising edge event detection enabled</li></ul> PEVNTRISRm.RIS[n] corresponds to EVNTPmn | RW     |

 **Note**

Event Port is used as an event source. When the RIS bit is set to 1, the event is output corresponding to the rising edge of the EVNTPmn input to trigger other peripherals. The edge events of EVNTPm0-15 are combined into one event EVENT\_PORTm output, and any one of the ports will output the event EVENT\_PORTm after detecting the edge.

### 10.4.20 Event Port Falling Edge Input Enable Register (PEVNTFALRm)

Offset:  $0x118 + 0x1C * (m-1)$

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |           |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-----------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| '      | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15        | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |           |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | FAL[15:0] |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RW        |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit   | Name      | Function                          | Description                                                                                                                                                                                              | Access |
|-------|-----------|-----------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:16 | Reserved  | Reserved                          | -                                                                                                                                                                                                        | -      |
| 15:0  | FAL[15:0] | Falling Edge Detection Permission | <ul style="list-style-type: none"> <li>0b0: EVNTPmn falling edge event detection disabled</li> <li>0b1: EVNTPmn falling edge event detection enabled</li> </ul> PEVNTRISRm.FAL[n] corresponds to EVNTPmn | RW     |



#### Note

Event Port is used as an event source. When the FAL bit is set to 1, the event is output corresponding to the falling edge of the EVNTP input to trigger other peripherals. The edge events of EVNTPm0-15 are combined into one event EVENT\_PORTm output, and any one of the ports will output the event EVENT\_PORTm after detecting the edge.

## 10.4.21 Event Port Input Filter Control Register (PEVNTNFCR)

| Offset | Bit Position |    |    |    |       |    |       |    |          |    |    |    |       |    |       |    |          |    |    |    |       |    |       |   |          |   |   |   |       |   |       |   |
|--------|--------------|----|----|----|-------|----|-------|----|----------|----|----|----|-------|----|-------|----|----------|----|----|----|-------|----|-------|---|----------|---|---|---|-------|---|-------|---|
| 0x170  | 31           | 30 | 29 | 28 | 27    | 26 | 25    | 24 | 23       | 22 | 21 | 20 | 19    | 18 | 17    | 16 | 15       | 14 | 13 | 12 | 11    | 10 | 9     | 8 | 7        | 6 | 5 | 4 | 3     | 2 | 1     | 0 |
| Reset  | 0x00000000   |    |    |    |       |    |       |    |          |    |    |    |       |    |       |    |          |    |    |    |       |    |       |   |          |   |   |   |       |   |       |   |
| Name   | Reserved     |    |    |    | DIVS4 |    | NFEN4 |    | Reserved |    |    |    | DIVS3 |    | NFEN3 |    | Reserved |    |    |    | DIVS2 |    | NFEN2 |   | Reserved |   |   |   | DIVS1 |   | NFEN1 |   |
| Access |              |    |    |    | RW    |    | RW    |    |          |    |    |    | RW    |    | RW    |    |          |    |    |    | RW    |    | RW    |   |          |   |   |   | RW    |   | RW    |   |

| Bit   | Name     | Function                                            | Description                                                                                                                          | Access |
|-------|----------|-----------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:27 | Reserved | Reserved                                            | -                                                                                                                                    | -      |
| 26:25 | DIVS4    | Event Port4 digital filter sampling clock selection | <ul style="list-style-type: none"> <li>0b00: PCLK1</li> <li>0b01: PCLK1/8</li> <li>0b10: PCLK1/32</li> <li>0b11: PCLK1/64</li> </ul> | RW     |
| 24    | NFEN4    | Event Port4 digital filter permission               | <ul style="list-style-type: none"> <li>0b0: Digital filter disabled</li> <li>0b1: Digital filter enabled</li> </ul>                  | RW     |
| 23:19 | Reserved | Reserved                                            | -                                                                                                                                    | -      |
| 18:17 | DIVS3    | Event Port3 digital filter sampling clock selection | <ul style="list-style-type: none"> <li>0b00: PCLK1</li> <li>0b01: PCLK1/8</li> <li>0b10: PCLK1/32</li> <li>0b11: PCLK1/64</li> </ul> | RW     |
| 16    | NFEN3    | Event Port3 digital filter permission               | <ul style="list-style-type: none"> <li>0b0: Digital filter disabled</li> <li>0b1: Digital filter enabled</li> </ul>                  | RW     |
| 15:11 | Reserved | Reserved                                            | -                                                                                                                                    | -      |
| 10:9  | DIVS2    | Event Port2 digital filter sampling clock selection | <ul style="list-style-type: none"> <li>0b00: PCLK1</li> <li>0b01: PCLK1/8</li> <li>0b10: PCLK1/32</li> <li>0b11: PCLK1/64</li> </ul> | RW     |
| 8     | NFEN2    | Event Port2 digital filter permission               | <ul style="list-style-type: none"> <li>0b0: Digital filter disabled</li> <li>0b1: Digital filter enabled</li> </ul>                  | RW     |
| 7:3   | Reserved | Reserved                                            | -                                                                                                                                    | -      |
| 2:1   | DIVS1    | Event Port1 digital filter sampling clock selection | <ul style="list-style-type: none"> <li>0b00: PCLK1</li> <li>0b01: PCLK1/8</li> <li>0b10: PCLK1/32</li> <li>0b11: PCLK1/64</li> </ul> | RW     |
| 0     | NFEN1    | Event Port1 digital filter permission               | <ul style="list-style-type: none"> <li>0b0: Digital filter disabled</li> <li>0b1: Digital filter enabled</li> </ul>                  | RW     |

## 10.5 Precautions

1. Do not set the same feature to multiple ports.
2. When using the analog function, please turn off the digital function of the corresponding port (PCRxy.DDIS=0b1).
3. Please switch the port function when the output latch enabled (PCRxy.LTE=0b1) to avoid unexpected glitches on the port output during switching.

## 11 Interrupt Controller (INTC)

### 11.1 Introduction

The interrupt controller (INTC) selects an interrupt event as an interrupt request and sends it to NVIC to wake up WFI; selects interrupt event as event input (RXEV) to wake up WFE; select low power consumption mode (Sleep mode and Stop mode) of interrupt event wake-up system ; control external interrupts and software interrupts.

The main features of INTC are as follows:

1. Maskable interrupts: INTC is equipped with 457 interrupt events, which are processed and sent to NVIC as interrupt requests (IRQ). It supports 144 IRQs, each IRQ corresponds to one or more interrupt events.



#### Note

For more instructions on exception and NVIC programming, please refer to the "Arm Cortex-M4 Processor Technical Reference Manual".

2. Programmable priority of NVIC: 16 programmable priorities (4-bit interrupt priority register is used).
3. Non-maskable interrupts: Multiple system interrupt event requests can be independently selected as non-maskable interrupts, and each interrupt event is equipped with independent enable, flag and flag clear registers.
4. Equipped with 16 external pin interrupt events.
5. Equipped with multiple interrupt events, please refer to the "Interrupt Event List" for the specific number.
6. Equipped with 32 software interrupt events.
7. Interrupt wakes up system Sleep mode and Stop mode.

**Table 11-1 External Interrupt Input Pin**

| Pin Name | I/O   | Description                 |
|----------|-------|-----------------------------|
| EIRQ0    | Input | External Interrupt Event 0  |
| EIRQ1    | Input | External Interrupt Event 1  |
| EIRQ2    | Input | External Interrupt Event 2  |
| EIRQ3    | Input | External Interrupt Event 3  |
| EIRQ4    | Input | External Interrupt Event 4  |
| EIRQ5    | Input | External Interrupt Event 5  |
| EIRQ6    | Input | External Interrupt Event 6  |
| EIRQ7    | Input | External Interrupt Event 7  |
| EIRQ8    | Input | External Interrupt Event 8  |
| EIRQ9    | Input | External Interrupt Event 9  |
| EIRQ10   | Input | External Interrupt Event 10 |
| EIRQ11   | Input | External Interrupt Event 11 |
| EIRQ12   | Input | External Interrupt Event 12 |

| Pin Name | I/O   | Description                 |
|----------|-------|-----------------------------|
| EIRQ13   | Input | External Interrupt Event 13 |
| EIRQ14   | Input | External Interrupt Event 14 |
| EIRQ15   | Input | External Interrupt Event 15 |

## 11.2 Vector Digraph

### 11.2.1 Interrupt Vector Digraph

Table 11-2 Interrupt Vector Digraph

| Vector Address          | Vector No. | IRQ No. | Interrupt Source | Description                                                            |
|-------------------------|------------|---------|------------------|------------------------------------------------------------------------|
| ARM Core Interrupts     |            |         |                  |                                                                        |
| 0x00000000              | 0          | -       | ARM core         | Initial Stack Pointer                                                  |
| 0x00000004              | 1          | -       | ARM core         | Initial Program Counter                                                |
| 0x00000008              | 2          | -       | ARM core         | Non-maskable Interrupt(NMI)                                            |
| 0x0000000C              | 3          | -       | ARM core         | Hard Fault                                                             |
| 0x00000010              | 4          | -       | ARM core         | Mem Manage Fault                                                       |
| 0x00000014              | 5          | -       | ARM core         | Bus Fault                                                              |
| 0x00000018              | 6          | -       | ARM core         | Usage Fault                                                            |
| 0x0000001C              | 7          | -       | ARM core         | Reserved                                                               |
| 0x00000020              | 8          | -       | ARM core         | Reserved                                                               |
| 0x00000024              | 9          | -       | ARM core         | Reserved                                                               |
| 0x00000028              | 10         | -       | ARM core         | Reserved                                                               |
| 0x0000002C              | 11         | -       | ARM core         | Supervisor Call(SVCall)                                                |
| 0x00000030              | 12         | -       | ARM core         | Debug Monitor                                                          |
| 0x00000034              | 13         | -       | ARM core         | Reserved                                                               |
| 0x00000038              | 14         | -       | ARM core         | Pendable request for system service (PendSV)                           |
| 0x0000003C              | 15         | -       | ARM core         | System tick timer (SysTick)                                            |
| Non-ARM core interrupts |            |         |                  |                                                                        |
| 0x00000040              | 16         | 0       | INTC_INTSEL0     | Interrupt event/software interrupt 0 selected by register INTC_INTSEL0 |
| 0x00000044              | 17         | 1       | INTC_INTSEL1     | Interrupt event/software interrupt 1 selected by register INTC_INTSEL1 |
| 0x00000048              | 18         | 2       | INTC_INTSEL2     | Interrupt event/software interrupt 2 selected by register INTC_INTSEL2 |
| 0x0000004C              | 19         | 3       | INTC_INTSEL3     | Interrupt event/software interrupt 3 selected by register INTC_INTSEL3 |
| 0x00000050              | 20         | 4       | INTC_INTSEL4     | Interrupt event/software interrupt 4 selected by register INTC_INTSEL4 |
| 0x00000054              | 21         | 5       | INTC_INTSEL5     | Interrupt event/software interrupt 5 selected by register INTC_INTSEL5 |
| 0x00000058              | 22         | 6       | INTC_INTSEL6     | Interrupt event/software interrupt 6 selected by register INTC_INTSEL6 |
| 0x0000005C              | 23         | 7       | INTC_INTSEL7     | Interrupt event/software interrupt 7 selected by register INTC_INTSEL7 |
| 0x00000060              | 24         | 8       | INTC_INTSEL8     | Interrupt event/software interrupt 8 selected by register INTC_INTSEL8 |
| 0x00000064              | 25         | 9       | INTC_INTSEL9     | Interrupt event/software interrupt 9 selected by register INTC_INTSEL9 |

| Vector Address | Vector No. | IRQ No. | Interrupt Source | Description                                                                   |
|----------------|------------|---------|------------------|-------------------------------------------------------------------------------|
| 0x00000068     | 26         | 10      | INTC_INTSEL10    | Interrupt event/software interrupt 10 selected by register INTC_INTSEL10      |
| 0x0000006C     | 27         | 11      | INTC_INTSEL11    | Interrupt event /software interrupt 11selected by register INTC_INTSEL11      |
| 0x00000070     | 28         | 12      | INTC_INTSEL12    | Interrupt event/software interrupt 12 selected by register INTC_INTSEL12      |
| 0x00000074     | 29         | 13      | INTC_INTSEL13    | Interrupt event/software interrupt 13 selected by register INTC_INTSEL13      |
| 0x00000078     | 30         | 14      | INTC_INTSEL14    | Interrupt event/software interrupt 14 selected by register INTC_INTSEL14      |
| 0x0000007C     | 31         | 15      | INTC_INTSEL15    | Interrupt event/software interrupt 15 selected by register INTC_INTSEL15      |
| 0x00000080     | 32         | 16      | INTC_INTSEL16    | Interrupt event request/software interrupt selected by register INTC_INTSEL16 |
| 0x00000084     | 33         | 17      | INTC_INTSEL17    | Interrupt event request/software interrupt selected by register INTC_INTSEL17 |
| 0x00000088     | 34         | 18      | INTC_INTSEL18    | Interrupt event request/software interrupt selected by register INTC_INTSEL18 |
| 0x0000008C     | 35         | 19      | INTC_INTSEL19    | Interrupt event request/software interrupt selected by register INTC_INTSEL19 |
| 0x00000090     | 36         | 20      | INTC_INTSEL20    | Interrupt event request/software interrupt selected by register INTC_INTSEL20 |
| 0x00000094     | 37         | 21      | INTC_INTSEL21    | Interrupt event request/software interrupt selected by register INTC_INTSEL21 |
| 0x00000098     | 38         | 22      | INTC_INTSEL22    | Interrupt event request/software interrupt selected by register INTC_INTSEL22 |
| 0x0000009C     | 39         | 23      | INTC_INTSEL23    | Interrupt event request/software interrupt selected by register INTC_INTSEL23 |
| 0x000000A0     | 40         | 24      | INTC_INTSEL24    | Interrupt event request/software interrupt selected by register INTC_INTSEL24 |
| 0x000000A4     | 41         | 25      | INTC_INTSEL25    | Interrupt event request/software interrupt selected by register INTC_INTSEL25 |
| 0x000000A8     | 42         | 26      | INTC_INTSEL26    | Interrupt event request/software interrupt selected by register INTC_INTSEL26 |
| 0x000000AC     | 43         | 27      | INTC_INTSEL27    | Interrupt event request/software interrupt selected by register INTC_INTSEL27 |
| 0x000000B0     | 44         | 28      | INTC_INTSEL28    | Interrupt event request/software interrupt selected by register INTC_INTSEL28 |
| 0x000000B4     | 45         | 29      | INTC_INTSEL29    | Interrupt event request/software interrupt selected by register INTC_INTSEL29 |
| 0x000000B8     | 46         | 30      | INTC_INTSEL30    | Interrupt event request/software interrupt selected by register INTC_INTSEL30 |
| 0x000000BC     | 47         | 31      | INTC_INTSEL31    | Interrupt event request/software interrupt selected by register INTC_INTSEL31 |
| 0x000000C0     | 48         | 32      | INTC_INTSEL32    | Interrupt event request selected by register INTC_INTSEL32                    |
| 0x000000C4     | 49         | 33      | INTC_INTSEL33    | Interrupt event request selected by register INTC_INTSEL33                    |
| 0x000000C8     | 50         | 34      | INTC_INTSEL34    | Interrupt event request selected by register INTC_INTSEL34                    |
| 0x000000CC     | 51         | 35      | INTC_INTSEL35    | Interrupt event request selected by register INTC_INTSEL35                    |
| 0x000000D0     | 52         | 36      | INTC_INTSEL36    | Interrupt event request selected by register INTC_INTSEL36                    |
| 0x000000D4     | 53         | 37      | INTC_INTSEL37    | Interrupt event request selected by register INTC_INTSEL37                    |
| 0x000000D8     | 54         | 38      | INTC_INTSEL38    | Interrupt event request selected by register INTC_INTSEL38                    |
| 0x000000DC     | 55         | 39      | INTC_INTSEL39    | Interrupt event request selected by register INTC_INTSEL39                    |
| 0x000000E0     | 56         | 40      | INTC_INTSEL40    | Interrupt event request selected by register INTC_INTSEL40                    |



| Vector Address | Vector No. | IRQ No. | Interrupt Source | Description                                                |
|----------------|------------|---------|------------------|------------------------------------------------------------|
| 0x000000E4     | 57         | 41      | INTC_INTSEL41    | Interrupt event request selected by register INTC_INTSEL41 |
| 0x000000E8     | 58         | 42      | INTC_INTSEL42    | Interrupt event request selected by register INTC_INTSEL42 |
| 0x000000EC     | 59         | 43      | INTC_INTSEL43    | Interrupt event request selected by register INTC_INTSEL43 |
| 0x000000F0     | 60         | 44      | INTC_INTSEL44    | Interrupt event request selected by register INTC_INTSEL44 |
| 0x000000F4     | 61         | 45      | INTC_INTSEL45    | Interrupt event request selected by register INTC_INTSEL45 |
| 0x000000F8     | 62         | 46      | INTC_INTSEL46    | Interrupt event request selected by register INTC_INTSEL46 |
| 0x000000FC     | 63         | 47      | INTC_INTSEL47    | Interrupt event request selected by register INTC_INTSEL47 |
| 0x00000100     | 64         | 48      | INTC_INTSEL48    | Interrupt event request selected by register INTC_INTSEL48 |
| 0x00000104     | 65         | 49      | INTC_INTSEL49    | Interrupt event request selected by register INTC_INTSEL49 |
| 0x00000108     | 66         | 50      | INTC_INTSEL50    | Interrupt event request selected by register INTC_INTSEL50 |
| 0x0000010C     | 67         | 51      | INTC_INTSEL51    | Interrupt event request selected by register INTC_INTSEL51 |
| 0x00000110     | 68         | 52      | INTC_INTSEL52    | Interrupt event request selected by register INTC_INTSEL52 |
| 0x00000114     | 69         | 53      | INTC_INTSEL53    | Interrupt event request selected by register INTC_INTSEL53 |
| 0x00000118     | 70         | 54      | INTC_INTSEL54    | Interrupt event request selected by register INTC_INTSEL54 |
| 0x0000011C     | 71         | 55      | INTC_INTSEL55    | Interrupt event request selected by register INTC_INTSEL55 |
| 0x00000120     | 72         | 56      | INTC_INTSEL56    | Interrupt event request selected by register INTC_INTSEL56 |
| 0x00000124     | 73         | 57      | INTC_INTSEL57    | Interrupt event request selected by register INTC_INTSEL57 |
| 0x00000128     | 74         | 58      | INTC_INTSEL58    | Interrupt event request selected by register INTC_INTSEL58 |
| 0x0000012C     | 75         | 59      | INTC_INTSEL59    | Interrupt event request selected by register INTC_INTSEL59 |
| 0x00000130     | 76         | 60      | INTC_INTSEL60    | Interrupt event request selected by register INTC_INTSEL60 |
| 0x00000134     | 77         | 61      | INTC_INTSEL61    | Interrupt event request selected by register INTC_INTSEL61 |
| 0x00000138     | 78         | 62      | INTC_INTSEL62    | Interrupt event request selected by register INTC_INTSEL62 |
| 0x0000013C     | 79         | 63      | INTC_INTSEL63    | Interrupt event request selected by register INTC_INTSEL63 |
| 0x00000140     | 80         | 64      | INTC_INTSEL64    | Interrupt event request selected by register INTC_INTSEL64 |
| 0x00000144     | 81         | 65      | INTC_INTSEL65    | Interrupt event request selected by register INTC_INTSEL65 |
| 0x00000148     | 82         | 66      | INTC_INTSEL66    | Interrupt event request selected by register INTC_INTSEL66 |
| 0x0000014C     | 83         | 67      | INTC_INTSEL67    | Interrupt event request selected by register INTC_INTSEL67 |
| 0x00000150     | 84         | 68      | INTC_INTSEL68    | Interrupt event request selected by register INTC_INTSEL68 |
| 0x00000154     | 85         | 69      | INTC_INTSEL69    | Interrupt event request selected by register INTC_INTSEL69 |
| 0x00000158     | 86         | 70      | INTC_INTSEL70    | Interrupt event request selected by register INTC_INTSEL70 |
| 0x0000015C     | 87         | 71      | INTC_INTSEL71    | Interrupt event request selected by register INTC_INTSEL71 |
| 0x00000160     | 88         | 72      | INTC_INTSEL72    | Interrupt event request selected by register INTC_INTSEL72 |
| 0x00000164     | 89         | 73      | INTC_INTSEL73    | Interrupt event request selected by register INTC_INTSEL73 |
| 0x00000168     | 90         | 74      | INTC_INTSEL74    | Interrupt event request selected by register INTC_INTSEL74 |
| 0x0000016C     | 91         | 75      | INTC_INTSEL75    | Interrupt event request selected by register INTC_INTSEL75 |
| 0x00000170     | 92         | 76      | INTC_INTSEL76    | Interrupt event request selected by register INTC_INTSEL76 |
| 0x00000174     | 93         | 77      | INTC_INTSEL77    | Interrupt event request selected by register INTC_INTSEL77 |
| 0x00000178     | 94         | 78      | INTC_INTSEL78    | Interrupt event request selected by register INTC_INTSEL78 |
| 0x0000017C     | 95         | 79      | INTC_INTSEL79    | Interrupt event request selected by register INTC_INTSEL79 |
| 0x00000180     | 96         | 80      | INTC_INTSEL80    | Interrupt event request selected by register INTC_INTSEL80 |
| 0x00000184     | 97         | 81      | INTC_INTSEL81    | Interrupt event request selected by register INTC_INTSEL81 |
| 0x00000188     | 98         | 82      | INTC_INTSEL82    | Interrupt event request selected by register INTC_INTSEL82 |
| 0x0000018C     | 99         | 83      | INTC_INTSEL83    | Interrupt event request selected by register INTC_INTSEL83 |
| 0x00000190     | 100        | 84      | INTC_INTSEL84    | Interrupt event request selected by register INTC_INTSEL84 |
| 0x00000194     | 101        | 85      | INTC_INTSEL85    | Interrupt event request selected by register INTC_INTSEL85 |

| Vector Address | Vector No. | IRQ No. | Interrupt Source | Description                                                 |
|----------------|------------|---------|------------------|-------------------------------------------------------------|
| 0x00000198     | 102        | 86      | INTC_INTSEL86    | Interrupt event request selected by register INTC_INTSEL86  |
| 0x0000019C     | 103        | 87      | INTC_INTSEL87    | Interrupt event request selected by register INTC_INTSEL87  |
| 0x000001A0     | 104        | 88      | INTC_INTSEL88    | Interrupt event request selected by register INTC_INTSEL88  |
| 0x000001A4     | 105        | 89      | INTC_INTSEL89    | Interrupt event request selected by register INTC_INTSEL89  |
| 0x000001A8     | 106        | 90      | INTC_INTSEL90    | Interrupt event request selected by register INTC_INTSEL90  |
| 0x000001AC     | 107        | 91      | INTC_INTSEL91    | Interrupt event request selected by register INTC_INTSEL91  |
| 0x000001B0     | 108        | 92      | INTC_INTSEL92    | Interrupt event request selected by register INTC_INTSEL92  |
| 0x000001B4     | 109        | 93      | INTC_INTSEL93    | Interrupt event request selected by register INTC_INTSEL93  |
| 0x000001B8     | 110        | 94      | INTC_INTSEL94    | Interrupt event request selected by register INTC_INTSEL94  |
| 0x000001BC     | 111        | 95      | INTC_INTSEL95    | Interrupt event request selected by register INTC_INTSEL95  |
| 0x000001C0     | 112        | 96      | INTC_INTSEL96    | Interrupt event request selected by register INTC_INTSEL96  |
| 0x000001C4     | 113        | 97      | INTC_INTSEL97    | Interrupt event request selected by register INTC_INTSEL97  |
| 0x000001C8     | 114        | 98      | INTC_INTSEL98    | Interrupt event request selected by register INTC_INTSEL98  |
| 0x000001CC     | 115        | 99      | INTC_INTSEL99    | Interrupt event request selected by register INTC_INTSEL99  |
| 0x000001D0     | 116        | 100     | INTC_INTSEL100   | Interrupt event request selected by register INTC_INTSEL100 |
| 0x000001D4     | 117        | 101     | INTC_INTSEL101   | Interrupt event request selected by register INTC_INTSEL101 |
| 0x000001D8     | 118        | 102     | INTC_INTSEL102   | Interrupt event request selected by register INTC_INTSEL102 |
| 0x000001DC     | 119        | 103     | INTC_INTSEL103   | Interrupt event request selected by register INTC_INTSEL103 |
| 0x000001E0     | 120        | 104     | INTC_INTSEL104   | Interrupt event request selected by register INTC_INTSEL104 |
| 0x000001E4     | 121        | 105     | INTC_INTSEL105   | Interrupt event request selected by register INTC_INTSEL105 |
| 0x000001E8     | 122        | 106     | INTC_INTSEL106   | Interrupt event request selected by register INTC_INTSEL106 |
| 0x000001EC     | 123        | 107     | INTC_INTSEL107   | Interrupt event request selected by register INTC_INTSEL107 |
| 0x000001F0     | 124        | 108     | INTC_INTSEL108   | Interrupt event request selected by register INTC_INTSEL108 |
| 0x000001F4     | 125        | 109     | INTC_INTSEL109   | Interrupt event request selected by register INTC_INTSEL109 |
| 0x000001F8     | 126        | 110     | INTC_INTSEL110   | Interrupt event request selected by register INTC_INTSEL110 |
| 0x000001FC     | 127        | 111     | INTC_INTSEL111   | Interrupt event request selected by register INTC_INTSEL111 |
| 0x00000200     | 128        | 112     | INTC_INTSEL112   | Interrupt event request selected by register INTC_INTSEL112 |
| 0x00000204     | 129        | 113     | INTC_INTSEL113   | Interrupt event request selected by register INTC_INTSEL113 |
| 0x00000208     | 130        | 114     | INTC_INTSEL114   | Interrupt event request selected by register INTC_INTSEL114 |
| 0x0000020C     | 131        | 115     | INTC_INTSEL115   | Interrupt event request selected by register INTC_INTSEL115 |
| 0x00000210     | 132        | 116     | INTC_INTSEL116   | Interrupt event request selected by register INTC_INTSEL116 |
| 0x00000214     | 133        | 117     | INTC_INTSEL117   | Interrupt event request selected by register INTC_INTSEL117 |
| 0x00000218     | 134        | 118     | INTC_INTSEL118   | Interrupt event request selected by register INTC_INTSEL118 |

| Vector Address | Vector No. | IRQ No. | Interrupt Source | Description                                                                                      |
|----------------|------------|---------|------------------|--------------------------------------------------------------------------------------------------|
| 0x0000021C     | 135        | 119     | INTC_INTSEL119   | Interrupt event request selected by register INTC_INTSEL119                                      |
| 0x00000220     | 136        | 120     | INTC_INTSEL120   | Interrupt event request selected by register INTC_INTSEL120                                      |
| 0x00000224     | 137        | 121     | INTC_INTSEL121   | Interrupt event request selected by register INTC_INTSEL121                                      |
| 0x00000228     | 138        | 122     | INTC_INTSEL122   | Interrupt event request selected by register INTC_INTSEL122                                      |
| 0x00000234     | 141        | 125     | INTC_INTSEL123   | Interrupt event request selected by register INTC_INTSEL123                                      |
| 0x00000238     | 142        | 126     | INTC_INTSEL124   | Interrupt event request selected by register INTC_INTSEL124                                      |
| 0x0000023C     | 143        | 127     | INTC_INTSEL125   | Interrupt event request selected by register INTC_INTSEL125                                      |
| 0x00000240     | 144        | 128     | INTC_VSSEL128    | Register INTC_VSSEL128 shares this vector with interrupt event requests selected by enable bits. |
| 0x00000244     | 145        | 129     | INTC_VSSEL129    | Register INTC_VSSEL129 shares this vector with interrupt event requests selected by enable bits. |
| 0x00000248     | 146        | 130     | INTC_VSSEL130    | Register INTC_VSSEL130 shares this vector with interrupt event requests selected by enable bits. |
| 0x0000024C     | 147        | 131     | INTC_VSSEL131    | Register INTC_VSSEL131 shares this vector with interrupt event requests selected by enable bits. |
| 0x00000250     | 148        | 132     | INTC_VSSEL132    | Register INTC_VSSEL132 shares this vector with interrupt event requests selected by enable bits. |
| 0x00000254     | 149        | 133     | INTC_VSSEL133    | Register INTC_VSSEL133 shares this vector with interrupt event requests selected by enable bits. |
| 0x00000258     | 150        | 134     | INTC_VSSEL134    | Register INTC_VSSEL134 shares this vector with interrupt event requests selected by enable bits. |
| 0x0000025C     | 151        | 135     | INTC_VSSEL135    | Register INTC_VSSEL135 shares this vector with interrupt event requests selected by enable bits. |
| 0x00000260     | 152        | 136     | INTC_VSSEL136    | Register INTC_VSSEL136 shares this vector with interrupt event requests selected by enable bits. |
| 0x00000264     | 153        | 137     | INTC_VSSEL137    | Register INTC_VSSEL137 shares this vector with interrupt event requests selected by enable bits. |
| 0x00000268     | 154        | 138     | INTC_VSSEL138    | Register INTC_VSSEL138 shares this vector with interrupt event requests selected by enable bits. |
| 0x0000026C     | 155        | 139     | INTC_VSSEL139    | Register INTC_VSSEL139 shares this vector with interrupt event requests selected by enable bits. |
| 0x00000270     | 156        | 140     | INTC_VSSEL140    | Register INTC_VSSEL140 shares this vector with interrupt event requests selected by enable bits. |
| 0x00000274     | 157        | 141     | INTC_VSSEL141    | Register INTC_VSSEL141 shares this vector with interrupt event requests selected by enable bits. |
| 0x00000278     | 158        | 142     | INTC_VSSEL142    | Register INTC_VSSEL142 shares this vector with interrupt event requests selected by enable bits. |
| 0x0000027C     | 159        | 143     | INTC_VSSEL143    | Register INTC_VSSEL143 shares this vector with interrupt event requests selected by enable bits. |

**Note**

For the interrupt event number selected by the INTC\_INTSEL register, see the [Interrupt Event List](#).

## 11.2.2 Interrupt Event List

Interrupt events are generated by peripherals and are called interrupt sources when they are input as IRQs; When used as RXEV input, it is called an event source; when used as a peripheral trigger event, it is called a trigger source; Interrupt events can also be used as a condition to return from low power mode.

| SN | Inter-<br>rupt<br>Event<br>No. | Modules  | Interrupt Name | Inter-<br>rupt<br>Type <sup>(1)</sup> | Inter-<br>rupt<br>Source | Trig-<br>ger<br>Source | NVIC Interrupt Request (IRQ) Number Mapping <sup>(2)</sup> |                       |                        |
|----|--------------------------------|----------|----------------|---------------------------------------|--------------------------|------------------------|------------------------------------------------------------|-----------------------|------------------------|
|    |                                |          |                |                                       |                          |                        | NVIC Vector<br>0-31                                        | NVIC Vector<br>32-127 | NVIC Vector<br>128-143 |
| 0  | 0x000                          | PORT     | EIRQ0          | P/L                                   | ✓                        | ✓                      | INTSEL0-31                                                 | INTSEL32-37           | VSSEL128[0]            |
| 1  | 0x001                          |          | EIRQ1          | P/L                                   | ✓                        | ✓                      | INTSEL0-31                                                 | INTSEL32-37           | VSSEL128[1]            |
| 2  | 0x002                          |          | EIRQ2          | P/L                                   | ✓                        | ✓                      | INTSEL0-31                                                 | INTSEL32-37           | VSSEL128[2]            |
| 3  | 0x003                          |          | EIRQ3          | P/L                                   | ✓                        | ✓                      | INTSEL0-31                                                 | INTSEL32-37           | VSSEL128[3]            |
| 4  | 0x004                          |          | EIRQ4          | P/L                                   | ✓                        | ✓                      | INTSEL0-31                                                 | INTSEL32-37           | VSSEL128[4]            |
| 5  | 0x005                          |          | EIRQ5          | P/L                                   | ✓                        | ✓                      | INTSEL0-31                                                 | INTSEL32-37           | VSSEL128[5]            |
| 6  | 0x006                          |          | EIRQ6          | P/L                                   | ✓                        | ✓                      | INTSEL0-31                                                 | INTSEL32-37           | VSSEL128[6]            |
| 7  | 0x007                          |          | EIRQ7          | P/L                                   | ✓                        | ✓                      | INTSEL0-31                                                 | INTSEL32-37           | VSSEL128[7]            |
| 8  | 0x008                          |          | EIRQ8          | P/L                                   | ✓                        | ✓                      | INTSEL0-31                                                 | INTSEL32-37           | VSSEL128[8]            |
| 9  | 0x009                          |          | EIRQ9          | P/L                                   | ✓                        | ✓                      | INTSEL0-31                                                 | INTSEL32-37           | VSSEL128[9]            |
| 10 | 0x00A                          |          | EIRQ10         | P/L                                   | ✓                        | ✓                      | INTSEL0-31                                                 | INTSEL32-37           | VSSEL128[10]           |
| 11 | 0x00B                          |          | EIRQ11         | P/L                                   | ✓                        | ✓                      | INTSEL0-31                                                 | INTSEL32-37           | VSSEL128[11]           |
| 12 | 0x00C                          |          | EIRQ12         | P/L                                   | ✓                        | ✓                      | INTSEL0-31                                                 | INTSEL32-37           | VSSEL128[12]           |
| 13 | 0x00D                          |          | EIRQ13         | P/L                                   | ✓                        | ✓                      | INTSEL0-31                                                 | INTSEL32-37           | VSSEL128[13]           |
| 14 | 0x00E                          |          | EIRQ14         | P/L                                   | ✓                        | ✓                      | INTSEL0-31                                                 | INTSEL32-37           | VSSEL128[14]           |
| 15 | 0x00F                          |          | EIRQ15         | P/L                                   | ✓                        | ✓                      | INTSEL0-31                                                 | INTSEL32-37           | VSSEL128[15]           |
| 16 | 0x010                          | Timer0_3 | TMR0_3_CMPA    | P                                     | ✓                        | ✓                      | INTSEL0-31                                                 | INTSEL32-37           | VSSEL128[16]           |
| 17 | 0x011                          |          | TMR0_3_CMPB    | P                                     | ✓                        | ✓                      | INTSEL0-31                                                 | INTSEL32-37           | VSSEL128[17]           |
| 18 | 0x012                          | Timer0_4 | TMR0_4_CMPA    | P                                     | ✓                        | ✓                      | INTSEL0-31                                                 | INTSEL32-37           | VSSEL128[18]           |
| 19 | 0x013                          |          | TMR0_4_CMPB    | P                                     | ✓                        | ✓                      | INTSEL0-31                                                 | INTSEL32-37           | VSSEL128[19]           |
| 20 | 0x014                          | Timer0_5 | TMR0_5_CMPA    | P                                     | ✓                        | ✓                      | INTSEL0-31                                                 | INTSEL32-37           | VSSEL128[20]           |
| 21 | 0x015                          |          | TMR0_5_CMPB    | P                                     | ✓                        | ✓                      | INTSEL0-31                                                 | INTSEL32-37           | VSSEL128[21]           |
| 22 | 0x016                          | Timer0_1 | TMR0_1_OVFA    | P                                     | ✓                        | ✓                      | INTSEL0-31                                                 | INTSEL32-37           | VSSEL128[22]           |
| 23 | 0x017                          |          | TMR0_1_OVFB    | P                                     | ✓                        | ✓                      | INTSEL0-31                                                 | INTSEL32-37           | VSSEL128[23]           |
| 24 | 0x018                          | Timer0_2 | TMR0_2_OVFA    | P                                     | ✓                        | ✓                      | INTSEL0-31                                                 | INTSEL32-37           | VSSEL128[24]           |
| 25 | 0x019                          |          | TMR0_2_OVFB    | P                                     | ✓                        | ✓                      | INTSEL0-31                                                 | INTSEL32-37           | VSSEL128[25]           |
| 26 | 0x01A                          | Timer0_3 | TMR0_3_OVFA    | P                                     | ✓                        | ✓                      | INTSEL0-31                                                 | INTSEL32-37           | VSSEL128[26]           |
| 27 | 0x01B                          |          | TMR0_3_OVFB    | P                                     | ✓                        | ✓                      | INTSEL0-31                                                 | INTSEL32-37           | VSSEL128[27]           |
| 28 | 0x01C                          | Timer0_4 | TMR0_4_OVFA    | P                                     | ✓                        | ✓                      | INTSEL0-31                                                 | INTSEL32-37           | VSSEL128[28]           |
| 29 | 0x01D                          |          | TMR0_4_OVFB    | P                                     | ✓                        | ✓                      | INTSEL0-31                                                 | INTSEL32-37           | VSSEL128[29]           |
| 30 | 0x01E                          | Timer0_5 | TMR0_5_OVFA    | P                                     | ✓                        | ✓                      | INTSEL0-31                                                 | INTSEL32-37           | VSSEL128[30]           |
| 31 | 0x01F                          |          | TMR0_5_OVFB    | P                                     | ✓                        | ✓                      | INTSEL0-31                                                 | INTSEL32-37           | VSSEL128[31]           |
| 32 | 0x020                          | DMA_1    | DMA_1_TC0      | P                                     | ✓                        | ✓                      | INTSEL0-31                                                 | INTSEL38-43           | VSSEL129[0]            |
| 33 | 0x021                          |          | DMA_1_TC1      | P                                     | ✓                        | ✓                      | INTSEL0-31                                                 | INTSEL38-43           | VSSEL129[1]            |
| 34 | 0x022                          |          | DMA_1_TC2      | P                                     | ✓                        | ✓                      | INTSEL0-31                                                 | INTSEL38-43           | VSSEL129[2]            |
| 35 | 0x023                          |          | DMA_1_TC3      | P                                     | ✓                        | ✓                      | INTSEL0-31                                                 | INTSEL38-43           | VSSEL129[3]            |

| SN | Interrupt Event No. | Modules | Interrupt Name | Interrupt Type <sup>(1)</sup> | Interrupt Source | Trigger Source | NVIC Interrupt Request (IRQ) Number Mapping <sup>(2)</sup> |                    |                     |
|----|---------------------|---------|----------------|-------------------------------|------------------|----------------|------------------------------------------------------------|--------------------|---------------------|
|    |                     |         |                |                               |                  |                | NVIC Vector 0-31                                           | NVIC Vector 32-127 | NVIC Vector 128-143 |
| 36 | 0x024               | DMA_1   | DMA_1_TC4      | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL38-43        | VSSEL129[4]         |
| 37 | 0x025               |         | DMA_1_TC5      | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL38-43        | VSSEL129[5]         |
| 38 | 0x026               |         | DMA_1_TC6      | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL38-43        | VSSEL129[6]         |
| 39 | 0x027               |         | DMA_1_TC7      | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL38-43        | VSSEL129[7]         |
| 40 | 0x028               |         | DMA_1_BTC0     | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL38-43        | VSSEL129[8]         |
| 41 | 0x029               |         | DMA_1_BTC1     | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL38-43        | VSSEL129[9]         |
| 42 | 0x02A               |         | DMA_1_BTC2     | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL38-43        | VSSEL129[10]        |
| 43 | 0x02B               |         | DMA_1_BTC3     | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL38-43        | VSSEL129[11]        |
| 44 | 0x02C               |         | DMA_1_BTC4     | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL38-43        | VSSEL129[12]        |
| 45 | 0x02D               |         | DMA_1_BTC5     | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL38-43        | VSSEL129[13]        |
| 46 | 0x02E               |         | DMA_1_BTC6     | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL38-43        | VSSEL129[14]        |
| 47 | 0x02F               |         | DMA_1_BTC7     | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL38-43        | VSSEL129[15]        |
| 48 | 0x030               |         | DMA_1_ERR      | P                             | ✓                | -              | INTSEL0-31                                                 | INTSEL38-43        | VSSEL129[16]        |
| 49 | 0x031               | EFM     | EFM_PEERR      | P                             | ✓                | -              | INTSEL0-31                                                 | INTSEL38-43        | VSSEL129[17]        |
| 50 | 0x032               |         | EFM_RDCOL      | P                             | ✓                | -              | INTSEL0-31                                                 | INTSEL38-43        | VSSEL129[18]        |
| 51 | 0x033               |         | EFM_OPTEND     | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL38-43        | VSSEL129[19]        |
| 52 | 0x034               | USBFS   | USBFS_SOF      | -                             | -                | ✓              | INTSEL0-31                                                 | INTSEL38-43        | VSSEL129[20]        |
| 53 | 0x035               | USBHS   | USBHS_SOF      | -                             | -                | ✓              | INTSEL0-31                                                 | INTSEL38-43        | VSSEL129[21]        |
| 54 | 0x036               | QSPI    | QSPI_INTR      | L                             | ✓                | -              | INTSEL0-31                                                 | INTSEL38-43        | VSSEL129[22]        |
| 55 | 0x037               | DCU     | DCU1           | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL38-43        | VSSEL129[23]        |
| 56 | 0x038               |         | DCU2           | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL38-43        | VSSEL129[24]        |
| 57 | 0x039               |         | DCU3           | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL38-43        | VSSEL129[25]        |
| 58 | 0x03A               |         | DCU4           | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL38-43        | VSSEL129[26]        |
| 59 | 0x03B               |         | DCU5           | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL38-43        | VSSEL129[27]        |
| 60 | 0x03C               |         | DCU6           | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL38-43        | VSSEL129[28]        |
| 61 | 0x03D               |         | DCU7           | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL38-43        | VSSEL129[29]        |
| 62 | 0x03E               |         | DCU8           | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL38-43        | VSSEL129[30]        |
| 63 | 0x03F               | FPU     | FPU_ERR        | L                             | ✓                | -              | INTSEL0-31                                                 | INTSEL38-43        | VSSEL129[31]        |

| SN  | Interrupt Event No. | Modules  | Interrupt Name | Interrupt Type <sup>(1)</sup> | Interrupt Source | Trigger Source | NVIC Interrupt Request (IRQ) Number Mapping <sup>(2)</sup> |                    |                     |
|-----|---------------------|----------|----------------|-------------------------------|------------------|----------------|------------------------------------------------------------|--------------------|---------------------|
|     |                     |          |                |                               |                  |                | NVIC Vector 0-31                                           | NVIC Vector 32-127 | NVIC Vector 128-143 |
| 64  | 0x040               | DMA_2    | DMA_2_TC0      | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL44-49        | VSSEL130[0]         |
| 65  | 0x041               |          | DMA_2_TC1      | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL44-49        | VSSEL130[1]         |
| 66  | 0x042               |          | DMA_2_TC2      | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL44-49        | VSSEL130[2]         |
| 67  | 0x043               |          | DMA_2_TC3      | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL44-49        | VSSEL130[3]         |
| 68  | 0x044               |          | DMA_2_TC4      | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL44-49        | VSSEL130[4]         |
| 69  | 0x045               |          | DMA_2_TC5      | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL44-49        | VSSEL130[5]         |
| 70  | 0x046               |          | DMA_2_TC6      | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL44-49        | VSSEL130[6]         |
| 71  | 0x047               |          | DMA_2_TC7      | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL44-49        | VSSEL130[7]         |
| 72  | 0x048               |          | DMA_2 BTC0     | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL44-49        | VSSEL130[8]         |
| 73  | 0x049               |          | DMA_2 BTC1     | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL44-49        | VSSEL130[9]         |
| 74  | 0x04A               |          | DMA_2 BTC2     | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL44-49        | VSSEL130[10]        |
| 75  | 0x04B               |          | DMA_2 BTC3     | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL44-49        | VSSEL130[11]        |
| 76  | 0x04C               |          | DMA_2 BTC4     | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL44-49        | VSSEL130[12]        |
| 77  | 0x04D               |          | DMA_2 BTC5     | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL44-49        | VSSEL130[13]        |
| 78  | 0x04E               |          | DMA_2 BTC6     | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL44-49        | VSSEL130[14]        |
| 79  | 0x04F               |          | DMA_2 BTC7     | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL44-49        | VSSEL130[15]        |
| 80  | 0x050               |          | DMA_2_ERR      | P                             | ✓                | -              | INTSEL0-31                                                 | INTSEL44-49        | VSSEL130[16]        |
| 81  | 0x051               | -        | -              | -                             | -                | -              | INTSEL0-31                                                 | INTSEL44-49        | VSSEL130[17]        |
| 82  | 0x052               | -        | -              | -                             | -                | -              | INTSEL0-31                                                 | INTSEL44-49        | VSSEL130[18]        |
| 83  | 0x053               | MAU      | MAU_SQRT       | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL44-49        | VSSEL130[19]        |
| 84  | 0x054               | DVP      | DVP_FRAMSTA    | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL44-49        | VSSEL130[20]        |
| 85  | 0x055               |          | DVP_LINESTA    | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL44-49        | VSSEL130[21]        |
| 86  | 0x056               |          | DVP_LINEEND    | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL44-49        | VSSEL130[22]        |
| 87  | 0x057               |          | DVP_FRAMEND    | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL44-49        | VSSEL130[23]        |
| 88  | 0x058               |          | DVP_SQUERR     | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL44-49        | VSSEL130[24]        |
| 89  | 0x059               |          | DVP_FIFOERR    | L                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL44-49        | VSSEL130[25]        |
| 90  | 0x05A               |          | DVP_DMAREQ     | -                             | -                | ✓              | INTSEL0-31                                                 | INTSEL44-49        | VSSEL130[26]        |
| 91  | 0x05B               | FMAC     | FMAC_1_FIR     | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL44-49        | VSSEL130[27]        |
| 92  | 0x05C               |          | FMAC_2_FIR     | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL44-49        | VSSEL130[28]        |
| 93  | 0x05D               |          | FMAC_3_FIR     | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL44-49        | VSSEL130[29]        |
| 94  | 0x05E               |          | FMAC_4_FIR     | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL44-49        | VSSEL130[30]        |
| 95  | 0x05F               | -        | -              | -                             | -                | -              | INTSEL0-31                                                 | INTSEL44-49        | VSSEL130[31]        |
| 96  | 0x060               | Timer0_1 | TMR0_1_CMPA    | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL50-55        | VSSEL131[0]         |
| 97  | 0x061               |          | TMR0_1_CMPB    | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL50-55        | VSSEL131[1]         |
| 98  | 0x062               | Timer0_2 | TMR0_2_CMPA    | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL50-55        | VSSEL131[2]         |
| 99  | 0x063               |          | TMR0_2_CMPB    | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL50-55        | VSSEL131[3]         |
| 100 | 0x064               | Timer2_1 | TMR2_1_CMPA    | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL50-55        | VSSEL131[4]         |
| 101 | 0x065               |          | TMR2_1_CMPB    | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL50-55        | VSSEL131[5]         |
| 102 | 0x066               |          | TMR2_1_OVFA    | P                             | ✓                | -              | INTSEL0-31                                                 | INTSEL50-55        | VSSEL131[6]         |
| 103 | 0x067               |          | TMR2_1_OVFB    | P                             | ✓                | -              | INTSEL0-31                                                 | INTSEL50-55        | VSSEL131[7]         |



| SN  | Interrupt Event No. | Modules  | Interrupt Name | Interrupt Type <sup>(1)</sup> | Interrupt Source | Trigger Source | NVIC Interrupt Request (IRQ) Number Mapping <sup>(2)</sup> |                    |                     |
|-----|---------------------|----------|----------------|-------------------------------|------------------|----------------|------------------------------------------------------------|--------------------|---------------------|
|     |                     |          |                |                               |                  |                | NVIC Vector 0-31                                           | NVIC Vector 32-127 | NVIC Vector 128-143 |
| 104 | 0x068               | Timer2_2 | TMR2_2_CMPA    | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL50-55        | VSSEL131[8]         |
| 105 | 0x069               |          | TMR2_2_CMPB    | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL50-55        | VSSEL131[9]         |
| 106 | 0x06A               |          | TMR2_2_OVFA    | P                             | ✓                | -              | INTSEL0-31                                                 | INTSEL50-55        | VSSEL131[10]        |
| 107 | 0x06B               |          | TMR2_2_OVFB    | P                             | ✓                | -              | INTSEL0-31                                                 | INTSEL50-55        | VSSEL131[11]        |
| 108 | 0x06C               | Timer2_3 | TMR2_3_CMPA    | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL50-55        | VSSEL131[12]        |
| 109 | 0x06D               |          | TMR2_3_CMPB    | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL50-55        | VSSEL131[13]        |
| 110 | 0x06E               |          | TMR2_3_OVFA    | P                             | ✓                | -              | INTSEL0-31                                                 | INTSEL50-55        | VSSEL131[14]        |
| 111 | 0x06F               |          | TMR2_3_OVFB    | P                             | ✓                | -              | INTSEL0-31                                                 | INTSEL50-55        | VSSEL131[15]        |
| 112 | 0x070               | Timer2_4 | TMR2_4_CMPA    | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL50-55        | VSSEL131[16]        |
| 113 | 0x071               |          | TMR2_4_CMPB    | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL50-55        | VSSEL131[17]        |
| 114 | 0x072               |          | TMR2_4_OVFA    | P                             | ✓                | -              | INTSEL0-31                                                 | INTSEL50-55        | VSSEL131[18]        |
| 115 | 0x073               |          | TMR2_4_OVFB    | P                             | ✓                | -              | INTSEL0-31                                                 | INTSEL50-55        | VSSEL131[19]        |
| 116 | 0x074               | -        | -              | -                             | -                | -              | INTSEL0-31                                                 | INTSEL50-55        | VSSEL131[20]        |
| 117 | 0x075               | -        | -              | -                             | -                | -              | INTSEL0-31                                                 | INTSEL50-55        | VSSEL131[21]        |
| 118 | 0x076               | -        | -              | -                             | -                | -              | INTSEL0-31                                                 | INTSEL50-55        | VSSEL131[22]        |
| 119 | 0x077               | -        | -              | -                             | -                | -              | INTSEL0-31                                                 | INTSEL50-55        | VSSEL131[23]        |
| 120 | 0x078               | RTC      | RTC_TP         | P                             | ✓                | -              | INTSEL0-31                                                 | INTSEL50-55        | VSSEL131[24]        |
| 121 | 0x079               |          | RTC_ALM        | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL50-55        | VSSEL131[25]        |
| 122 | 0x07A               |          | RTC_PRD        | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL50-55        | VSSEL131[26]        |
| 123 | 0x07B               | -        | -              | -                             | -                | -              | INTSEL0-31                                                 | INTSEL50-55        | VSSEL131[27]        |
| 124 | 0x07C               | -        | -              | -                             | -                | -              | INTSEL0-31                                                 | INTSEL50-55        | VSSEL131[28]        |
| 125 | 0x07D               | XTAL     | XTAL_STOP      | L                             | ✓                | -              | INTSEL0-31                                                 | INTSEL50-55        | VSSEL131[29]        |
| 126 | 0x07E               | WKTm     | WKTm_PRD       | P                             | ✓                | -              | INTSEL0-31                                                 | INTSEL50-55        | VSSEL131[30]        |
| 127 | 0x07F               | SWDT     | SWDT_REFUFD    | P                             | ✓                | -              | INTSEL0-31                                                 | INTSEL50-55        | VSSEL131[31]        |
| 128 | 0x080               | Timer6_1 | TMR6_1_GCMA    | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL56-61        | VSSEL132[0]         |
| 129 | 0x081               |          | TMR6_1_GCMB    | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL56-61        | VSSEL132[1]         |
| 130 | 0x082               |          | TMR6_1_GCMC    | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL56-61        | VSSEL132[2]         |
| 131 | 0x083               |          | TMR6_1_GCMD    | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL56-61        | VSSEL132[3]         |
| 132 | 0x084               |          | TMR6_1_GCME    | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL56-61        | VSSEL132[4]         |
| 133 | 0x085               |          | TMR6_1_GCMF    | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL56-61        | VSSEL132[5]         |
| 134 | 0x086               |          | TMR6_1_GOVF    | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL56-61        | VSSEL132[6]         |
| 135 | 0x087               |          | TMR6_1_GUDF    | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL56-61        | VSSEL132[7]         |

| SN  | Interrupt Event No. | Modules  | Interrupt Name | Interrupt Type <sup>(1)</sup> | Interrupt Source | Trigger Source | NVIC Interrupt Request (IRQ) Number Mapping <sup>(2)</sup> |                    |                     |
|-----|---------------------|----------|----------------|-------------------------------|------------------|----------------|------------------------------------------------------------|--------------------|---------------------|
|     |                     |          |                |                               |                  |                | NVIC Vector 0-31                                           | NVIC Vector 32-127 | NVIC Vector 128-143 |
| 136 | 0x088               | Timer4_1 | TMR4_1_GCMUH   | L                             | ✓                | -              | INTSEL0-31                                                 | INTSEL56-61        | VSSEL132[8]         |
|     |                     |          | TMR4_1_SCM0    | -                             | -                | ✓              |                                                            |                    |                     |
| 137 | 0x089               |          | TMR4_1_GCMUL   | L                             | ✓                | -              | INTSEL0-31                                                 | INTSEL56-61        | VSSEL132[9]         |
|     |                     |          | TMR4_1_SCM1    | -                             | -                | ✓              |                                                            |                    |                     |
| 138 | 0x08A               |          | TMR4_1_GCMVH   | L                             | ✓                | -              | INTSEL0-31                                                 | INTSEL56-61        | VSSEL132[10]        |
|     |                     |          | TMR4_1_SCM2    | -                             | -                | ✓              |                                                            |                    |                     |
| 139 | 0x08B               |          | TMR4_1_GCMVL   | L                             | ✓                | -              | INTSEL0-31                                                 | INTSEL56-61        | VSSEL132[11]        |
|     |                     |          | TMR4_1_SCM3    | -                             | -                | ✓              |                                                            |                    |                     |
| 140 | 0x08C               |          | TMR4_1_GCMWH   | L                             | ✓                | -              | INTSEL0-31                                                 | INTSEL56-61        | VSSEL132[12]        |
|     |                     |          | TMR4_1_SCM4    | -                             | -                | ✓              |                                                            |                    |                     |
| 141 | 0x08D               |          | TMR4_1_GCMWL   | L                             | ✓                | -              | INTSEL0-31                                                 | INTSEL56-61        | VSSEL132[13]        |
|     |                     |          | TMR4_1_SCM5    | -                             | -                | ✓              |                                                            |                    |                     |
| 142 | 0x08E               | Timer6_2 | TMR4_1_GOVF    | L                             | ✓                | -              | INTSEL0-31                                                 | INTSEL56-61        | VSSEL132[14]        |
| 143 | 0x08F               |          | TMR4_1_GUDF    | L                             | ✓                | -              | INTSEL0-31                                                 | INTSEL56-61        | VSSEL132[15]        |
| 144 | 0x090               |          | TMR6_2_GCMA    | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL56-61        | VSSEL132[16]        |
| 145 | 0x091               |          | TMR6_2_GCMB    | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL56-61        | VSSEL132[17]        |
| 146 | 0x092               |          | TMR6_2_GCMC    | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL56-61        | VSSEL132[18]        |
| 147 | 0x093               |          | TMR6_2_GCMD    | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL56-61        | VSSEL132[19]        |
| 148 | 0x094               |          | TMR6_2_GCME    | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL56-61        | VSSEL132[20]        |
| 149 | 0x095               |          | TMR6_2_GCMF    | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL56-61        | VSSEL132[21]        |
| 150 | 0x096               |          | TMR6_2_GOVF    | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL56-61        | VSSEL132[22]        |
| 151 | 0x097               |          | TMR6_2_GUDF    | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL56-61        | VSSEL132[23]        |
| 152 | 0x098               | Timer4_2 | TMR4_2_GCMUH   | L                             | ✓                | -              | INTSEL0-31                                                 | INTSEL56-61        | VSSEL132[24]        |
|     |                     |          | TMR4_2_SCM0    | -                             | -                | ✓              |                                                            |                    |                     |
| 153 | 0x099               |          | TMR4_2_GCMUL   | L                             | ✓                | -              | INTSEL0-31                                                 | INTSEL56-61        | VSSEL132[25]        |
|     |                     |          | TMR4_2_SCM1    | -                             | -                | ✓              |                                                            |                    |                     |
| 154 | 0x09A               |          | TMR4_2_GCMVH   | L                             | ✓                | -              | INTSEL0-31                                                 | INTSEL56-61        | VSSEL132[26]        |
|     |                     |          | TMR4_2_SCM2    | -                             | -                | ✓              |                                                            |                    |                     |
| 155 | 0x09B               |          | TMR4_2_GCMVL   | L                             | ✓                | -              | INTSEL0-31                                                 | INTSEL56-61        | VSSEL132[27]        |
|     |                     |          | TMR4_2_SCM3    | -                             | -                | ✓              |                                                            |                    |                     |
| 156 | 0x09C               |          | TMR4_2_GCMWH   | L                             | ✓                | -              | INTSEL0-31                                                 | INTSEL56-61        | VSSEL132[28]        |
|     |                     |          | TMR4_2_SCM4    | -                             | -                | ✓              |                                                            |                    |                     |
| 157 | 0x09D               |          | TMR4_2_GCMWL   | L                             | ✓                | -              | INTSEL0-31                                                 | INTSEL56-61        | VSSEL132[29]        |
|     |                     |          | TMR4_2_SCM5    | -                             | -                | ✓              |                                                            |                    |                     |
| 158 | 0x09E               | Timer6_3 | TMR4_2_GOVF    | L                             | ✓                | -              | INTSEL0-31                                                 | INTSEL56-61        | VSSEL132[30]        |
| 159 | 0x09F               |          | TMR4_2_GUDF    | L                             | ✓                | -              | INTSEL0-31                                                 | INTSEL56-61        | VSSEL132[31]        |
| 160 | 0x0A0               |          | TMR6_3_GCMA    | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL62-67        | VSSEL133[0]         |
| 161 | 0x0A1               |          | TMR6_3_GCMB    | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL62-67        | VSSEL133[1]         |
| 162 | 0x0A2               |          | TMR6_3_GCMC    | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL62-67        | VSSEL133[2]         |
| 163 | 0x0A3               |          | TMR6_3_GCMD    | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL62-67        | VSSEL133[3]         |
| 164 | 0x0A4               |          | TMR6_3_GCME    | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL62-67        | VSSEL133[4]         |
| 165 | 0x0A5               |          | TMR6_3_GCMF    | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL62-67        | VSSEL133[5]         |



| SN  | Interrupt Event No. | Modules  | Interrupt Name | Interrupt Type <sup>(1)</sup> | Interrupt Source | Trigger Source | NVIC Interrupt Request (IRQ) Number Mapping <sup>(2)</sup> |                    |                     |
|-----|---------------------|----------|----------------|-------------------------------|------------------|----------------|------------------------------------------------------------|--------------------|---------------------|
|     |                     |          |                |                               |                  |                | NVIC Vector 0-31                                           | NVIC Vector 32-127 | NVIC Vector 128-143 |
| 166 | 0x0A6               | Timer6_3 | TMR6_3_GOVF    | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL62-67        | VSSEL133[6]         |
| 167 | 0x0A7               |          | TMR6_3_GUDF    | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL62-67        | VSSEL133[7]         |
| 168 | 0x0A8               | Timer4_3 | TMR4_3_GCMUH   | L                             | ✓                | -              | INTSEL0-31                                                 | INTSEL62-67        | VSSEL133[8]         |
|     |                     |          | TMR4_3_SCM0    | -                             | -                | ✓              |                                                            |                    |                     |
| 169 | 0x0A9               |          | TMR4_3_GCMUL   | L                             | ✓                | -              | INTSEL0-31                                                 | INTSEL62-67        | VSSEL133[9]         |
|     |                     |          | TMR4_3_SCM1    | -                             | -                | ✓              |                                                            |                    |                     |
| 170 | 0x0AA               |          | TMR4_3_GCMVH   | L                             | ✓                | -              | INTSEL0-31                                                 | INTSEL62-67        | VSSEL133[10]        |
|     |                     |          | TMR4_3_SCM2    | -                             | -                | ✓              |                                                            |                    |                     |
| 171 | 0x0AB               |          | TMR4_3_GCMVL   | L                             | ✓                | -              | INTSEL0-31                                                 | INTSEL62-67        | VSSEL133[11]        |
|     |                     |          | TMR4_3_SCM3    | -                             | -                | ✓              |                                                            |                    |                     |
| 172 | 0x0AC               |          | TMR4_3_GCMWH   | L                             | ✓                | -              | INTSEL0-31                                                 | INTSEL62-67        | VSSEL133[12]        |
|     |                     |          | TMR4_3_SCM4    | -                             | -                | ✓              |                                                            |                    |                     |
| 173 | 0x0AD               |          | TMR4_3_GCMWL   | L                             | ✓                | -              | INTSEL0-31                                                 | INTSEL62-67        | VSSEL133[13]        |
|     |                     |          | TMR4_3_SCM5    | -                             | -                | ✓              |                                                            |                    |                     |
| 174 | 0x0AE               |          | TMR4_3_GOVF    | L                             | ✓                | -              | INTSEL0-31                                                 | INTSEL62-67        | VSSEL133[14]        |
| 175 | 0x0AF               |          | TMR4_3_GUDF    | L                             | ✓                | -              | INTSEL0-31                                                 | INTSEL62-67        | VSSEL133[15]        |
| 176 | 0x0B0               | Timer6_1 | TMR6_1_GDTE    | P                             | ✓                | -              | INTSEL0-31                                                 | INTSEL62-67        | VSSEL133[16]        |
| 177 | 0x0B1               | -        | -              | -                             | -                | -              | INTSEL0-31                                                 | INTSEL62-67        | VSSEL133[17]        |
| 178 | 0x0B2               | -        | -              | -                             | -                | -              | INTSEL0-31                                                 | INTSEL62-67        | VSSEL133[18]        |
| 179 | 0x0B3               | Timer6_1 | TMR6_1_SCMA    | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL62-67        | VSSEL133[19]        |
| 180 | 0x0B4               |          | TMR6_1_SCMB    | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL62-67        | VSSEL133[20]        |
| 181 | 0x0B5               | Timer4_1 | TMR4_1_RLOU    | L                             | ✓                | -              | INTSEL0-31                                                 | INTSEL62-67        | VSSEL133[21]        |
| 182 | 0x0B6               |          | TMR4_1_RLOV    | L                             | ✓                | -              | INTSEL0-31                                                 | INTSEL62-67        | VSSEL133[22]        |
| 183 | 0x0B7               |          | TMR4_1_RLOW    | L                             | ✓                | -              | INTSEL0-31                                                 | INTSEL62-67        | VSSEL133[23]        |
| 184 | 0x0B8               | Timer6_2 | TMR6_2_GDTE    | P                             | ✓                | -              | INTSEL0-31                                                 | INTSEL62-67        | VSSEL133[24]        |
| 185 | 0x0B9               | -        | -              | -                             | -                | -              | INTSEL0-31                                                 | INTSEL62-67        | VSSEL133[25]        |
| 186 | 0x0BA               | -        | -              | -                             | -                | -              | INTSEL0-31                                                 | INTSEL62-67        | VSSEL133[26]        |
| 187 | 0x0BB               | Timer6_2 | TMR6_2_SCMA    | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL62-67        | VSSEL133[27]        |
| 188 | 0x0BC               |          | TMR6_2_SCMB    | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL62-67        | VSSEL133[28]        |
| 189 | 0x0BD               | Timer4_2 | TMR4_2_RLOU    | L                             | ✓                | -              | INTSEL0-31                                                 | INTSEL62-67        | VSSEL133[29]        |
| 190 | 0x0BE               |          | TMR4_2_RLOV    | L                             | ✓                | -              | INTSEL0-31                                                 | INTSEL62-67        | VSSEL133[30]        |
| 191 | 0x0BF               |          | TMR4_2_RLOW    | L                             | ✓                | -              | INTSEL0-31                                                 | INTSEL62-67        | VSSEL133[31]        |
| 192 | 0x0C0               | Timer6_3 | TMR6_3_GDTE    | P                             | ✓                | -              | INTSEL0-31                                                 | INTSEL68-73        | VSSEL134[0]         |
| 193 | 0x0C1               | -        | -              | -                             | -                | -              | INTSEL0-31                                                 | INTSEL68-73        | VSSEL134[1]         |
| 194 | 0x0C2               | -        | -              | -                             | -                | -              | INTSEL0-31                                                 | INTSEL68-73        | VSSEL134[2]         |
| 195 | 0x0C3               | Timer6_3 | TMR6_3_SCMA    | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL68-73        | VSSEL134[3]         |
| 196 | 0x0C4               |          | TMR6_3_SCMB    | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL68-73        | VSSEL134[4]         |
| 197 | 0x0C5               | Timer4_3 | TMR4_3_RLOU    | L                             | ✓                | -              | INTSEL0-31                                                 | INTSEL68-73        | VSSEL134[5]         |
| 198 | 0x0C6               |          | TMR4_3_RLOV    | L                             | ✓                | -              | INTSEL0-31                                                 | INTSEL68-73        | VSSEL134[6]         |
| 199 | 0x0C7               |          | TMR4_3_RLOW    | L                             | ✓                | -              | INTSEL0-31                                                 | INTSEL68-73        | VSSEL134[7]         |
| 200 | 0x0C8               | USART_3  | USART_3_RTO    | L                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL68-73        | VSSEL134[8]         |
| 201 | 0x0C9               | USART_4  | USART_4_RTO    | L                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL68-73        | VSSEL134[9]         |

| SN  | Interrupt Event No. | Modules  | Interrupt Name | Interrupt Type <sup>(1)</sup> | Interrupt Source | Trigger Source | NVIC Interrupt Request (IRQ) Number Mapping <sup>(2)</sup> |                    |                     |
|-----|---------------------|----------|----------------|-------------------------------|------------------|----------------|------------------------------------------------------------|--------------------|---------------------|
|     |                     |          |                |                               |                  |                | NVIC Vector 0-31                                           | NVIC Vector 32-127 | NVIC Vector 128-143 |
| 202 | 0x0CA               | USART_5  | USART_5_RTO    | L                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL68-73        | VSSEL134[10]        |
| 203 | 0x0CB               | USART_8  | USART_8_RTO    | L                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL68-73        | VSSEL134[11]        |
| 204 | 0x0CC               | USART_9  | USART_9_RTO    | L                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL68-73        | VSSEL134[12]        |
| 205 | 0x0CD               | USART_10 | USART_10_RTO   | L                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL68-73        | VSSEL134[13]        |
| 206 | 0x0CE               | PKE      | PKE            | L                             | ✓                | -              | INTSEL0-31                                                 | INTSEL68-73        | VSSEL134[14]        |
| 207 | 0x0CF               | PKERAM   | PKERAM_ECCER   | L                             | ✓                | -              | INTSEL0-31                                                 | INTSEL68-73        | VSSEL134[15]        |
| 208 | 0x0D0               | Timer6_4 | TMR6_4_GCMA    | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL68-73        | VSSEL134[16]        |
| 209 | 0x0D1               |          | TMR6_4_GCMB    | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL68-73        | VSSEL134[17]        |
| 210 | 0x0D2               |          | TMR6_4_GCMC    | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL68-73        | VSSEL134[18]        |
| 211 | 0x0D3               |          | TMR6_4_GCMD    | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL68-73        | VSSEL134[19]        |
| 212 | 0x0D4               |          | TMR6_4_GCME    | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL68-73        | VSSEL134[20]        |
| 213 | 0x0D5               |          | TMR6_4_GCMF    | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL68-73        | VSSEL134[21]        |
| 214 | 0x0D6               |          | TMR6_4_GOVF    | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL68-73        | VSSEL134[22]        |
| 215 | 0x0D7               |          | TMR6_4_GUDF    | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL68-73        | VSSEL134[23]        |
| 216 | 0x0D8               |          | TMR6_4_GDTE    | P                             | ✓                | -              | INTSEL0-31                                                 | INTSEL68-73        | VSSEL134[24]        |
| 217 | 0x0D9               | -        | -              | -                             | -                | -              | INTSEL0-31                                                 | INTSEL68-73        | VSSEL134[25]        |
| 218 | 0x0DA               | -        | -              | -                             | -                | -              | INTSEL0-31                                                 | INTSEL68-73        | VSSEL134[26]        |
| 219 | 0x0DB               | Timer6_4 | TMR6_4_SCMA    | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL68-73        | VSSEL134[27]        |
| 220 | 0x0DC               |          | TMR6_4_SCMB    | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL68-73        | VSSEL134[28]        |
| 221 | 0x0DD               | -        | -              | -                             | -                | -              | INTSEL0-31                                                 | INTSEL68-73        | VSSEL134[29]        |
| 222 | 0x0DE               | -        | -              | -                             | -                | -              | INTSEL0-31                                                 | INTSEL68-73        | VSSEL134[30]        |
| 223 | 0x0DF               | -        | -              | -                             | -                | -              | INTSEL0-31                                                 | INTSEL68-73        | VSSEL134[31]        |
| 224 | 0x0E0               | Timer6_5 | TMR6_5_GCMA    | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL74-79        | VSSEL135[0]         |
| 225 | 0x0E1               |          | TMR6_5_GCMB    | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL74-79        | VSSEL135[1]         |
| 226 | 0x0E2               |          | TMR6_5_GCMC    | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL74-79        | VSSEL135[2]         |
| 227 | 0x0E3               |          | TMR6_5_GCMD    | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL74-79        | VSSEL135[3]         |
| 228 | 0x0E4               |          | TMR6_5_GCME    | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL74-79        | VSSEL135[4]         |
| 229 | 0x0E5               |          | TMR6_5_GCMF    | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL74-79        | VSSEL135[5]         |
| 230 | 0x0E6               |          | TMR6_5_GOVF    | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL74-79        | VSSEL135[6]         |
| 231 | 0x0E7               |          | TMR6_5_GUDF    | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL74-79        | VSSEL135[7]         |
| 232 | 0x0E8               |          | TMR6_5_GDTE    | P                             | ✓                | -              | INTSEL0-31                                                 | INTSEL74-79        | VSSEL135[8]         |
| 233 | 0x0E9               | -        | -              | -                             | -                | -              | INTSEL0-31                                                 | INTSEL74-79        | VSSEL135[9]         |
| 234 | 0x0EA               | -        | -              | -                             | -                | -              | INTSEL0-31                                                 | INTSEL74-79        | VSSEL135[10]        |
| 235 | 0x0EB               | Timer6_5 | TMR6_5_SCMA    | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL74-79        | VSSEL135[11]        |
| 236 | 0x0EC               |          | TMR6_5_SCMB    | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL74-79        | VSSEL135[12]        |
| 237 | 0x0ED               | TimerA_1 | TMRA_1_OVF     | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL74-79        | VSSEL135[13]        |
| 238 | 0x0EE               |          | TMRA_1_UDF     | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL74-79        | VSSEL135[14]        |
| 239 | 0x0EF               |          | TMRA_1_CMP     | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL74-79        | VSSEL135[15]        |
| 240 | 0x0F0               | Timer6_6 | TMR6_6_GCMA    | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL74-79        | VSSEL135[16]        |
| 241 | 0x0F1               |          | TMR6_6_GCMB    | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL74-79        | VSSEL135[17]        |
| 242 | 0x0F2               |          | TMR6_6_GCMC    | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL74-79        | VSSEL135[18]        |
| 243 | 0x0F3               |          | TMR6_6_GCMD    | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL74-79        | VSSEL135[19]        |

| SN  | Interrupt Event No. | Modules  | Interrupt Name | Interrupt Type <sup>(1)</sup> | Interrupt Source | Trigger Source | NVIC Interrupt Request (IRQ) Number Mapping <sup>(2)</sup> |                    |                     |
|-----|---------------------|----------|----------------|-------------------------------|------------------|----------------|------------------------------------------------------------|--------------------|---------------------|
|     |                     |          |                |                               |                  |                | NVIC Vector 0-31                                           | NVIC Vector 32-127 | NVIC Vector 128-143 |
| 244 | 0x0F4               | Timer6_6 | TMR6_6_GCME    | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL74-79        | VSSEL135[20]        |
| 245 | 0x0F5               |          | TMR6_6_GCMF    | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL74-79        | VSSEL135[21]        |
| 246 | 0x0F6               |          | TMR6_6_GOVF    | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL74-79        | VSSEL135[22]        |
| 247 | 0x0F7               |          | TMR6_6_GUDF    | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL74-79        | VSSEL135[23]        |
| 248 | 0x0F8               |          | TMR6_6_GDTE    | P                             | ✓                | -              | INTSEL0-31                                                 | INTSEL74-79        | VSSEL135[24]        |
| 249 | 0x0F9               | -        | -              | -                             | -                | -              | INTSEL0-31                                                 | INTSEL74-79        | VSSEL135[25]        |
| 250 | 0x0FA               | -        | -              | -                             | -                | -              | INTSEL0-31                                                 | INTSEL74-79        | VSSEL135[26]        |
| 251 | 0x0FB               | Timer6_6 | TMR6_6_SCMA    | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL74-79        | VSSEL135[27]        |
| 252 | 0x0FC               |          | TMR6_6_SCMB    | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL74-79        | VSSEL135[28]        |
| 253 | 0x0FD               | TimerA_2 | TMRA_2_OVF     | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL74-79        | VSSEL135[29]        |
| 254 | 0x0FE               |          | TMRA_2_UDF     | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL74-79        | VSSEL135[30]        |
| 255 | 0x0FF               |          | TMRA_2_CMP     | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL74-79        | VSSEL135[31]        |
| 256 | 0x100               | Timer6_7 | TMR6_7_GCMA    | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL80-85        | VSSEL136[0]         |
| 257 | 0x101               |          | TMR6_7_GCMB    | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL80-85        | VSSEL136[1]         |
| 258 | 0x102               |          | TMR6_7_GCMC    | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL80-85        | VSSEL136[2]         |
| 259 | 0x103               |          | TMR6_7_GCMD    | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL80-85        | VSSEL136[3]         |
| 260 | 0x104               |          | TMR6_7_GCME    | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL80-85        | VSSEL136[4]         |
| 261 | 0x105               |          | TMR6_7_GCMF    | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL80-85        | VSSEL136[5]         |
| 262 | 0x106               |          | TMR6_7_GOVF    | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL80-85        | VSSEL136[6]         |
| 263 | 0x107               |          | TMR6_7_GUDF    | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL80-85        | VSSEL136[7]         |
| 264 | 0x108               |          | TMR6_7_GDTE    | P                             | ✓                | -              | INTSEL0-31                                                 | INTSEL80-85        | VSSEL136[8]         |
| 265 | 0x109               | -        | -              | -                             | -                | -              | INTSEL0-31                                                 | INTSEL80-85        | VSSEL136[9]         |
| 266 | 0x10A               | -        | -              | -                             | -                | -              | INTSEL0-31                                                 | INTSEL80-85        | VSSEL136[10]        |
| 267 | 0x10B               | Timer6_7 | TMR6_7_SCMA    | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL80-85        | VSSEL136[11]        |
| 268 | 0x10C               |          | TMR6_7_SCMB    | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL80-85        | VSSEL136[12]        |
| 269 | 0x10D               | TimerA_3 | TMRA_3_OVF     | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL80-85        | VSSEL136[13]        |
| 270 | 0x10E               |          | TMRA_3_UDF     | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL80-85        | VSSEL136[14]        |
| 271 | 0x10F               |          | TMRA_3_CMP     | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL80-85        | VSSEL136[15]        |
| 272 | 0x110               | Timer6_8 | TMR6_8_GCMA    | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL80-85        | VSSEL136[16]        |
| 273 | 0x111               |          | TMR6_8_GCMB    | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL80-85        | VSSEL136[17]        |
| 274 | 0x112               |          | TMR6_8_GCMC    | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL80-85        | VSSEL136[18]        |
| 275 | 0x113               |          | TMR6_8_GCMD    | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL80-85        | VSSEL136[19]        |
| 276 | 0x114               |          | TMR6_8_GCME    | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL80-85        | VSSEL136[20]        |
| 277 | 0x115               |          | TMR6_8_GCMF    | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL80-85        | VSSEL136[21]        |
| 278 | 0x116               |          | TMR6_8_GOVF    | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL80-85        | VSSEL136[22]        |
| 279 | 0x117               |          | TMR6_8_GUDF    | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL80-85        | VSSEL136[23]        |
| 280 | 0x118               |          | TMR6_8_GDTE    | P                             | ✓                | -              | INTSEL0-31                                                 | INTSEL80-85        | VSSEL136[24]        |
| 281 | 0x119               | -        | -              | -                             | -                | -              | INTSEL0-31                                                 | INTSEL80-85        | VSSEL136[25]        |
| 282 | 0x11A               | -        | -              | -                             | -                | -              | INTSEL0-31                                                 | INTSEL80-85        | VSSEL136[26]        |
| 283 | 0x11B               | Timer6_8 | TMR6_8_SCMA    | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL80-85        | VSSEL136[27]        |
| 284 | 0x11C               |          | TMR6_8_SCMB    | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL80-85        | VSSEL136[28]        |

| SN  | Interrupt Event No. | Modules                 | Interrupt Name | Interrupt Type <sup>(1)</sup> | Interrupt Source | Trigger Source | NVIC Interrupt Request (IRQ) Number Mapping <sup>(2)</sup> |                    |                     |
|-----|---------------------|-------------------------|----------------|-------------------------------|------------------|----------------|------------------------------------------------------------|--------------------|---------------------|
|     |                     |                         |                |                               |                  |                | NVIC Vector 0-31                                           | NVIC Vector 32-127 | NVIC Vector 128-143 |
| 285 | 0x11D               | TimerA_4                | TMRA_4_OVF     | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL80-85        | VSSEL136[29]        |
| 286 | 0x11E               |                         | TMRA_4_UDF     | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL80-85        | VSSEL136[30]        |
| 287 | 0x11F               |                         | TMRA_4_CMP     | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL80-85        | VSSEL136[31]        |
| 288 | 0x120               | EMB                     | EMB_GR0        | L                             | ✓                | -              | INTSEL0-31                                                 | INTSEL86-91        | VSSEL137[0]         |
| 289 | 0x121               |                         | EMB_GR1        | L                             | ✓                | -              | INTSEL0-31                                                 | INTSEL86-91        | VSSEL137[1]         |
| 290 | 0x122               |                         | EMB_GR2        | L                             | ✓                | -              | INTSEL0-31                                                 | INTSEL86-91        | VSSEL137[2]         |
| 291 | 0x123               |                         | EMB_GR3        | L                             | ✓                | -              | INTSEL0-31                                                 | INTSEL86-91        | VSSEL137[3]         |
| 292 | 0x124               |                         | EMB_GR4        | L                             | ✓                | -              | INTSEL0-31                                                 | INTSEL86-91        | VSSEL137[4]         |
| 293 | 0x125               |                         | EMB_GR5        | L                             | ✓                | -              | INTSEL0-31                                                 | INTSEL86-91        | VSSEL137[5]         |
| 294 | 0x126               |                         | EMB_GR6        | L                             | ✓                | -              | INTSEL0-31                                                 | INTSEL86-91        | VSSEL137[6]         |
| 295 | 0x127               | USBHS                   | USBHS_EP1_OUT  | L                             | ✓                | -              | INTSEL0-31                                                 | INTSEL86-91        | VSSEL137[7]         |
| 296 | 0x128               |                         | USBHS_EP1_IN   | L                             | ✓                | -              | INTSEL0-31                                                 | INTSEL86-91        | VSSEL137[8]         |
| 297 | 0x129               |                         | USBHS_GLB      | L                             | ✓                | -              | INTSEL0-31                                                 | INTSEL86-91        | VSSEL137[9]         |
| 298 | 0x12A               |                         | USBHS_WKUP     | P                             | ✓                | -              | INTSEL0-31                                                 | INTSEL86-91        | VSSEL137[10]        |
| 299 | 0x12B               | AOS_STRG <sup>(3)</sup> | AOS_STRG       | -                             | -                | ✓              | INTSEL0-31                                                 | INTSEL86-91        | VSSEL137[11]        |
| 300 | 0x12C               | USART_1                 | USART_1_EI     | L                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL86-91        | VSSEL137[12]        |
| 301 | 0x12D               |                         | USART_1_RI     | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL86-91        | VSSEL137[13]        |
| 302 | 0x12E               |                         | USART_1_TI     | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL86-91        | VSSEL137[14]        |
| 303 | 0x12F               |                         | USART_1_TCI    | L                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL86-91        | VSSEL137[15]        |
| 304 | 0x130               |                         | USART_1_RTO    | L                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL86-91        | VSSEL137[16]        |
| 305 | 0x131               | USART_2                 | USART_2_EI     | L                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL86-91        | VSSEL137[17]        |
| 306 | 0x132               |                         | USART_2_RI     | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL86-91        | VSSEL137[18]        |
| 307 | 0x133               |                         | USART_2_TI     | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL86-91        | VSSEL137[19]        |
| 308 | 0x134               |                         | USART_2_TCI    | L                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL86-91        | VSSEL137[20]        |
| 309 | 0x135               |                         | USART_2_RTO    | L                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL86-91        | VSSEL137[21]        |
| 310 | 0x136               | SPI_1                   | SPI_1_SPRI     | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL86-91        | VSSEL137[22]        |
| 311 | 0x137               |                         | SPI_1_SPTI     | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL86-91        | VSSEL137[23]        |
| 312 | 0x138               |                         | SPI_1_SPII     | L                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL86-91        | VSSEL137[24]        |
| 313 | 0x139               |                         | SPI_1_SPEI     | L                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL86-91        | VSSEL137[25]        |
| 314 | 0x13A               |                         | SPI_1_SPEND    | -                             | -                | ✓              | INTSEL0-31                                                 | INTSEL86-91        | VSSEL137[26]        |
| 315 | 0x13B               | SPI_2                   | SPI_2_SPRI     | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL86-91        | VSSEL137[27]        |
| 316 | 0x13C               |                         | SPI_2_SPTI     | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL86-91        | VSSEL137[28]        |
| 317 | 0x13D               |                         | SPI_2_SPII     | L                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL86-91        | VSSEL137[29]        |
| 318 | 0x13E               |                         | SPI_2_SPEI     | L                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL86-91        | VSSEL137[30]        |
| 319 | 0x13F               |                         | SPI_2_SPEND    | -                             | -                | ✓              | INTSEL0-31                                                 | INTSEL86-91        | VSSEL137[31]        |
| 320 | 0x140               | TimerA_5                | TMRA_5_OVF     | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL92-97        | VSSEL138[0]         |
| 321 | 0x141               |                         | TMRA_5_UDF     | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL92-97        | VSSEL138[1]         |
| 322 | 0x142               |                         | TMRA_5_CMP     | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL92-97        | VSSEL138[2]         |
| 323 | 0x143               | TimerA_6                | TMRA_6_OVF     | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL92-97        | VSSEL138[3]         |
| 324 | 0x144               |                         | TMRA_6_UDF     | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL92-97        | VSSEL138[4]         |
| 325 | 0x145               |                         | TMRA_6_CMP     | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL92-97        | VSSEL138[5]         |

| SN  | Interrupt Event No. | Modules   | Interrupt Name | Interrupt Type <sup>(1)</sup> | Interrupt Source | Trigger Source | NVIC Interrupt Request (IRQ) Number Mapping <sup>(2)</sup> |                    |                     |
|-----|---------------------|-----------|----------------|-------------------------------|------------------|----------------|------------------------------------------------------------|--------------------|---------------------|
|     |                     |           |                |                               |                  |                | NVIC Vector 0-31                                           | NVIC Vector 32-127 | NVIC Vector 128-143 |
| 326 | 0x146               | TimerA_7  | TMRA_7_OVF     | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL92-97        | VSSEL138[6]         |
| 327 | 0x147               |           | TMRA_7_UDF     | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL92-97        | VSSEL138[7]         |
| 328 | 0x148               |           | TMRA_7_CMP     | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL92-97        | VSSEL138[8]         |
| 329 | 0x149               | TimerA_8  | TMRA_8_OVF     | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL92-97        | VSSEL138[9]         |
| 330 | 0x14A               |           | TMRA_8_UDF     | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL92-97        | VSSEL138[10]        |
| 331 | 0x14B               |           | TMRA_8_CMP     | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL92-97        | VSSEL138[11]        |
| 332 | 0x14C               | USART_3   | USART_3_EI     | L                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL92-97        | VSSEL138[12]        |
| 333 | 0x14D               |           | USART_3_RI     | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL92-97        | VSSEL138[13]        |
| 334 | 0x14E               |           | USART_3_TI     | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL92-97        | VSSEL138[14]        |
| 335 | 0x14F               |           | USART_3_TCI    | L                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL92-97        | VSSEL138[15]        |
| 336 | 0x150               | USART_4   | USART_4_EI     | L                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL92-97        | VSSEL138[16]        |
| 337 | 0x151               |           | USART_4_RI     | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL92-97        | VSSEL138[17]        |
| 338 | 0x152               |           | USART_4_TI     | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL92-97        | VSSEL138[18]        |
| 339 | 0x153               |           | USART_4_TCI    | L                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL92-97        | VSSEL138[19]        |
| 340 | 0x154               | CAN_1     | CAN_1_HOST     | L                             | ✓                | -              | INTSEL0-31                                                 | INTSEL92-97        | VSSEL138[20]        |
| 341 | 0x155               | CAN_2     | CAN_2_HOST     | L                             | ✓                | -              | INTSEL0-31                                                 | INTSEL92-97        | VSSEL138[21]        |
| 342 | 0x156               | SPI_3     | SPI_3_SPRI     | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL92-97        | VSSEL138[22]        |
| 343 | 0x157               |           | SPI_3_SPTI     | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL92-97        | VSSEL138[23]        |
| 344 | 0x158               |           | SPI_3_SPII     | L                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL92-97        | VSSEL138[24]        |
| 345 | 0x159               |           | SPI_3_SPEI     | L                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL92-97        | VSSEL138[25]        |
| 346 | 0x15A               |           | SPI_3_SPEND    | -                             | -                | ✓              | INTSEL0-31                                                 | INTSEL92-97        | VSSEL138[26]        |
| 347 | 0x15B               | SPI_4     | SPI_4_SPRI     | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL92-97        | VSSEL138[27]        |
| 348 | 0x15C               |           | SPI_4_SPTI     | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL92-97        | VSSEL138[28]        |
| 349 | 0x15D               |           | SPI_4_SPII     | L                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL92-97        | VSSEL138[29]        |
| 350 | 0x15E               |           | SPI_4_SPEI     | L                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL92-97        | VSSEL138[30]        |
| 351 | 0x15F               |           | SPI_4_SPEND    | -                             | -                | ✓              | INTSEL0-31                                                 | INTSEL92-97        | VSSEL138[31]        |
| 352 | 0x160               | TimerA_9  | TMRA_9_OVF     | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL98-103       | VSSEL139[0]         |
| 353 | 0x161               |           | TMRA_9_UDF     | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL98-103       | VSSEL139[1]         |
| 354 | 0x162               |           | TMRA_9_CMP     | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL98-103       | VSSEL139[2]         |
| 355 | 0x163               | TimerA_10 | TMRA_10_OVF    | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL98-103       | VSSEL139[3]         |
| 356 | 0x164               |           | TMRA_10_UDF    | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL98-103       | VSSEL139[4]         |
| 357 | 0x165               |           | TMRA_10_CMP    | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL98-103       | VSSEL139[5]         |
| 358 | 0x166               | TimerA_11 | TMRA_11_OVF    | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL98-103       | VSSEL139[6]         |
| 359 | 0x167               |           | TMRA_11_UDF    | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL98-103       | VSSEL139[7]         |
| 360 | 0x168               |           | TMRA_11_CMP    | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL98-103       | VSSEL139[8]         |
| 361 | 0x169               | TimerA_12 | TMRA_12_OVF    | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL98-103       | VSSEL139[9]         |
| 362 | 0x16A               |           | TMRA_12_UDF    | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL98-103       | VSSEL139[10]        |
| 363 | 0x16B               |           | TMRA_12_CMP    | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL98-103       | VSSEL139[11]        |

| SN  | Interrupt Event No. | Modules | Interrupt Name  | Interrupt Type <sup>(1)</sup> | Interrupt Source | Trigger Source | NVIC Interrupt Request (IRQ) Number Mapping <sup>(2)</sup> |                    |                     |
|-----|---------------------|---------|-----------------|-------------------------------|------------------|----------------|------------------------------------------------------------|--------------------|---------------------|
|     |                     |         |                 |                               |                  |                | NVIC Vector 0-31                                           | NVIC Vector 32-127 | NVIC Vector 128-143 |
| 364 | 0x16C               | USART_5 | USART_5_BRKWKPI | L                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL98-103       | VSSEL139[12]        |
| 365 | 0x16D               |         | USART_5_EI      | L                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL98-103       | VSSEL139[13]        |
| 366 | 0x16E               |         | USART_5_RI      | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL98-103       | VSSEL139[14]        |
| 367 | 0x16F               |         | USART_5_TI      | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL98-103       | VSSEL139[15]        |
| 368 | 0x170               |         | USART_5_TCI     | L                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL98-103       | VSSEL139[16]        |
| 369 | 0x171               | USART_6 | USART_6_EI      | L                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL98-103       | VSSEL139[17]        |
| 370 | 0x172               |         | USART_6_RI      | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL98-103       | VSSEL139[18]        |
| 371 | 0x173               |         | USART_6_TI      | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL98-103       | VSSEL139[19]        |
| 372 | 0x174               |         | USART_6_TCI     | L                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL98-103       | VSSEL139[20]        |
| 373 | 0x175               |         | USART_6_RTO     | L                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL98-103       | VSSEL139[21]        |
| 374 | 0x176               | SPI_5   | SPI_5_SPRI      | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL98-103       | VSSEL139[22]        |
| 375 | 0x177               |         | SPI_5_SPTI      | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL98-103       | VSSEL139[23]        |
| 376 | 0x178               |         | SPI_5_SPII      | L                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL98-103       | VSSEL139[24]        |
| 377 | 0x179               |         | SPI_5_SPEI      | L                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL98-103       | VSSEL139[25]        |
| 378 | 0x17A               |         | SPI_5_SPEND     | -                             | -                | ✓              | INTSEL0-31                                                 | INTSEL98-103       | VSSEL139[26]        |
| 379 | 0x17B               | SPI_6   | SPI_6_SPRI      | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL98-103       | VSSEL139[27]        |
| 380 | 0x17C               |         | SPI_6_SPTI      | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL98-103       | VSSEL139[28]        |
| 381 | 0x17D               |         | SPI_6_SPII      | L                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL98-103       | VSSEL139[29]        |
| 382 | 0x17E               |         | SPI_6_SPEI      | L                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL98-103       | VSSEL139[30]        |
| 383 | 0x17F               |         | SPI_6_SPEND     | -                             | -                | ✓              | INTSEL0-31                                                 | INTSEL98-103       | VSSEL139[31]        |
| 384 | 0x180               | I2S_1   | I2S_1_TXIRQOUT  | L                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL104-109      | VSSEL140[0]         |
| 385 | 0x181               |         | I2S_1_RXIRQOUT  | L                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL104-109      | VSSEL140[1]         |
| 386 | 0x182               |         | I2S_1_ERRIRQOUT | L                             | ✓                | -              | INTSEL0-31                                                 | INTSEL104-109      | VSSEL140[2]         |
| 387 | 0x183               | I2S_2   | I2S_2_TXIRQOUT  | L                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL104-109      | VSSEL140[3]         |
| 388 | 0x184               |         | I2S_2_RXIRQOUT  | L                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL104-109      | VSSEL140[4]         |
| 389 | 0x185               |         | I2S_2_ERRIRQOUT | L                             | ✓                | -              | INTSEL0-31                                                 | INTSEL104-109      | VSSEL140[5]         |
| 390 | 0x186               | USART_7 | USART_7_EI      | L                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL104-109      | VSSEL140[6]         |
| 391 | 0x187               |         | USART_7_RI      | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL104-109      | VSSEL140[7]         |
| 392 | 0x188               |         | USART_7_TI      | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL104-109      | VSSEL140[8]         |
| 393 | 0x189               |         | USART_7_TCI     | L                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL104-109      | VSSEL140[9]         |
| 394 | 0x18A               |         | USART_7_RTO     | L                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL104-109      | VSSEL140[10]        |
| 395 | 0x18B               | USART_8 | USART_8_EI      | L                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL104-109      | VSSEL140[11]        |
| 396 | 0x18C               |         | USART_8_RI      | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL104-109      | VSSEL140[12]        |
| 397 | 0x18D               |         | USART_8_TI      | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL104-109      | VSSEL140[13]        |
| 398 | 0x18E               |         | USART_8_TCI     | L                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL104-109      | VSSEL140[14]        |
| 399 | 0x18F               | USBFS   | USBFS_GLB       | L                             | ✓                | -              | INTSEL0-31                                                 | INTSEL104-109      | VSSEL140[15]        |
| 400 | 0x190               |         | USBFS_WKUP      | P                             | ✓                | -              | INTSEL0-31                                                 | INTSEL104-109      | VSSEL140[16]        |
| 401 | 0x191               | HASH    | HASH_INT        | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL104-109      | VSSEL140[17]        |
| 402 | 0x192               | SDIOC_1 | SDIOC_1_DMAR    | -                             | -                | ✓              | INTSEL0-31                                                 | INTSEL104-109      | VSSEL140[18]        |
| 403 | 0x193               |         | SDIOC_1_DMAW    | -                             | -                | ✓              | INTSEL0-31                                                 | INTSEL104-109      | VSSEL140[19]        |
| 404 | 0x194               |         | SDIOC_1_SD      | L                             | ✓                | -              | INTSEL0-31                                                 | INTSEL104-109      | VSSEL140[20]        |



| SN  | Interrupt Event No. | Modules    | Interrupt Name   | Interrupt Type <sup>(1)</sup> | Interrupt Source | Trigger Source | NVIC Interrupt Request (IRQ) Number Mapping <sup>(2)</sup> |                    |                     |
|-----|---------------------|------------|------------------|-------------------------------|------------------|----------------|------------------------------------------------------------|--------------------|---------------------|
|     |                     |            |                  |                               |                  |                | NVIC Vector 0-31                                           | NVIC Vector 32-127 | NVIC Vector 128-143 |
| 405 | 0x195               | SDIOC_2    | SDIOC_2_DMAR     | -                             | -                | ✓              | INTSEL0-31                                                 | INTSEL104-109      | VSSEL140[21]        |
| 406 | 0x196               |            | SDIOC_2_DMAW     | -                             | -                | ✓              | INTSEL0-31                                                 | INTSEL104-109      | VSSEL140[22]        |
| 407 | 0x197               |            | SDIOC_2_SD       | L                             | ✓                | -              | INTSEL0-31                                                 | INTSEL104-109      | VSSEL140[23]        |
| 408 | 0x198               | EVENT port | EVENT_PORT1      | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL104-109      | VSSEL140[24]        |
| 409 | 0x199               |            | EVENT_PORT2      | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL104-109      | VSSEL140[25]        |
| 410 | 0x19A               |            | EVENT_PORT3      | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL104-109      | VSSEL140[26]        |
| 411 | 0x19B               |            | EVENT_PORT4      | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL104-109      | VSSEL140[27]        |
| 412 | 0x19C               | ETHMAC     | ETH_GLB_INT      | L                             | ✓                | -              | INTSEL0-31                                                 | INTSEL104-109      | VSSEL140[28]        |
| 413 | 0x19D               |            | ETH_WKP_INT      | L                             | ✓                | -              | INTSEL0-31                                                 | INTSEL104-109      | VSSEL140[29]        |
| 414 | 0x19E               |            | ETH_PPS_OUT_0    | -                             | -                | ✓              | INTSEL0-31                                                 | INTSEL104-109      | VSSEL140[30]        |
| 415 | 0x19F               |            | ETH_PPS_OUT_1    | -                             | -                | ✓              | INTSEL0-31                                                 | INTSEL104-109      | VSSEL140[31]        |
| 416 | 0x1A0               | I2S_3      | I2S_3_TXIRQOUT   | L                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL110-115      | VSSEL141[0]         |
| 417 | 0x1A1               |            | I2S_3_RXIRQOUT   | L                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL110-115      | VSSEL141[1]         |
| 418 | 0x1A2               |            | I2S_3_ERRIRQOUT  | L                             | ✓                | -              | INTSEL0-31                                                 | INTSEL110-115      | VSSEL141[2]         |
| 419 | 0x1A3               | I2S_4      | I2S_4_TXIRQOUT   | L                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL110-115      | VSSEL141[3]         |
| 420 | 0x1A4               |            | I2S_4_RXIRQOUT   | L                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL110-115      | VSSEL141[4]         |
| 421 | 0x1A5               |            | I2S_4_ERRIRQOUT  | L                             | ✓                | -              | INTSEL0-31                                                 | INTSEL110-115      | VSSEL141[5]         |
| 422 | 0x1A6               | USART_9    | USART_9_EI       | L                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL110-115      | VSSEL141[6]         |
| 423 | 0x1A7               |            | USART_9_RI       | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL110-115      | VSSEL141[7]         |
| 424 | 0x1A8               |            | USART_9_TI       | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL110-115      | VSSEL141[8]         |
| 425 | 0x1A9               |            | USART_9_TCI      | L                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL110-115      | VSSEL141[9]         |
| 426 | 0x1AA               | USART_10   | USART_10_BRKWKPI | L                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL110-115      | VSSEL141[10]        |
| 427 | 0x1AB               |            | USART_10_EI      | L                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL110-115      | VSSEL141[11]        |
| 428 | 0x1AC               |            | USART_10_RI      | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL110-115      | VSSEL141[12]        |
| 429 | 0x1AD               |            | USART_10_TI      | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL110-115      | VSSEL141[13]        |
| 430 | 0x1AE               |            | USART_10_TCI     | L                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL110-115      | VSSEL141[14]        |
| 431 | 0x1AF               | SKE        | SKE              | L                             | ✓                | -              | INTSEL0-31                                                 | INTSEL110-115      | VSSEL141[15]        |
| 432 | 0x1B0               | I2C_1      | I2C_1_RXI        | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL110-115      | VSSEL141[16]        |
| 433 | 0x1B1               |            | I2C_1_TXI        | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL110-115      | VSSEL141[17]        |
| 434 | 0x1B2               |            | I2C_1_TEI        | L                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL110-115      | VSSEL141[18]        |
| 435 | 0x1B3               |            | I2C_1_EEI        | L                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL110-115      | VSSEL141[19]        |
| 436 | 0x1B4               | I2C_2      | I2C_2_RXI        | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL110-115      | VSSEL141[20]        |
| 437 | 0x1B5               |            | I2C_2_TXI        | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL110-115      | VSSEL141[21]        |
| 438 | 0x1B6               |            | I2C_2_TEI        | L                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL110-115      | VSSEL141[22]        |
| 439 | 0x1B7               |            | I2C_2_EEI        | L                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL110-115      | VSSEL141[23]        |
| 440 | 0x1B8               | I2C_3      | I2C_3_RXI        | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL110-115      | VSSEL141[24]        |
| 441 | 0x1B9               |            | I2C_3_TXI        | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL110-115      | VSSEL141[25]        |
| 442 | 0x1BA               |            | I2C_3_TEI        | L                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL110-115      | VSSEL141[26]        |
| 443 | 0x1BB               |            | I2C_3_EEI        | L                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL110-115      | VSSEL141[27]        |

| SN  | Interrupt Event No. | Modules  | Interrupt Name | Interrupt Type <sup>(1)</sup> | Interrupt Source | Trigger Source | NVIC Interrupt Request (IRQ) Number Mapping <sup>(2)</sup> |                    |                     |
|-----|---------------------|----------|----------------|-------------------------------|------------------|----------------|------------------------------------------------------------|--------------------|---------------------|
|     |                     |          |                |                               |                  |                | NVIC Vector 0-31                                           | NVIC Vector 32-127 | NVIC Vector 128-143 |
| 444 | 0x1BC               | CMP      | CMP1           | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL110-115      | VSSEL141[28]        |
| 445 | 0x1BD               |          | CMP2           | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL110-115      | VSSEL141[29]        |
| 446 | 0x1BE               |          | CMP3           | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL110-115      | VSSEL141[30]        |
| 447 | 0x1BF               |          | CMP4           | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL110-115      | VSSEL141[31]        |
| 448 | 0x1C0               | I2C_4    | I2C_4_RXI      | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL116-121      | VSSEL142[0]         |
| 449 | 0x1C1               |          | I2C_4_TXI      | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL116-121      | VSSEL142[1]         |
| 450 | 0x1C2               |          | I2C_4_TEI      | L                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL116-121      | VSSEL142[2]         |
| 451 | 0x1C3               |          | I2C_4_EEI      | L                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL116-121      | VSSEL142[3]         |
| 452 | 0x1C4               | I2C_5    | I2C_5_RXI      | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL116-121      | VSSEL142[4]         |
| 453 | 0x1C5               |          | I2C_5_TXI      | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL116-121      | VSSEL142[5]         |
| 454 | 0x1C6               |          | I2C_5_TEI      | L                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL116-121      | VSSEL142[6]         |
| 455 | 0x1C7               |          | I2C_5_EEI      | L                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL116-121      | VSSEL142[7]         |
| 456 | 0x1C8               | I2C_6    | I2C_6_RXI      | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL116-121      | VSSEL142[8]         |
| 457 | 0x1C9               |          | I2C_6_TXI      | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL116-121      | VSSEL142[9]         |
| 458 | 0x1CA               |          | I2C_6_TEI      | L                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL116-121      | VSSEL142[10]        |
| 459 | 0x1CB               |          | I2C_6_EEI      | L                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL116-121      | VSSEL142[11]        |
| 460 | 0x1CC               | USART_1  | USART_1_WUPI   | Edge                          | ✓                | -              | INTSEL0-31                                                 | INTSEL116-121      | VSSEL142[12]        |
| 461 | 0x1CD               | PVD      | PVD_PVD1       | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL116-121      | VSSEL142[13]        |
| 462 | 0x1CE               |          | PVD_PVD2       | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL116-121      | VSSEL142[14]        |
| 463 | 0x1CF               | OTS      | OTS            | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL116-121      | VSSEL142[15]        |
| 464 | 0x1D0               | FCM      | FCMFERRI       | L                             | ✓                | -              | INTSEL0-31                                                 | INTSEL116-121      | VSSEL142[16]        |
| 465 | 0x1D1               |          | FCMMENDI       | L                             | ✓                | -              | INTSEL0-31                                                 | INTSEL116-121      | VSSEL142[17]        |
| 466 | 0x1D2               |          | FCMCOVFI       | L                             | ✓                | -              | INTSEL0-31                                                 | INTSEL116-121      | VSSEL142[18]        |
| 467 | 0x1D3               | WDT      | WDT_REFUDF     | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL116-121      | VSSEL142[19]        |
| 468 | 0x1D4               | CTC      | CTC_ERR        | P                             | ✓                | -              | INTSEL0-31                                                 | INTSEL116-121      | VSSEL142[20]        |
| 469 | 0x1D5               | ERMU     | ERMU_LPI       | P                             | ✓                | -              | INTSEL0-31                                                 | INTSEL116-121      | VSSEL142[21]        |
| 470 | 0x1D6               | USART_1  | USART_1_TENDI  | L                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL116-121      | VSSEL142[22]        |
| 471 | 0x1D7               | USART_2  | USART_2_TENDI  | L                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL116-121      | VSSEL142[23]        |
| 472 | 0x1D8               | USART_3  | USART_3_TENDI  | L                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL116-121      | VSSEL142[24]        |
| 473 | 0x1D9               | USART_4  | USART_4_TENDI  | L                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL116-121      | VSSEL142[25]        |
| 474 | 0x1DA               | USART_5  | USART_5_TENDI  | L                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL116-121      | VSSEL142[26]        |
| 475 | 0x1DB               | USART_6  | USART_6_TENDI  | L                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL116-121      | VSSEL142[27]        |
| 476 | 0x1DC               | USART_7  | USART_7_TENDI  | L                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL116-121      | VSSEL142[28]        |
| 477 | 0x1DD               | USART_8  | USART_8_TENDI  | L                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL116-121      | VSSEL142[29]        |
| 478 | 0x1DE               | USART_9  | USART_9_TENDI  | L                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL116-121      | VSSEL142[30]        |
| 479 | 0x1DF               | USART_10 | USART_10_TENDI | L                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL116-121      | VSSEL142[31]        |
| 480 | 0x1E0               | ADC_1    | ADC_1_EOCA     | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL122-127      | VSSEL143[0]         |
| 481 | 0x1E1               |          | ADC_1_EOCB     | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL122-127      | VSSEL143[1]         |
| 482 | 0x1E2               |          | ADC_1_CMP0     | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL122-127      | VSSEL143[2]         |
| 483 | 0x1E3               |          | ADC_1_CMP1     | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL122-127      | VSSEL143[3]         |



| SN  | Interrupt Event No. | Modules | Interrupt Name | Interrupt Type <sup>(1)</sup> | Interrupt Source | Trigger Source | NVIC Interrupt Request (IRQ) Number Mapping <sup>(2)</sup> |                    |                     |
|-----|---------------------|---------|----------------|-------------------------------|------------------|----------------|------------------------------------------------------------|--------------------|---------------------|
|     |                     |         |                |                               |                  |                | NVIC Vector 0-31                                           | NVIC Vector 32-127 | NVIC Vector 128-143 |
| 484 | 0x1E4               | ADC_2   | ADC_2_EOCA     | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL122-127      | VSSEL143[4]         |
| 485 | 0x1E5               |         | ADC_2_EOCB     | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL122-127      | VSSEL143[5]         |
| 486 | 0x1E6               |         | ADC_2_CMP0     | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL122-127      | VSSEL143[6]         |
| 487 | 0x1E7               |         | ADC_2_CMP1     | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL122-127      | VSSEL143[7]         |
| 488 | 0x1E8               | ADC_3   | ADC_3_EOCA     | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL122-127      | VSSEL143[8]         |
| 489 | 0x1E9               |         | ADC_3_EOCB     | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL122-127      | VSSEL143[9]         |
| 490 | 0x1EA               |         | ADC_3_CMP0     | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL122-127      | VSSEL143[10]        |
| 491 | 0x1EB               |         | ADC_3_CMP1     | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL122-127      | VSSEL143[11]        |
| 492 | 0x1EC               | TRNG    | TRNG_END       | P                             | ✓                | ✓              | INTSEL0-31                                                 | INTSEL122-127      | VSSEL143[12]        |
| 493 | 0x1ED               | CANRAM  | CAN1RAM_ECCER  | L                             | ✓                | -              | INTSEL0-31                                                 | INTSEL122-127      | VSSEL143[13]        |
| 494 | 0x1EE               |         | CAN2RAM_ECCER  | L                             | ✓                | -              | INTSEL0-31                                                 | INTSEL122-127      | VSSEL143[14]        |
| 495 | 0x1EF               | MCANRAM | MCANRAM_ECCER  | L                             | ✓                | -              | INTSEL0-31                                                 | INTSEL122-127      | VSSEL143[15]        |
| 496 | 0x1F0               | NFC     | NFC_INT        | L                             | ✓                | -              | INTSEL0-31                                                 | INTSEL122-127      | VSSEL143[16]        |
| 497 | 0x1F1               | MCAN_1  | MCAN_1_INT0    | L                             | ✓                | -              | INTSEL0-31                                                 | INTSEL122-127      | VSSEL143[17]        |
| 498 | 0x1F2               |         | MCAN_1_INT1    | L                             | ✓                | -              | INTSEL0-31                                                 | INTSEL122-127      | VSSEL143[18]        |
| 499 | 0x1F3               | MCAN_2  | MCAN_2_INT0    | L                             | ✓                | -              | INTSEL0-31                                                 | INTSEL122-127      | VSSEL143[19]        |
| 500 | 0x1F4               |         | MCAN_2_INT1    | L                             | ✓                | -              | INTSEL0-31                                                 | INTSEL122-127      | VSSEL143[20]        |
| 501 | 0x1F5               | -       | -              | -                             | -                | -              | INTSEL0-31                                                 | INTSEL122-127      | VSSEL143[21]        |
| 502 | 0x1F6               | -       | -              | -                             | -                | -              | INTSEL0-31                                                 | INTSEL122-127      | VSSEL143[22]        |
| 503 | 0x1F7               | -       | -              | -                             | -                | -              | INTSEL0-31                                                 | INTSEL122-127      | VSSEL143[23]        |
| 504 | 0x1F8               | -       | -              | -                             | -                | -              | INTSEL0-31                                                 | INTSEL122-127      | VSSEL143[24]        |
| 505 | 0x1F9               | -       | -              | -                             | -                | -              | INTSEL0-31                                                 | INTSEL122-127      | VSSEL143[25]        |
| 506 | 0x1FA               | -       | -              | -                             | -                | -              | INTSEL0-31                                                 | INTSEL122-127      | VSSEL143[26]        |
| 507 | 0x1FB               | -       | -              | -                             | -                | -              | INTSEL0-31                                                 | INTSEL122-127      | VSSEL143[27]        |
| 508 | 0x1FC               | -       | -              | -                             | -                | -              | INTSEL0-31                                                 | INTSEL122-127      | VSSEL143[28]        |
| 509 | 0x1FD               | -       | -              | -                             | -                | -              | INTSEL0-31                                                 | INTSEL122-127      | VSSEL143[29]        |
| 510 | 0x1FE               | -       | -              | -                             | -                | -              | INTSEL0-31                                                 | INTSEL122-127      | VSSEL143[30]        |
| 511 | 0x1FF               | -       | -              | -                             | -                | -              | INTSEL0-31                                                 | INTSEL122-127      | VSSEL143[31]        |



#### Note

1. "P" means pulse interruption, and "L" means level interruption. The interrupt flag bit must be cleared in the interrupt handler for level interrupts.
2. INTSEL refers to the INTC\_INTSEL register, and VSSEL refers to the INTC\_VSSEL register. If the interrupt event is not configured, the number selected by the interrupt selection register has no effect.
3. AOS\_STRG is generated by software setting the INTSFTTRG.STRG bit register.

## 11.3 Functional Description

### 11.3.1 Function Block Diagram

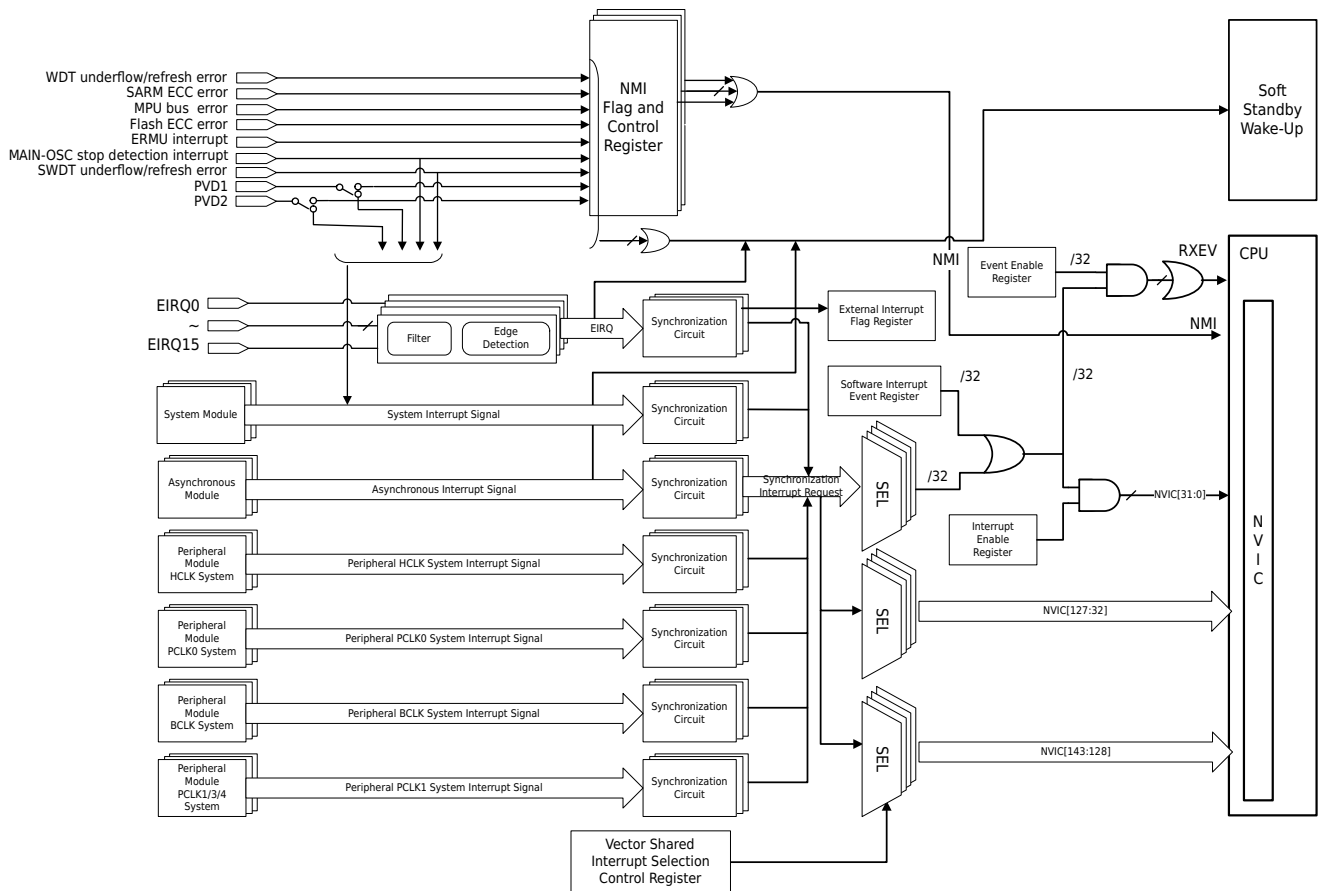


Figure 11-1 Interrupt System Block Diagram

### 11.3.2 Non-maskable Interrupt

The non-maskable interrupt events are as follows:

- ERMU interrupt
- XTAL oscillation fault interrupt
- WDT underflow/refresh interrupt
- SWDT underflow/refresh error interrupt
- Low voltage detection PVD1 interrupt
- Low voltage detection PVD2 interrupt
- Flash ECC check error interrupt
- SRAM ECC parity error interrupt
- MPU bus error interrupt

Non-maskable interrupts have the highest priority and can select multiple interrupt events. The status of each interrupt event can be queried through the interrupt flag register (INTC\_NMIFR). Please make sure all the flag bits are "0" before the non-maskable interrupt processing exits.

When you need to use non-maskable interrupts, please set them according to the following process:

1. Configure the corresponding functions for non-maskable interrupt events.
2. Write 1 to each bit of INTC\_NMIFCR to clear the interrupt flag and prevent malfunction.

3. Set the interrupt enable register INTC\_NMIER to enable the corresponding interrupt event.

**Notice**

Once the corresponding bit of INTC\_NMIER is set to 1, it cannot be changed unless a reset occurs.

### 11.3.3 External Interruption

Equipped with 16 external interrupt events, the interrupt status can be queried through the interrupt flag register (INTC\_EIFR). When using external interrupts, follow the procedure below:

1. Configure the corresponding external pin to the external interrupt function.
2. Write 1 to the respective bits of INTC\_EIFCR to clear the interrupt flags and prevent false triggering.
3. Clear INTC\_EIRQCRx.EFEN (x=0 to 15) bit to disable the digital filter.
4. Configure INTC\_EIRQCRx.EIRQTRG bit to select the trigger edge or level; set the EISMPCLK bit to choose the digital filter sampling clock; then set the EFEN bit to enable the digital filter.

### 11.3.4 Interrupt Event Multiplexer

This interrupt controller uses a total of 144 interrupt vectors, provides 3 kinds of interrupt event request selection methods, and meets various interrupt configuration requirements through flexible combination. For specific interrupt event numbers, refer to the "[Interrupt Event List](#)".

- The First Method

There are 32 interrupt vectors, corresponding to NVIC interrupt vectors 0-31. The interrupt source is selected via interrupt/event selection registers INTC\_INTSEL0 to INTC\_INTSEL31, allowing any one of the available interrupt event requests to be chosen. Interrupt vectors 0-31 must be enabled via the INTC\_IER register.

- The Second Method

There are 96 interrupt vectors in total, corresponding to NVIC interrupt vectors 32-127, which are selected through the interrupt selection registers INTC\_INTSEL32-127. The 96 interrupt vectors are grouped into 16 groups with 6 vectors per group. Meanwhile, all interrupt event requests are also divided into 16 groups with 32 requests per group. Each of the 6 interrupt vectors in a group can select any one of the 32 requests in the corresponding interrupt event request group through the interrupt selection register.

- The Third Method

There are 16 interrupt vectors, corresponding to NVIC interrupt vectors 128-143. Interrupt event requests are enabled through INTC\_VSSEL128-143. Every 32 peripheral interrupt event requests can share one interrupt vector. By selecting the interrupt event requests with peripheral flag bits, each peripheral can request an interrupt. Refer to the [Interrupt Event List](#) NVIC vectors 128-143 column for details.

### 11.3.5 Software Interrupt

The software interrupt function can generate an interrupt request by directly writing to the software interrupt register INTC\_SWIER. A total of 32 software interrupts are available, corresponding to NVIC interrupt request numbers 0-31. Their enablement is controlled by the interrupt enable register INTC\_IER.

### 11.3.6 Event Input

NIVC interrupt requests numbered 0-31 can also be used as event inputs (RXEV) to wake up the core (WFE), with their enablement controlled by the event enable register INTC\_EVTER.

The functional block diagram of interrupt requests and event inputs is shown below:

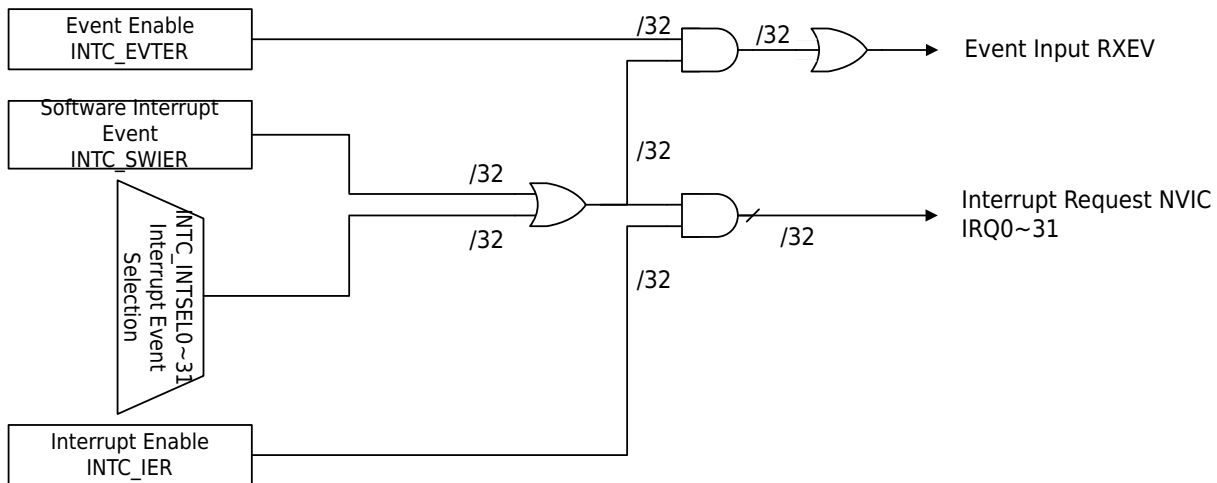


Figure 11-2 Interrupt Request/Event Input Selection

### 11.3.7 WFE Event Management

The interrupt controller can wake up the core (WFE) through interrupt requests and event inputs.

- Interrupt Request Wake-up Core

Select an interrupt event, set INTC\_INTSEL according to the selected interrupt request and enable the INTC\_IER register (without enabling the interrupt request in NVIC). Set the INTC\_EVTER register to disabled, and enable the SEVONPEND bit in the Cortex-M4 core control register SCR. Executing WFE puts the core into Sleep mode. When the selected interrupt event occurs, the core is awakened from Sleep mode but does not enter the interrupt handler.

- Event Input Wake-up Core

Select an interrupt event, set INTC\_INTSEL according to the selected interrupt request and enable the INTC\_EVTER register. The INTC\_IER register is set to disabled. Executing WFE puts the core into Sleep mode. When the selected interrupt event occurs, the core is woken up from Sleep mode.

### 11.3.8 Noise Filter

In order to eliminate input noise, the external interrupt EIRQx (x=0-15) is configured with both digital and analog filters.

The digital filter is enabled via the INTC\_EIRQCRx.EFEN bit. The filter samples the EIRQ input signal using PCLK3 as the sampling clock. The filtering width is jointly determined by the setting value of the EIRQCR.EISMPCLK bit and NFSEL bit. The input signal is filtered out if its level is smaller than the sampling period × (NFSEL setting value + 1).

Example: When NFSEL=0x02, signals with widths less than 3 sampling periods will be filtered out. The sampling period is set by the INTC\_EIRQCRx.EISMPCLK bit.

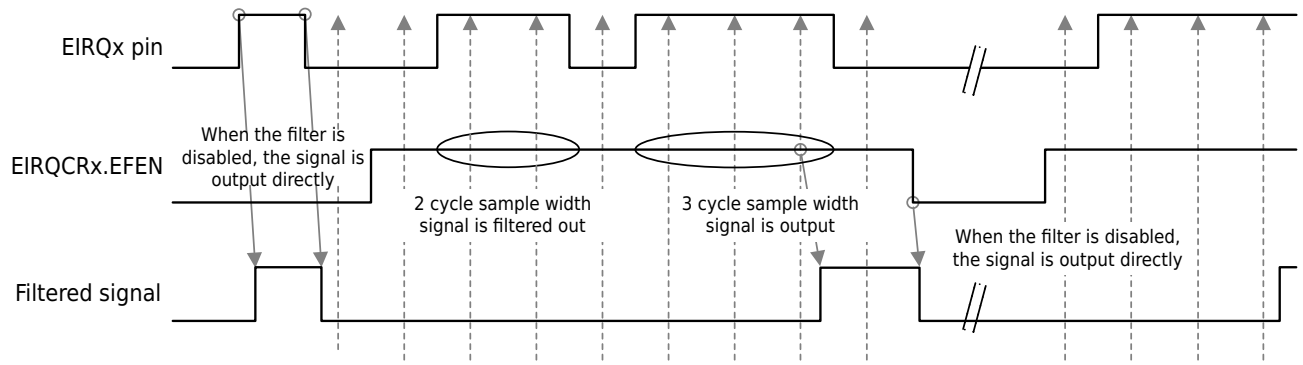


Figure 11-3 Digital Filter Operation Diagram (taking NFSEL=0x02 as an example)

The analog filter is enabled by the INTC\_EIRQCRx.NOCEN register. After enabling, the signal is filtered with the filter width selected by the INTC\_NOCCR.NOCSEL bit.

The two filters are independent of each other and can be enabled or disabled separately. When both are enabled, the signal first passes through the analog filter and then the digital filter before entering the internal circuit.

The digital filter must be turned off before entering the Stop mode. The digital filter is enabled after returning from the Stop mode. The analog filter can work in Stop mode.

### 11.3.9 Low Power Mode Return

#### Sleep Mode Return

To wake up the CPU from Sleep mode using an interrupt request, configure the following:

- Select an interrupt event as the interrupt request.  
If using interrupt requests 0-31, configure the INTC\_IER register.
- Enable the corresponding interrupt in the NVIC.
- If using a non-maskable interrupt, configure the INTC\_NMIER register.

#### Stop Mode Return

To return from Stop mode via an interrupt request, configure the following:

- Select an interrupt event as the interrupt request.  
If using interrupt requests 0-31, configure the INTC\_IER register.
- Enable the interrupt source via the INTC\_WKEN register.  
If non-maskable interrupts are used, enable the interrupt events through the INTC\_NMIER register.  
The non-maskable interrupts that can wake up the stop mode are: XTAL oscillation fault interrupt, SWDT underflow/refresh interrupt.
- Enable the corresponding interrupt request in the NVIC.

#### Power-down Mode Return

The return from Power-down mode can be returned by the conditions described in the Power Control (PWC) section, NRST pin reset, power-on reset, and brown-out reset conditions. After returning, the CPU enters the reset interrupt processing. For detailed instructions, please refer to the relevant description in the "Power Control (PWC)" section.

#### Non-maskable Interrupt and WFI Command

Before executing the WFI instruction, please confirm that all status bits of the non-maskable interrupt flag register INTC\_NMIFR are 0.

## 11.4 Register Description

### 11.4.1 Register List

Base Address: 0x40051000

Table 11-4 Interrupt Controller (INTC) Register Offset

| Offset             | Register     | Description                                              | Width |
|--------------------|--------------|----------------------------------------------------------|-------|
| 0x0000             | INTC_NOCCR   | External Interrupt Filter Control Register               | 32    |
| 0x0004             | INTC_NMIER   | Non-maskable Interrupt Enable Register                   | 32    |
| 0x0008             | INTC_NMIFR   | Non-maskable Interrupt Flag Register                     | 32    |
| 0x000C             | INTC_NMIFCR  | Non-maskable Interrupt Flag Clear Register               | 32    |
| 0x0010+0x4*x       | INTC_EIRQCRx | External Interrupt Control Register x<br>(x=0-15)        | 32    |
| 0x0054             | INTC_EIFR    | External Interrupt Flag Register                         | 32    |
| 0x0058             | INTC_EIFCR   | External Interrupt Flag Clear Register                   | 32    |
| 0x005C+0x4*x       | INTC_INTSELx | Interrupt Event Select Register x (x=0-31)               | 32    |
| 0x00DC+0x4*(y-32)  | INTC_INTSELy | Interrupt Select Register y (y=32-127)                   | 32    |
| 0x025C+0x4*(z-128) | INTC_VSSELz  | Vector Shared Interrupt Select Register z<br>(z=128-143) | 32    |
| 0x0050             | INTC_WKEN    | Stop Mode Wake-up Event Enable Register                  | 32    |
| 0x029C             | INTC_SWIER   | Software Interrupt Event Register                        | 32    |
| 0x02A0             | INTC_EVTER   | Event Enable Register                                    | 32    |
| 0x02A4             | INTC_IER     | Interrupt Enable Register                                | 32    |
| 0x02A8             | INTC_FPUIER  | FPU Interrupt Enable Register                            | 32    |

## 11.4.2 External Interrupt Filter Control Register (INTC\_NOCCR)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |        |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|--------|----|----|---|---|---|---|---|---|---|---|---|---|
| 0x0000 | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12     | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |        |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | NOCSEL |    |    |   |   |   |   |   |   |   |   |   |   |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RW     |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit   | Name     | Function                                | Description                                                                                                                                                                                                                                                                                                                                                                   | Access |
|-------|----------|-----------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:14 | Reserved | Reserved                                | -                                                                                                                                                                                                                                                                                                                                                                             | -      |
| 13:12 | NOCSEL   | Analog Filter Filtering Width Selection | <ul style="list-style-type: none"> <li>0b00: Filtering width level 1</li> <li>0b01: Filtering width level 2</li> <li>0b10: Filtering width level 3</li> <li>0b11: Filtering width level 4</li> </ul> For the specific values of each gear, please refer to the relevant description in the "Electrical Characteristics-EIRQ Filter Characteristics" section in the Datasheet. | RW     |
| 11:0  | Reserved | Reserved                                | -                                                                                                                                                                                                                                                                                                                                                                             | -      |

## 11.4.3 Non-maskable Interrupt Enable Register (INTC\_NMIER)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |        |           |       |          |           |          |   |   |           |          |        |        |        |          |  |  |  |  |  |  |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|--------|-----------|-------|----------|-----------|----------|---|---|-----------|----------|--------|--------|--------|----------|--|--|--|--|--|--|
| 0x0004 | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13     | 12        | 11    | 10       | 9         | 8        | 7 | 6 | 5         | 4        | 3      | 2      | 1      | 0        |  |  |  |  |  |  |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |        |           |       |          |           |          |   |   |           |          |        |        |        |          |  |  |  |  |  |  |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | ERMUEN | FECCERREN | WDTEN | BUSERREN | RECCERREN | Reserved |   |   | XATLSTPEN | Reserved | PVD2EN | PVD1EN | SWDTEN | Reserved |  |  |  |  |  |  |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RW     | RW        | RW    | RW       | RW        |          |   |   |           | RW       |        |        | RW     |          |  |  |  |  |  |  |

| Bit   | Name      | Function                                       | Description                                                                                                                                                                     | Access |
|-------|-----------|------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:14 | Reserved  | Reserved                                       | -                                                                                                                                                                               | -      |
| 13    | ERMUEN    | ERUM interrupt selection                       | <ul style="list-style-type: none"> <li>0b0: Disable interrupt as non-maskable interrupt source</li> <li>0b1: Select the interrupt as a non-maskable interrupt source</li> </ul> | RW     |
| 12    | FECCERREN | Flash ECC check error interrupt selection      | <ul style="list-style-type: none"> <li>0b0: Disable interrupt as non-maskable interrupt source</li> <li>0b1: Select the interrupt as a non-maskable interrupt source</li> </ul> | RW     |
| 11    | WDTEN     | WDT under-flow/refresh interrupt selection     | <ul style="list-style-type: none"> <li>0b0: Disable interrupt as non-maskable interrupt source</li> <li>0b1: Select the interrupt as a non-maskable interrupt source</li> </ul> | RW     |
| 10    | BUSERREN  | MPU main bus error interrupt selection         | <ul style="list-style-type: none"> <li>0b0: Disable interrupt as non-maskable interrupt source</li> <li>0b1: Select the interrupt as a non-maskable interrupt source</li> </ul> | RW     |
| 9     | RECCERREN | SRAM ECC check error interrupt selection       | <ul style="list-style-type: none"> <li>0b0: Disable interrupt as non-maskable interrupt source</li> <li>0b1: Select the interrupt as a non-maskable interrupt source</li> </ul> | RW     |
| 8:6   | Reserved  | Reserved                                       | -                                                                                                                                                                               | -      |
| 5     | XATLSTPEN | XTAL oscillation failure interrupt selection   | <ul style="list-style-type: none"> <li>0b0: Disable interrupt as non-maskable interrupt source</li> <li>0b1: Select the interrupt as a non-maskable interrupt source</li> </ul> | RW     |
| 4     | Reserved  | Reserved                                       | -                                                                                                                                                                               | -      |
| 3     | PVD2EN    | Low voltage detection PVD2 interrupt selection | <ul style="list-style-type: none"> <li>0b0: Disable interrupt as non-maskable interrupt source</li> <li>0b1: Select the interrupt as a non-maskable interrupt source</li> </ul> | RW     |



| Bit | Name            | Function                                          | Description                                                                                                                                                                         | Access |
|-----|-----------------|---------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 2   | PVD1EN          | Low voltage detection PVD1 interrupt selection    | <ul style="list-style-type: none"> <li>● 0b0: Disable interrupt as non-maskable interrupt source</li> <li>● 0b1: Select the interrupt as a non-maskable interrupt source</li> </ul> | RW     |
| 1   | SWDTEN          | SWDT under-flow/refresh error interrupt selection | <ul style="list-style-type: none"> <li>● 0b0: Disable interrupt as non-maskable interrupt source</li> <li>● 0b1: Select the interrupt as a non-maskable interrupt source</li> </ul> | RW     |
| 0   | <i>Reserved</i> | <i>Reserved</i>                                   | -                                                                                                                                                                                   | -      |

## 11.4.4 Non-maskable Interrupt Flag Register (INTC\_NMIFR)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |       |         |      |         |          |          |   |   |          |          |       |       |       |          |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-------|---------|------|---------|----------|----------|---|---|----------|----------|-------|-------|-------|----------|
| 0x0008 | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13    | 12      | 11   | 10      | 9        | 8        | 7 | 6 | 5        | 4        | 3     | 2     | 1     | 0        |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |       |         |      |         |          |          |   |   |          |          |       |       |       |          |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | ERMUF | FECERRF | WDTF | BUSERRF | RECCERRF | Reserved |   |   | XTALSTPF | Reserved | PVD2F | PVD1F | SWDTF | Reserved |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RO    | RO      | RO   | RO      | RO       |          |   |   | RO       |          | RO    | RO    | RO    |          |

| Bit   | Name     | Function                                  | Description                                                                                                                                                                                | Access |
|-------|----------|-------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:14 | Reserved | Reserved                                  | -                                                                                                                                                                                          | -      |
| 13    | ERMUF    | ERMU interrupt flag                       | <ul style="list-style-type: none"> <li>0b0: No ERMU interrupt request occurred</li> <li>0b1: ERMU interrupt request occurred</li> </ul>                                                    | RO     |
| 12    | FECERRF  | Flash ECC check error interrupt flag      | <ul style="list-style-type: none"> <li>0b0: No Flash ECC check error request occurred</li> <li>0b1: Flash ECC check error request occurred</li> </ul>                                      | RO     |
| 11    | WDTF     | WDT underflow/refresh interrupt flag      | <ul style="list-style-type: none"> <li>0b0: No WDT underflow/refresh request occurred</li> <li>0b1: WDT underflow/refresh request occurred</li> </ul>                                      | RO     |
| 10    | BUSERRF  | MPU main bus error interrupt flag         | <ul style="list-style-type: none"> <li>0b0: No MPU master bus error request occurred</li> <li>0b1: MPU master bus error request occurred</li> </ul>                                        | RO     |
| 9     | RECCERRF | SRAM ECC check error interrupt flag       | <ul style="list-style-type: none"> <li>0b0: No SRAM ECC check error request occurred</li> <li>0b1: SRAM ECC check error request occurred</li> </ul>                                        | RO     |
| 8:6   | Reserved | Reserved                                  | -                                                                                                                                                                                          | -      |
| 5     | XTALSTPF | XTAL Oscillation fault interrupt flag     | <ul style="list-style-type: none"> <li>0b0: No XTAL oscillation failure interrupt request has occurred</li> <li>0b1: An XTAL oscillation failure interrupt request has occurred</li> </ul> | RO     |
| 4     | Reserved | Reserved                                  | -                                                                                                                                                                                          | -      |
| 3     | PVD2F    | Low voltage detection PVD2 interrupt flag | <ul style="list-style-type: none"> <li>0b0: No low voltage detection PVD2 request occurred</li> <li>0b1: Low voltage detection PVD2 request occurred</li> </ul>                            | RO     |
| 2     | PVD1F    | Low voltage detection PVD1 interrupt flag | <ul style="list-style-type: none"> <li>0b0: No low voltage detection PVD1 request occurred</li> <li>0b1: Low voltage detection PVD1 request occurred</li> </ul>                            | RO     |

| Bit | Name            | Function                                    | Description                                                                                                                                              | Access |
|-----|-----------------|---------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 1   | SWDTF           | SWDT underflow/refresh error interrupt flag | <ul style="list-style-type: none"><li>● 0b0: No SWDT underflow/refresh request occurred</li><li>● 0b1: SWDT underflow/refresh request occurred</li></ul> | RO     |
| 0   | <i>Reserved</i> | <i>Reserved</i>                             | -                                                                                                                                                        | -      |

## 11.4.5 Non-maskable Interrupt Flag Clear Register (INTC\_NMIFCR)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |          |            |         |            |             |          |   |   |   |             |          |   |          |          |          |          |  |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----------|------------|---------|------------|-------------|----------|---|---|---|-------------|----------|---|----------|----------|----------|----------|--|
| 0x000C | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13       | 12         | 11      | 10         | 9           | 8        | 7 | 6 | 5 | 4           | 3        | 2 | 1        | 0        |          |          |  |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |          |            |         |            |             |          |   |   |   |             |          |   |          |          |          |          |  |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | ERMUFCLR | FECERRFCLR | WDTFCLR | BUSERRFCLR | RECCERRFCLR | Reserved |   |   |   | XATLSTPFCLR | Reserved |   | PVD2FCLR | PVD1FCLR | SWDTFCLR | Reserved |  |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | WO       | WO         | WO      | WO         | WO          |          |   |   |   | WO          |          |   | WO       | WO       |          |          |  |

| Bit   | Name        | Function                                        | Description                                                              | Access |
|-------|-------------|-------------------------------------------------|--------------------------------------------------------------------------|--------|
| 31:14 | Reserved    | Reserved                                        | -                                                                        | -      |
| 13    | ERMUFCLR    | ERMU interrupt flag cleared                     | Write 1 to clear the flag, write 0 has no effect; read always returns 0. | WO     |
| 12    | FECERRFCLR  | Flash ECC check error interrupt flag clear      | Write 1 to clear the flag, write 0 has no effect; read always returns 0. | WO     |
| 11    | WDTFCLR     | WDT under-flow/refresh interrupt flag clear     | Write 1 to clear the flag, write 0 has no effect; read always returns 0. | WO     |
| 10    | BUSERRFCLR  | MPU main bus error interrupt flag cleared       | Write 1 to clear the flag, write 0 has no effect; read always returns 0. | WO     |
| 9     | RECCERRFCLR | SRAM ECC parity error interrupt flag clear      | Write 1 to clear the flag, write 0 has no effect; read always returns 0. | WO     |
| 8:6   | Reserved    | Reserved                                        | -                                                                        | -      |
| 5     | XATLSTPFCLR | XTAL oscillation failure interrupt flag clear   | Write 1 to clear the flag, write 0 has no effect; read always returns 0. | WO     |
| 4     | Reserved    | Reserved                                        | -                                                                        | -      |
| 3     | PVD2FCLR    | Low voltage detection PVD2 interrupt flag clear | Write 1 to clear the flag, write 0 has no effect; read always returns 0. | WO     |
| 2     | PVD1FCLR    | Low voltage detection PVD1 interrupt flag clear | Write 1 to clear the flag, write 0 has no effect; read always returns 0. | WO     |

| Bit | Name            | Function                                             | Description                                                              | Access |
|-----|-----------------|------------------------------------------------------|--------------------------------------------------------------------------|--------|
| 1   | SWDTFCLR        | SWDT under-flow/refresh error interrupt flag cleared | Write 1 to clear the flag, write 0 has no effect; read always returns 0. | WO     |
| 0   | <i>Reserved</i> | <i>Reserved</i>                                      | -                                                                        | -      |

### 11.4.6 External Interrupt Control Register (INTC\_EIRQCRx) (x=0-15)

Offset: 0x0010+0x4\*x

| Offset | Bit Position |    |    |    |    |    |    |    |       |    |    |    |    |    |    |    |       |          |    |    |    |    |   |   |   |      |          |          |   |          |         |   |    |  |
|--------|--------------|----|----|----|----|----|----|----|-------|----|----|----|----|----|----|----|-------|----------|----|----|----|----|---|---|---|------|----------|----------|---|----------|---------|---|----|--|
| -      | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23    | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15    | 14       | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6    | 5        | 4        | 3 | 2        | 1       | 0 |    |  |
| Reset  | 0x00020000   |    |    |    |    |    |    |    |       |    |    |    |    |    |    |    |       |          |    |    |    |    |   |   |   |      |          |          |   |          |         |   |    |  |
| Name   | Reserved     |    |    |    |    |    |    |    | NFSEL |    |    |    |    |    |    |    | NOCEN | Reserved |    |    |    |    |   |   |   | EFEN | Reserved | EISMPCLK |   | Reserved | EIRQTRG |   |    |  |
| Access |              |    |    |    |    |    |    |    | RW    |    |    |    |    |    |    |    | RW    |          |    |    |    |    |   |   |   | RW   |          | RW       |   |          |         |   | RW |  |

| Bit   | Name     | Function                                 | Description                                                                                                                                                                            | Access |
|-------|----------|------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:24 | Reserved | Reserved                                 | -                                                                                                                                                                                      | -      |
| 23:16 | NFSEL    | EIRQ digital filter width selection      | The setting range is 0x00-0xFF<br>Filter width=sampling clock period×(NFS setting value + 1)                                                                                           | RW     |
| 15    | NOCEN    | EIRQ Analog filter enabled               | <ul style="list-style-type: none"> <li>0b0: Disable analog filter function</li> <li>0b1: Enable analog filter function</li> </ul> The filter width is determined by INTC_NOCCR.NOCSEL. | RW     |
| 14:8  | Reserved | Reserved                                 | -                                                                                                                                                                                      | -      |
| 7     | EFEN     | EIRQ digital filter enabled              | <ul style="list-style-type: none"> <li>0b0: Disable digital filter function</li> <li>0b1: Enable digital filter function</li> </ul>                                                    | RW     |
| 6     | Reserved | Reserved                                 | -                                                                                                                                                                                      | -      |
| 5:4   | EISMPCLK | Digital filter sampling period selection | <ul style="list-style-type: none"> <li>0b00: 1 × PCLK3 period</li> <li>0b01: 8 × PCLK3 period</li> <li>0b10: 32 × PCLK3 period</li> <li>0b11: 64 × PCLK3 period</li> </ul>             | RW     |
| 3:2   | Reserved | Reserved                                 | -                                                                                                                                                                                      | -      |
| 1:0   | EIRQTRG  | Trigger selection                        | <ul style="list-style-type: none"> <li>0b00: Falling edge</li> <li>0b01: Rising edge</li> <li>0b10: Double edge</li> <li>0b11: Low level</li> </ul>                                    | RW     |

## 11.4.7 External Interrupt Flag Register (INTC\_EIFR)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |               |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---------------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0x0054 | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15            | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |               |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | EIF<br><br>RO |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |               |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit   | Name     | Function      | Description                                                                                                                                                                                           | Access |
|-------|----------|---------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:16 | Reserved | Reserved      | -                                                                                                                                                                                                     | -      |
| 15:0  | EIF      | EIRQ Flag Bit | Corresponding to EIF15-EIF0:<br><ul style="list-style-type: none"> <li>0b0: EIRQ event did not occur, or the EIFCR bit was cleared by writing</li> <li>0b1: The selected EIRQ event occurs</li> </ul> | RO     |

## 11.4.8 External Interrupt Flag Clear Register (INTC{EIFCR)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |        |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|--------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0x0058 | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15     | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |        |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | EIFCLR |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | WO     |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit   | Name     | Function            | Description                                                                                                                                   | Access |
|-------|----------|---------------------|-----------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:16 | Reserved | Reserved            | -                                                                                                                                             | -      |
| 15:0  | EIFCLR   | EIRQ Flag Clear Bit | Corresponding to EIFCLR15-EIFCLR0, write 1 to clear the corresponding bit of the EIFR register, write 0 has no effect; read always returns 0. | WO     |

11.4.9 Interrupt Event Select Register (INTC\_INTSELx) (x=0-31)

Offset: 0x005C+0x4\*x

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |        |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|--------|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| '      | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19     | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x000001FF   |    |    |    |    |    |    |    |    |    |    |    |        |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    | INTSEL |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access | RW           |    |    |    |    |    |    |    |    |    |    |    | RW     |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit  | Name     | Function                          | Description                                                                                                                | Access |
|------|----------|-----------------------------------|----------------------------------------------------------------------------------------------------------------------------|--------|
| 31:9 | Reserved | Reserved                          | -                                                                                                                          | -      |
| 8:0  | INTSEL   | Interrupt Event Request Selection | Select the interrupt event request corresponding to 0x000-0x1FE, see the <a href="#">Interrupt Event Table</a> for details | RW     |



### 11.4.10 Interrupt Select Register (INTC\_INTSELy) (y=32-127)

Offset: 0x00DC+0x4\*(y-32)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |        |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|--------|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| -      | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18     | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x000001FF   |    |    |    |    |    |    |    |    |    |    |    |    |        |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    | INTSEL |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    | RW     |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit  | Name     | Function                          | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | Access |
|------|----------|-----------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:9 | Reserved | Reserved                          | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | -      |
| 8:0  | INTSEL   | Interrupt Event Request Selection | <p>Select the events corresponding to the serial numbers in the <a href="#">Interrupt Event Table</a> respectively. The specific corresponding relationships are as follows:</p> <ul style="list-style-type: none"> <li>● INTC_INTSEL32-37: Select the interrupt event request corresponding to 0x000-0x01F, other selections are inactive.</li> <li>● INTC_INTSEL38-43: Select the interrupt event request corresponding to 0x020-0x03F, other selections are inactive.</li> <li>● INTC_INTSEL44-49: Select the interrupt event request corresponding to 0x040-0x05F, other selections are inactive.</li> <li>● INTC_INTSEL50-55: Select the interrupt event request corresponding to 0x060-0x07F, other selections are inactive.</li> <li>● INTC_INTSEL56-61: Select the interrupt event request corresponding to 0x080-0x09F, other selections are inactive.</li> <li>● INTC_INTSEL62-67: Select the interrupt event request corresponding to 0x0A0-0x0BF, other selections are inactive.</li> <li>● INTC_INTSEL68-73: Select the interrupt event request corresponding to 0x0C0-0x0DF, other selections are inactive.</li> <li>● INTC_INTSEL74-79: Select the interrupt event request corresponding to 0x0E0-0x0FF, other selections are inactive.</li> <li>● INTC_INTSEL80-85: Select the interrupt event request corresponding to 0x100-0x11F, other selections are inactive.</li> <li>● INTC_INTSEL86-91: Select the interrupt event request corresponding to 0x120-0x13F, other selections are inactive.</li> <li>● INTC_INTSEL92-97: Select the interrupt event request corresponding to 0x140-0x15F, other selections are inactive.</li> </ul> | RW     |

| Bit | Name   | Function                          | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | Access |
|-----|--------|-----------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 8:0 | INTSEL | Interrupt Event Request Selection | <ul style="list-style-type: none"> <li>● INTC_INTSEL98-103: Select the interrupt event request corresponding to 0x160-0x17F, other selections are inactive.</li> <li>● INTC_INTSEL104-109: Select the interrupt event request corresponding to 0x180-0x19F, other selections are inactive.</li> <li>● INTC_INTSEL110-115: Select the interrupt event request corresponding to 0x1A0-0x1BF, other selections are inactive.</li> <li>● INTC_INTSEL116-121: Select the interrupt event request corresponding to 0x1C0-0x1DF, other selections are inactive.</li> <li>● INTC_INTSEL122-127: Select the interrupt event request corresponding to 0x1E0-0x1FF, other selections are inactive.</li> </ul> | RW     |

### 11.4.11 Vector Shared Interrupt Select Register (INTC\_VSSELz) (z=128-143)

Offset: 0x025C+0x4\*(z-128)

| Offset | Bit Position |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |       |       |       |       |       |       |       |       |       |       |    |
|--------|--------------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|----|
| -      | 31           | 30     | 29     | 28     | 27     | 26     | 25     | 24     | 23     | 22     | 21     | 20     | 19     | 18     | 17     | 16     | 15     | 14     | 13     | 12     | 11     | 10     | 9     | 8     | 7     | 6     | 5     | 4     | 3     | 2     | 1     | 0     |    |
| Reset  | 0x00000000   |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |       |       |       |       |       |       |       |       |       |       |    |
| Name   | VSEL31       | VSEL30 | VSEL29 | VSEL28 | VSEL27 | VSEL26 | VSEL25 | VSEL24 | VSEL23 | VSEL22 | VSEL21 | VSEL20 | VSEL19 | VSEL18 | VSEL17 | VSEL16 | VSEL15 | VSEL14 | VSEL13 | VSEL12 | VSEL11 | VSEL10 | VSEL9 | VSEL8 | VSEL7 | VSEL6 | VSEL5 | VSEL4 | VSEL3 | VSEL2 | VSEL1 | VSEL0 |    |
| Access | RW           | RW     | RW     | RW     | RW     | RW     | RW     | RW     | RW     | RW     | RW     | RW     | RW     | RW     | RW     | RW     | RW     | RW     | RW     | RW     | RW     | RW     | RW    | RW    | RW    | RW    | RW    | RW    | RW    | RW    | RW    | RW    | RW |

| Bit  | Name              | Function          | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | Access |
|------|-------------------|-------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:0 | VSELx<br>(x=31-0) | Interrupt Enabled | <p>Select the events corresponding to the serial numbers in the <a href="#">Interrupt Event Table</a> respectively. The specific corresponding relationships are as follows:</p> <ul style="list-style-type: none"> <li>● INTC_VSSEL128: Each bit of VSELx enables the interrupt event request corresponding to 0x000-0x01F.</li> <li>● INTC_VSSEL129: Each bit of VSELx enables the interrupt event request corresponding to 0x020-0x03F.</li> <li>● INTC_VSSEL130: Each bit of VSELx enables the interrupt event request corresponding to 0x040-0x05F.</li> <li>● INTC_VSSEL131: Each bit of VSELx enables the interrupt event request corresponding to 0x060-0x07F.</li> <li>● INTC_VSSEL132: Each bit of VSELx enables the interrupt event request corresponding to 0x080-0x09F.</li> <li>● INTC_VSSEL133: Each bit of VSELx enables the interrupt event request corresponding to 0x0A0-0x0BF.</li> <li>● INTC_VSSEL134: Each bit of VSELx enables the interrupt event request corresponding to 0x0C0-0x0DF.</li> <li>● INTC_VSSEL135: Each bit of VSELx enables the interrupt event request corresponding to 0x0E0-0x0FF.</li> <li>● INTC_VSSEL136: Each bit of VSELx enables the interrupt event request corresponding to 0x100-0x11F.</li> <li>● INTC_VSSEL137: Each bit of VSELx enables the interrupt event request corresponding to 0x120-0x13F.</li> <li>● INTC_VSSEL138: Each bit of VSELx enables the interrupt event request corresponding to 0x140-0x15F.</li> <li>● INTC_VSSEL139: Each bit of VSELx enables the interrupt event request corresponding to 0x160-0x17F.</li> <li>● INTC_VSSEL140: Each bit of VSELx enables the interrupt event request corresponding to 0x180-0x19F.</li> <li>● INTC_VSSEL141: Each bit of VSELx enables the interrupt event request corresponding to 0x1A0-0x1BF.</li> <li>● INTC_VSSEL142: Each bit of VSELx enables the interrupt event request corresponding to 0x1C0-0x1DF.</li> <li>● INTC_VSSEL143: Each bit of VSELx enables the interrupt event request corresponding to 0x1E0-0x1FF.</li> </ul> | RW     |

## 11.4.12 Stop Mode Wake-up Event Enable Register (INTC\_WKEN)

| Offset | Bit Position |         |         |         |        |             |              |             |            |            |          |          |          |          |          |          |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|--------|--------------|---------|---------|---------|--------|-------------|--------------|-------------|------------|------------|----------|----------|----------|----------|----------|----------|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
| 0x0050 | 31           | 30      | 29      | 28      | 27     | 26          | 25           | 24          | 23         | 22         | 21       | 20       | 19       | 18       | 17       | 16       | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Reset  | 0x00000000   |         |         |         |        |             |              |             |            |            |          |          |          |          |          |          |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Name   | Reserved     | ETHWKEN | USFWKEN | USHWKEN | RXWKEN | TMR2OVFWKEN | TMR2GCMPWKEN | TMR0CMPWKEN | RTCPRDWKEN | RTCALMWKEN | WKTMWKEN | CMP1WKEN | PVD2WKEN | PVD1WKEN | SWDTWKEN | EIRQWKEN |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Access |              | RW      | RW      | RW      | RW     | RW          | RW           | RW          | RW         | RW         | RW       | RW       | RW       | RW       | RW       |          |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

| Bit   | Name         | Function                                  | Description                                                                                           | Access |
|-------|--------------|-------------------------------------------|-------------------------------------------------------------------------------------------------------|--------|
| 31:30 | Reserved     | Reserved                                  | -                                                                                                     | -      |
| 29    | ETHWKEN      | ETH_PMT Stop mode wake-up enabled         | <ul style="list-style-type: none"> <li>0b0: Wake-up disabled</li> <li>0b1: Wake-up enabled</li> </ul> | RW     |
| 28    | USFWKEN      | USBFS_WKUP Stop mode wakeup enabled       | <ul style="list-style-type: none"> <li>0b0: Wake-up disabled</li> <li>0b1: Wake-up enabled</li> </ul> | RW     |
| 27    | USHWKEN      | USBHS_WKUP Stop mode wakeup enabled       | <ul style="list-style-type: none"> <li>0b0: Wake-up disabled</li> <li>0b1: Wake-up enabled</li> </ul> | RW     |
| 26    | RXWKEN       | USART_1 RX line Stop mode wake-up enabled | <ul style="list-style-type: none"> <li>0b0: Wake-up disabled</li> <li>0b1: Wake-up enabled</li> </ul> | RW     |
| 25    | TMR2OVFWKEN  | TMR21_OVFA Stop mode wake-up enabled      | <ul style="list-style-type: none"> <li>0b0: Wake-up disabled</li> <li>0b1: Wake-up enabled</li> </ul> | RW     |
| 24    | TMR2GCMPWKEN | TMR21_GCMA Stop mode wake-up enabled      | <ul style="list-style-type: none"> <li>0b0: Wake-up disabled</li> <li>0b1: Wake-up enabled</li> </ul> | RW     |
| 23    | TMR0CMPWKEN  | TMR0_1_CMPA Stop mode wake-up enabled     | <ul style="list-style-type: none"> <li>0b0: Wake-up disabled</li> <li>0b1: Wake-up enabled</li> </ul> | RW     |
| 22    | RTCPRDWKEN   | RTC_PRD Stop mode wakeup enabled          | <ul style="list-style-type: none"> <li>0b0: Wake-up disabled</li> <li>0b1: Wake-up enabled</li> </ul> | RW     |

| Bit  | Name       | Function                              | Description                                                                                                                                      | Access |
|------|------------|---------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 21   | RTCALMWKEN | RTC_ALM Stop mode wake-up enabled     | <ul style="list-style-type: none"> <li>● 0b0: Wake-up disabled</li> <li>● 0b1: Wake-up enabled</li> </ul>                                        | RW     |
| 20   | WKTMWKEN   | WKT_MWKEN Stop mode wake-up enabled   | <ul style="list-style-type: none"> <li>● 0b0: Wake-up disabled</li> <li>● 0b1: Wake-up enabled</li> </ul>                                        | RW     |
| 19   | CMP1WKEN   | CMP1 Stop mode wake-up enabled        | <ul style="list-style-type: none"> <li>● 0b0: Wake-up disabled</li> <li>● 0b1: Wake-up enabled</li> </ul>                                        | RW     |
| 18   | PVD2WKEN   | PVD2 Stop mode wake-up enable         | <ul style="list-style-type: none"> <li>● 0b0: Wake-up disabled</li> <li>● 0b1: Wake-up enabled</li> </ul>                                        | RW     |
| 17   | PVD1WKEN   | PVD1 Stop mode wake-up enable         | <ul style="list-style-type: none"> <li>● 0b0: Wake-up disabled</li> <li>● 0b1: Wake-up enabled</li> </ul>                                        | RW     |
| 16   | SWDTWKEN   | SWDT_REFUDF Stop mode wake-up enabled | <ul style="list-style-type: none"> <li>● 0b0: Wake-up disabled</li> <li>● 0b1: Wake-up enabled</li> </ul>                                        | RW     |
| 15:0 | EIRQWKEN   | EIRQ Stop mode wakeup enabled         | Corresponding to EIRQWKEN15-EIRQWKEN0: <ul style="list-style-type: none"> <li>● 0b0: Wake-up disabled</li> <li>● 0b1: Wake-up enabled</li> </ul> | RW     |

### 11.4.13 Software Interrupt Event Register (INTC\_SWIER)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0x029C | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | SWIE         |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access | RW           |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit  | Name | Function                 | Description                                                                                                                                                                   | Access |
|------|------|--------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:0 | SWIE | Software Interrupt Event | Corresponding to SWIE31-SWIE0:<br><ul style="list-style-type: none"> <li>0b0: No software interrupt/event occurred</li> <li>0b1: Software interrupt/event occurred</li> </ul> | RW     |

### 11.4.14 Event Enable Register (INTC\_EVTER)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0x02A0 | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | EVTE         |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access | RW           |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit  | Name | Function      | Description                                                                                                                                             | Access |
|------|------|---------------|---------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:0 | EVTE | Event Enabled | Corresponding to EVTE31-EVTE0:<br><ul style="list-style-type: none"> <li>0b0: Event selection disabled</li> <li>0b1: Event selection enabled</li> </ul> | RW     |

## 11.4.15 Interrupt Enable Register (INTC\_IER)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0x02A4 | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0xFFFFFFFF   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | IEN          |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access | RW           |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit  | Name | Function         | Description                                                                                                                                                                                                                                                                                 | Access |
|------|------|------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:0 | IEN  | Interrupt Enable | IEN31-IEN0 correspond to NVIC interrupt requests 31-0 respectively: <ul style="list-style-type: none"> <li>● 0b0: Interrupt disabled</li> <li>● 0b1: Interrupt enabled</li> </ul> When disabled, the interrupt events corresponding to each interrupt request will not be received by NVIC. | RW     |

## 11.4.16 FPU Interrupt Enable Register (INTC\_FPUIER)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |        |        |        |        |        |        |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|--------|--------|--------|--------|--------|--------|
| 0x02A8 | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5      | 4      | 3      | 2      | 1      | 0      |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |        |        |        |        |        |        |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   | IDCIEN | IXCIEN | UFCIEN | OFCIEN | DZCIEN | IOCIEN |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   | RW     | RW     | RW     | RW     | RW     | RW     |

| Bit  | Name     | Function                      | Description                                                                                                   | Access |
|------|----------|-------------------------------|---------------------------------------------------------------------------------------------------------------|--------|
| 31:6 | Reserved | Reserved                      | -                                                                                                             | -      |
| 5    | IDCIEN   | FPU IDC Interrupt Enable Bit  | <ul style="list-style-type: none"> <li>● 0b0: Interrupt disabled</li> <li>● 0b1: Interrupt enabled</li> </ul> | RW     |
| 4    | IXCIEN   | FPU IXCI Interrupt Enable Bit | <ul style="list-style-type: none"> <li>● 0b0: Interrupt disabled</li> <li>● 0b1: Interrupt enabled</li> </ul> | RW     |
| 3    | UFCIEN   | FPU UFCI Interrupt Enable Bit | <ul style="list-style-type: none"> <li>● 0b0: Interrupt disabled</li> <li>● 0b1: Interrupt enabled</li> </ul> | RW     |
| 2    | OFCIEN   | FPU OFCI Interrupt Enable Bit | <ul style="list-style-type: none"> <li>● 0b0: Interrupt disabled</li> <li>● 0b1: Interrupt enabled</li> </ul> | RW     |
| 1    | DZCIEN   | FPU DZCI Interrupt Enable Bit | <ul style="list-style-type: none"> <li>● 0b0: Interrupt disabled</li> <li>● 0b1: Interrupt enabled</li> </ul> | RW     |
| 0    | IOCIEN   | FPU IOCI Interrupt Enable Bit | <ul style="list-style-type: none"> <li>● 0b0: Interrupt disabled</li> <li>● 0b1: Interrupt enabled</li> </ul> | RW     |

## 11.5 Precautions

For a description of ARM core interrupts, please refer to the "ARM Processor Cortex<sup>®</sup>-M4 Technical Reference Manual (ARM DDI 0439D)".



## 12 Automatic Operation System (AOS)

### 12.1 Introduction

The Automatic Operation System (AOS) is used to achieve coordination between peripherals without the assistance of the CPU. It utilizes events generated by peripherals as AOS sources, such as timer comparison matches, timer counting overflow, RTC periodic signals, various states of the sending and receiving data of the communication module (idle, receive data full, transmit data end, transmit data empty), ADC conversion end events, etc., to trigger actions of other peripherals. The triggered peripheral actions are referred to as AOS targets.

The AOS also equips 4 programmable logical units (PLUs) for logical operations between PORT and AOS sources. Users can select one or more AOS sources to trigger the same AOS target based on their needs.

### 12.2 Features

The AOS has multiple trigger sources, detailed in the [AOS Source Events](#). Unless specifically restricted, each AOS target can select one or more of these as trigger sources. When multiple trigger sources are required, the logic units (PLUs) equipped in the AOS can be used, allowing up to 4 trigger sources to be selected simultaneously. In addition, two common trigger sources can be selected through the common trigger source selection registers 1 and 2, which are shared by all AOS targets. Thus, for an AOS target, up to 6 trigger sources can be selected, and any of these can trigger the AOS target action when an event occurs.

Triggering can be done by hardware from peripheral circuits or by software through register writes.

The peripherals and their quantities that can serve as AOS targets are as follows:

- 4 DCU trigger targets
- 17 DMA trigger targets
- 1 Timer0 trigger target
- 1 Timer2 trigger target
- 4 Timer6 trigger targets
- 4 TimerA trigger targets
- 2 Event Port trigger targets
- 2 HASH trigger targets
- 1 OTS trigger target
- 3 sets of 2 ADC trigger targets for ADC1-ADC3 sequence triggering

12.3 Functional Description

12.3.1 Function Block Diagram

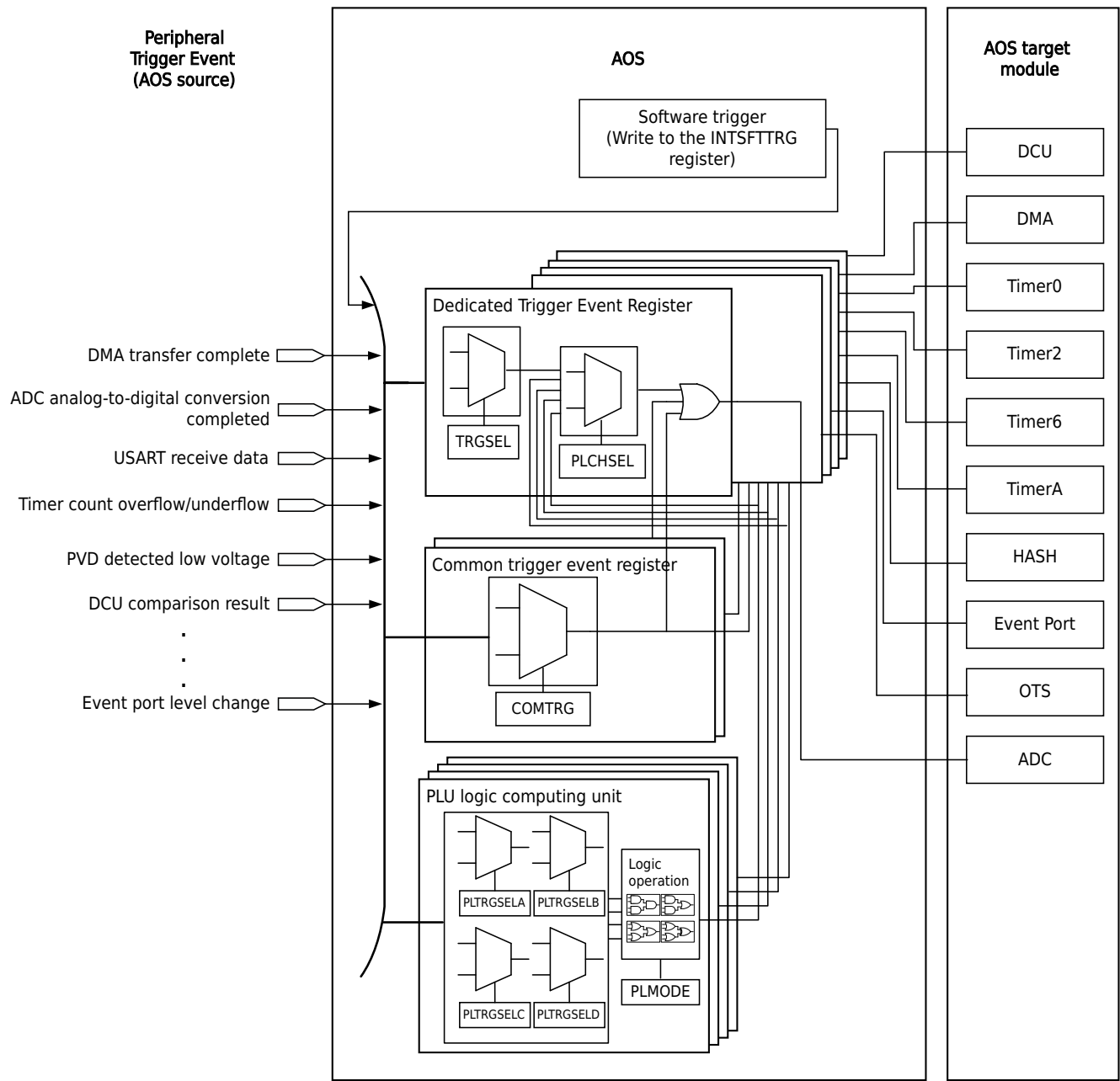


Figure 12-1 AOS Module Block Diagram

12.3.2 AOS Source Events and Targets

The AOS source event numbers are listed in the Interrupt Event List in the "Interrupt Controller (INTC) " chapter. Events marked with a "✓" in the "Trigger Source" column can be used as AOS sources. The AOS targets are listed in the table below.

Table 12-1 AOS Target List

| Modules | SN | Action                                                                                                            |
|---------|----|-------------------------------------------------------------------------------------------------------------------|
| DCU_1   | 0  | Trigger addition/subtraction operation, trigger data increment or decrement in triangular wave/sawtooth wave mode |

| Modules    | SN | Action                                                                                                            |
|------------|----|-------------------------------------------------------------------------------------------------------------------|
| DCU_2      | 1  | Trigger addition/subtraction operation, trigger data increment or decrement in triangular wave/sawtooth wave mode |
| DCU_3      | 2  | Trigger addition/subtraction operation, trigger data increment or decrement in triangular wave/sawtooth wave mode |
| DCU_4      | 3  | Trigger addition/subtraction operation, trigger data increment or decrement in triangular wave/sawtooth wave mode |
| DMA_1      | 4  | Trigger channel 0 starting transmission                                                                           |
|            | 5  | Trigger channel 1 starting transmission                                                                           |
|            | 6  | Trigger channel 2 starting transmission                                                                           |
|            | 7  | Trigger channel 3 starting transmission                                                                           |
|            | 8  | Trigger channel 4 starting transmission                                                                           |
|            | 9  | Trigger channel 5 starting transmission                                                                           |
|            | 10 | Trigger channel 6 starting transmission                                                                           |
|            | 11 | Trigger channel 7 starting transmission                                                                           |
| DMA_2      | 12 | Trigger channel 0 starting transmission                                                                           |
|            | 13 | Trigger channel 1 starting transmission                                                                           |
|            | 14 | Trigger channel 2 starting transmission                                                                           |
|            | 15 | Trigger channel 3 starting transmission                                                                           |
|            | 16 | Trigger channel 4 starting transmission                                                                           |
|            | 17 | Trigger channel 5 starting transmission                                                                           |
|            | 18 | Trigger channel 6 starting transmission                                                                           |
|            | 19 | Trigger channel 7 starting transmission                                                                           |
| DMA_1/2    | 20 | Trigger event trigger channel reset                                                                               |
| Timer6     | 21 | Trigger counter start/stop/clear/refresh/capture/increment/decrement                                              |
|            | 22 | Trigger counter start/stop/clear/refresh/capture/increment/decrement                                              |
|            | 23 | Trigger counter start/stop/clear/refresh/capture/increment/decrement                                              |
|            | 24 | Trigger counter start/stop/clear/refresh/capture/increment/decrement                                              |
| Event Port | 25 | Trigger Event Port 1/2 action                                                                                     |
|            | 26 | Trigger Event Port 3/4 action                                                                                     |
| Timer0     | 27 | Trigger counter start/stop/clear/capture                                                                          |
| Timer2     | 28 | Trigger counter start/stop/clear/capture                                                                          |
| HASH       | 29 | Trigger HASH operation start                                                                                      |
|            | 30 | Trigger HASH final operation start                                                                                |

| Modules | SN     | Action                                                           |
|---------|--------|------------------------------------------------------------------|
| TimerA  | 31     | Trigger Unit 1/5/9 counter start/stop/clear/increment/decrement  |
|         |        | Trigger Unit 2/6/10 counter input capture                        |
|         | 32     | Trigger Unit 2/6/10 counter start/stop/clear/increment/decrement |
|         |        | Trigger Unit 1/5/9 counter input capture                         |
|         | 33     | Trigger Unit 3/7/11 counter start/stop/clear/increment/decrement |
|         |        | Trigger Unit 4/8/12 counter input capture                        |
|         | 34     | Trigger Unit 4/8/12 counter start/stop/clear/increment/decrement |
|         |        | Trigger Unit 3/7/11 counter input capture                        |
| OTS     | 35     | Trigger OTS temperature measurement start                        |
| ADC_1   | 36, 37 | Trigger sequence A/B starting analog-to-digital conversion       |
| ADC_2   | 38, 39 | Trigger sequence A/B starting analog-to-digital conversion       |
| ADC_3   | 40, 41 | Trigger sequence A/B starting analog-to-digital conversion       |

### 12.3.3 Dedicated Trigger Source

For peripheral circuit modules with AOS targets, each AOS target is equipped with a dedicated peripheral trigger source selection register. When the event number corresponding to an AOS source is written to this register, the AOS target selects this AOS source as its trigger source. When an AOS source event occurs, the event is passed to the AOS target through the AOS, and the peripheral circuit of the AOS target starts its action based on its settings.

### 12.3.4 Common Trigger Source

In addition to the dedicated peripheral trigger source selection registers for each AOS target, the AOS also configures two common trigger source selection registers (AOS\_COMTRGSEL1, AOS\_COMTRGSEL2). These are used to achieve the function of triggering one or more AOS targets with the same trigger source, or triggering the same AOS target with multiple AOS trigger sources. To use them, first set the common trigger source enable bit to active in the dedicated peripheral trigger source selection register for the AOS target, and then write the event number corresponding to the AOS source in the common trigger source selection register. When an AOS source event occurs, the event is passed to the AOS target through the common trigger source of the AOS, and the peripheral circuit of the AOS target starts its action based on its settings. When both dedicated and common trigger sources are set, up to 3 AOS sources can trigger the same AOS target simultaneously. Any of the 3 AOS sources can trigger the AOS target when an event occurs.

All AOS targets share these two common trigger sources. Therefore, when other AOS targets are not using the event selected by the common trigger source selection register, the common trigger source enable bit should be set to inactive in its dedicated peripheral trigger source selection register to prevent erroneous triggering.

### 12.3.5 Programmable Logic Unit (PLU)

The AOS is equipped with a total of 4 programmable logic units (PLUs), each with 4 inputs (PLINA, PLINB, PLINC, PLIND) and 1 output (PLOUT).

The sources of the four inputs PLINA-D can be AOS trigger sources, PORT inputs, or fixed values. The result of the logical operation can be used not only for AOS trigger target modules but also output through PORT.

Set the AOS\_PLUX\_CR.PLINySEL (x=0-3, y=A, B, C, D) bit to select the input source for PLINA-D. Set the AOS\_PLUX\_CR.PLMODE bit to select the desired logical operation.

For configurations of PLUX\_INA, PLUX\_INB, PLUX\_INC, PLUX\_IND, and PLUX\_OUT, please refer to the "Pin Function Description" chapter in the Datasheet.

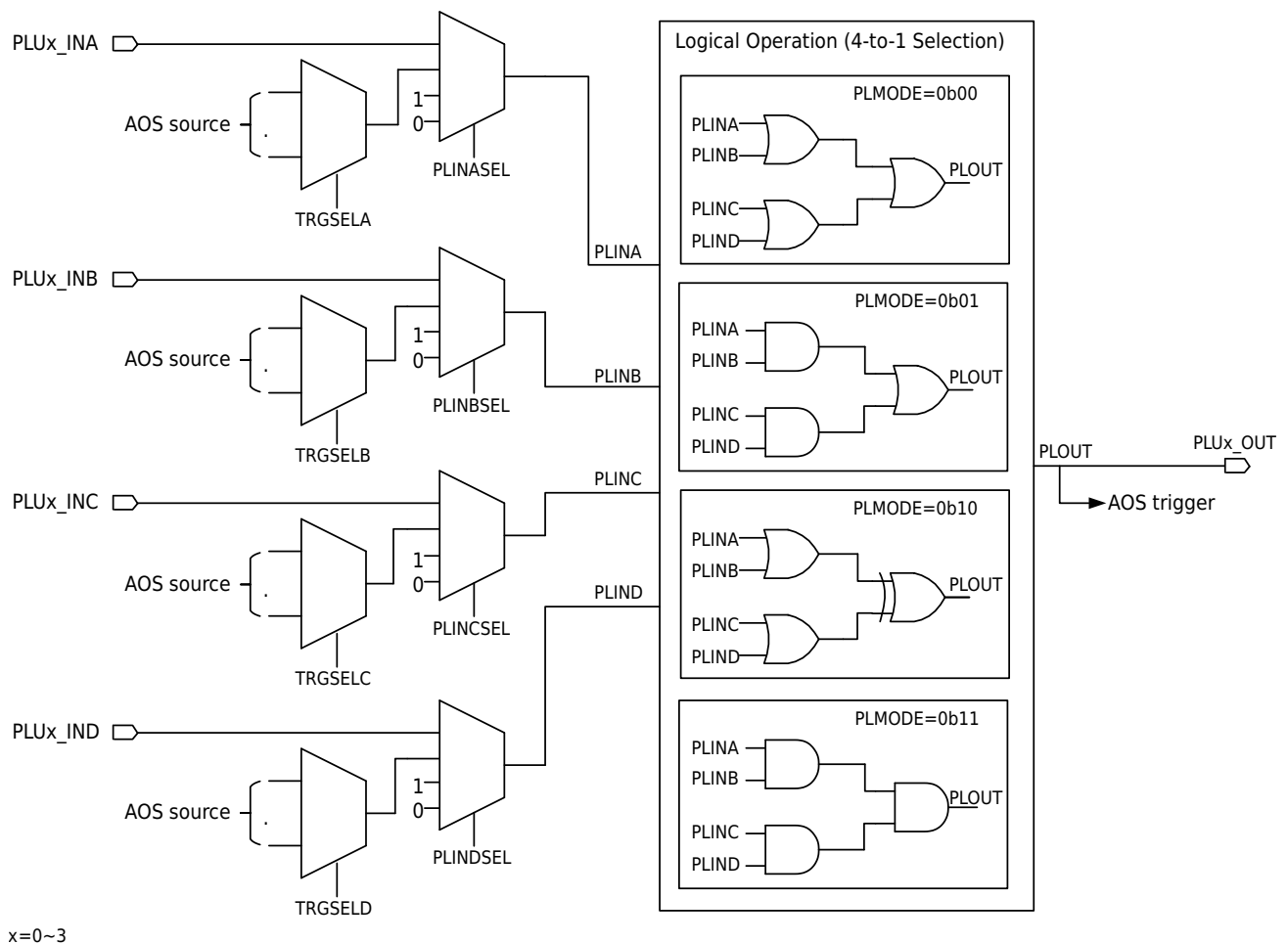


Figure 12-2 PLU Block Diagram

## 12.4 Register Description

### 12.4.1 Register List

Base address: 0x40010800

Table 12-2 Automatic Operation System (AOS) Register Offset

| Offset        | Register                         | Description                                                             | Width |
|---------------|----------------------------------|-------------------------------------------------------------------------|-------|
| 0x00          | <a href="#">INTSFTTRG</a>        | Software Trigger Event Register                                         | 32    |
| 0x04*x        | <a href="#">DCU_TRGSELx</a>      | DCU Trigger Source Selection Register (x=1-4)                           | 32    |
| 0x14+0x04*x   | <a href="#">DMA1_TRGSELx</a>     | DMA1 Transfer Start Trigger Source Selection Register (x=0-7)           | 32    |
| 0x34+0x04*x   | <a href="#">DMA2_TRGSELx</a>     | DMA2 Transfer Start Trigger Source Selection Register (x=0-7)           | 32    |
| 0x54          | <a href="#">DMA_RC_TRGSEL</a>    | DMA Re-config Trigger Source Selection Register                         | 32    |
| 0x58+0x04*x   | <a href="#">TMR6_TRGSELx</a>     | Timer6 Hardware Trigger Event Selection Register (x=0-3)                | 32    |
| 0x68          | <a href="#">PEVNT_TRGSEL12</a>   | Event Port1/2 Trigger Source Selection Register                         | 32    |
| 0x6C          | <a href="#">PEVNT_TRGSEL34</a>   | Event Port3/4 Trigger Source Selection Register                         | 32    |
| 0x70          | <a href="#">TMR0_TRGSEL</a>      | Timer0 Hardware Trigger Event Selection Register                        | 32    |
| 0x74          | <a href="#">TMR2_TRGSEL</a>      | Timer2 Hardware Trigger Event Selection Register                        | 32    |
| 0x78          | <a href="#">HASH_TRGSELA</a>     | HASH Hardware Trigger Event Selection Register A                        | 32    |
| 0x7C          | <a href="#">HASH_TRGSELB</a>     | HASH Hardware Trigger Event Selection Register B                        | 32    |
| 0x80+0x04*x   | <a href="#">TMRA_TRGSELx</a>     | TimerA Hardware Trigger Event Selection Register (x=0-3)                | 32    |
| 0x90          | <a href="#">OTS_TRGSEL</a>       | OTS Hardware Trigger Event Selection Register                           | 32    |
| 0x94+0x04*x   | <a href="#">ADC1_TRGSELx</a>     | ADC1 Conversion Start On-Chip Trigger Source Selection Register (x=0-1) | 32    |
| 0x9C+0x04*x   | <a href="#">ADC2_TRGSELx</a>     | ADC2 Conversion Start On-Chip Trigger Source Selection Register (x=0-1) | 32    |
| 0xA4+0x04*x   | <a href="#">ADC3_TRGSELx</a>     | ADC3 Conversion Start On-Chip Trigger Source Selection Register (x=0-1) | 32    |
| 0xAC          | <a href="#">AOS_COMTRGSEL1</a>   | Common Trigger Source Selection Register 1                              | 32    |
| 0xB0          | <a href="#">AOS_COMTRGSEL2</a>   | Common Trigger Source Selection Register 2                              | 32    |
| 0x200+0x004*x | <a href="#">AOS_PLUX_CR</a>      | PLU Control Register (x=0-3)                                            | 32    |
| 0x210+0x010*x | <a href="#">AOS_PLUX_TRGSELA</a> | PLU Trigger Source Selection Register A (x=0-3)                         | 32    |

| Offset        | Register                         | Description                                     | Width |
|---------------|----------------------------------|-------------------------------------------------|-------|
| 0x214+0x010*x | <a href="#">AOS_PLUx_TRGSELB</a> | PLU Trigger Source Selection Register B (x=0-3) | 32    |
| 0x218+0x010*x | <a href="#">AOS_PLUx_TRGSELC</a> | PLU Trigger Source Selection Register C (x=0-3) | 32    |
| 0x21C+0x010*x | <a href="#">AOS_PLUx_TRGSELD</a> | PLU Trigger Source Selection Register D (x=0-3) | 32    |

### 12.4.2 Software Trigger Event Register (INTSFTRG)

Writing 1 to this register generates a trigger event each time.

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |      |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|------|
| 0x00   | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0    |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |      |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   | STRG |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   | WO   |

| Bit  | Name     | Function         | Description                                                            | Access |
|------|----------|------------------|------------------------------------------------------------------------|--------|
| 31:1 | Reserved | Reserved         | -                                                                      | -      |
| 0    | STRG     | Software Trigger | Writing 1 generates a software trigger event; writing 0 has no effect. | WO     |



### 12.4.3 DCU Trigger Source Selection Register (DCU\_TRGSELx) (x=1-4)

After selecting the hardware trigger start mode in the DCU, use this register to select the trigger event source. When PLCHSEL[2:0] is set to 0b000, write the trigger event number to the TRGSEL[8:0] bits in this register. When PLCHSEL[2:0] is set to 0b001, 0b010, 0b011, or 0b100, write the trigger event number to the corresponding PLU Trigger Source Selection Register (AOS\_PLU\_TRGSELA/B/C/D), and configure the PLU Control Register (AOS\_PLU\_CR) accordingly. When the peripheral event corresponding to the number occurs, the DCU will be triggered to start and perform calculations by this event.

Each DCU unit corresponds to one register.

Offset: 0x04\*x

| Offset | Bit Position |          |    |    |    |    |    |    |    |    |    |    |    |         |          |    |    |    |    |    |    |    |        |   |   |   |   |   |   |   |   |   |
|--------|--------------|----------|----|----|----|----|----|----|----|----|----|----|----|---------|----------|----|----|----|----|----|----|----|--------|---|---|---|---|---|---|---|---|---|
| -      | 31           | 30       | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18      | 17       | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9      | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x000001FF   |          |    |    |    |    |    |    |    |    |    |    |    |         |          |    |    |    |    |    |    |    |        |   |   |   |   |   |   |   |   |   |
| Name   | COMEN        | Reserved |    |    |    |    |    |    |    |    |    |    |    | PLCHSEL | Reserved |    |    |    |    |    |    |    | TRGSEL |   |   |   |   |   |   |   |   |   |
| Access | RW           |          |    |    |    |    |    |    |    |    |    |    |    | RW      |          |    |    |    |    |    |    |    | RW     |   |   |   |   |   |   |   |   |   |

| Bit   | Name     | Function                 | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | Access |
|-------|----------|--------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31    | COMEN[1] | Common trigger enabled   | <ul style="list-style-type: none"> <li>0b0: Disable triggering of this AOS target by the common trigger event of AOS_COMTRGSEL2</li> <li>0b1: Enable triggering of this AOS target by the common trigger event of AOS_COMTRGSEL2</li> </ul>                                                                                                                                                                                                                                               | RW     |
| 30    | COMEN[0] | Common trigger enabled   | <ul style="list-style-type: none"> <li>0b0: Disable triggering of this AOS target by the common trigger event of AOS_COMTRGSEL1</li> <li>0b1: Enable triggering of this AOS target by the common trigger event of AOS_COMTRGSEL1</li> </ul>                                                                                                                                                                                                                                               | RW     |
| 29:19 | Reserved | Reserved                 | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | -      |
| 18:16 | PLCHSEL  | PLU selection bit        | <ul style="list-style-type: none"> <li>0b000: Trigger this AOS target using the event selected by the TRGSEL[8:0] bits</li> <li>0b001: Trigger this AOS target using the event after PLU0 operation</li> <li>0b010: Trigger this AOS target using the event after PLU1 operation</li> <li>0b011: Trigger this AOS target using the event after PLU2 operation</li> <li>0b100: Trigger this AOS target using the event after PLU3 operation</li> <li>Others: Setting prohibited</li> </ul> | RW     |
| 15:9  | Reserved | Reserved                 | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | -      |
| 8:0   | TRGSEL   | Trigger source selection | Write the number of the peripheral circuit event to be selected                                                                                                                                                                                                                                                                                                                                                                                                                           | RW     |

### 12.4.4 DMA1 Transfer Start Trigger Source Selection Register (DMA1\_TRGSELx) (x=0-7)

After enabling DMA1, use this register to select the trigger event source. When PLCHSEL[2:0] is set to 0b000, write the trigger event number to the TRGSEL[8:0] bits in this register. When PLCHSEL[2:0] is set to 0b001, 0b010, 0b011, or 0b100, write the trigger event number to the corresponding PLU Trigger Source Selection Register (AOS\_PLU\_TRGSELA/B/C/D), and configure the PLU Control Register (AOS\_PLU\_CR) accordingly. When the peripheral event corresponding to the number occurs, DMA1 will be triggered to start and perform the transfer by this event.

Each data transfer channel corresponds to one register.

Offset: 0x14+0x04\*x

| Offset | Bit Position |          |    |    |    |    |    |    |    |    |    |    |    |         |          |    |    |    |    |    |    |    |        |   |   |   |   |   |   |   |   |   |
|--------|--------------|----------|----|----|----|----|----|----|----|----|----|----|----|---------|----------|----|----|----|----|----|----|----|--------|---|---|---|---|---|---|---|---|---|
| -      | 31           | 30       | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18      | 17       | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9      | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x000001FF   |          |    |    |    |    |    |    |    |    |    |    |    |         |          |    |    |    |    |    |    |    |        |   |   |   |   |   |   |   |   |   |
| Name   | COMEN        | Reserved |    |    |    |    |    |    |    |    |    |    |    | PLCHSEL | Reserved |    |    |    |    |    |    |    | TRGSEL |   |   |   |   |   |   |   |   |   |
| Access | RW           |          |    |    |    |    |    |    |    |    |    |    |    | RW      |          |    |    |    |    |    |    |    | RW     |   |   |   |   |   |   |   |   |   |

| Bit   | Name     | Function                 | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | Access |
|-------|----------|--------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31    | COMEN[1] | Common trigger enabled   | <ul style="list-style-type: none"> <li>0b0: Disable triggering of this AOS target by the common trigger event of AOS_COMTRGSEL2</li> <li>0b1: Enable triggering of this AOS target by the common trigger event of AOS_COMTRGSEL2</li> </ul>                                                                                                                                                                                                                                               | RW     |
| 30    | COMEN[0] | Common trigger enabled   | <ul style="list-style-type: none"> <li>0b0: Disable triggering of this AOS target by the common trigger event of AOS_COMTRGSEL1</li> <li>0b1: Enable triggering of this AOS target by the common trigger event of AOS_COMTRGSEL1</li> </ul>                                                                                                                                                                                                                                               | RW     |
| 29:19 | Reserved | Reserved                 | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | -      |
| 18:16 | PLCHSEL  | PLU selection bit        | <ul style="list-style-type: none"> <li>0b000: Trigger this AOS target using the event selected by the TRGSEL[8:0] bits</li> <li>0b001: Trigger this AOS target using the event after PLU0 operation</li> <li>0b010: Trigger this AOS target using the event after PLU1 operation</li> <li>0b011: Trigger this AOS target using the event after PLU2 operation</li> <li>0b100: Trigger this AOS target using the event after PLU3 operation</li> <li>Others: Setting prohibited</li> </ul> | RW     |
| 15:9  | Reserved | Reserved                 | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | -      |
| 8:0   | TRGSEL   | Trigger source selection | Write the number of the peripheral circuit event to be selected                                                                                                                                                                                                                                                                                                                                                                                                                           | RW     |

### 12.4.5 DMA2 Transfer Start Trigger Source Selection Register (DMA2\_TRGSELx) (x=0-7)

After enabling DMA2, use this register to select the trigger event source. When PLCHSEL[2:0] is set to 0b000, write the trigger event number to the TRGSEL[8:0] bits in this register. When PLCHSEL[2:0] is set to 0b001, 0b010, 0b011, or 0b100, write the trigger event number to the corresponding PLU Trigger Source Selection Register (AOS\_PLU\_TRGSELA/B/C/D), and configure the AOS\_PLU\_CR Register accordingly. When the peripheral event corresponding to the number occurs, DMA2 will be triggered to start and perform the transfer by this event.

Each data transfer channel corresponds to one register.

Offset: 0x34+0x04\*x

| Offset | Bit Position |          |    |    |    |    |    |    |    |    |    |    |    |         |          |    |    |    |    |    |    |    |   |   |        |   |   |   |   |   |   |   |
|--------|--------------|----------|----|----|----|----|----|----|----|----|----|----|----|---------|----------|----|----|----|----|----|----|----|---|---|--------|---|---|---|---|---|---|---|
| -      | 31           | 30       | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18      | 17       | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7      | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x000001FF   |          |    |    |    |    |    |    |    |    |    |    |    |         |          |    |    |    |    |    |    |    |   |   |        |   |   |   |   |   |   |   |
| Name   | COMEN        | Reserved |    |    |    |    |    |    |    |    |    |    |    | PLCHSEL | Reserved |    |    |    |    |    |    |    |   |   | TRGSEL |   |   |   |   |   |   |   |
| Access | RW           |          |    |    |    |    |    |    |    |    |    |    |    | RW      |          |    |    |    |    |    |    |    |   |   | RW     |   |   |   |   |   |   |   |

| Bit   | Name     | Function                 | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | Access |
|-------|----------|--------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31    | COMEN[1] | Common trigger enabled   | <ul style="list-style-type: none"> <li>0b0: Disable triggering of this AOS target by the common trigger event of AOS_COMTRGSEL2</li> <li>0b1: Enable triggering of this AOS target by the common trigger event of AOS_COMTRGSEL2</li> </ul>                                                                                                                                                                                                                                               | RW     |
| 30    | COMEN[0] | Common trigger enabled   | <ul style="list-style-type: none"> <li>0b0: Disable triggering of this AOS target by the common trigger event of AOS_COMTRGSEL1</li> <li>0b1: Enable triggering of this AOS target by the common trigger event of AOS_COMTRGSEL1</li> </ul>                                                                                                                                                                                                                                               | RW     |
| 29:19 | Reserved | Reserved                 | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | -      |
| 18:16 | PLCHSEL  | PLU selection bit        | <ul style="list-style-type: none"> <li>0b000: Trigger this AOS target using the event selected by the TRGSEL[8:0] bits</li> <li>0b001: Trigger this AOS target using the event after PLU0 operation</li> <li>0b010: Trigger this AOS target using the event after PLU1 operation</li> <li>0b011: Trigger this AOS target using the event after PLU2 operation</li> <li>0b100: Trigger this AOS target using the event after PLU3 operation</li> <li>Others: Setting prohibited</li> </ul> | RW     |
| 15:9  | Reserved | Reserved                 | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | -      |
| 8:0   | TRGSEL   | Trigger source selection | Write the number of the peripheral circuit event to be selected                                                                                                                                                                                                                                                                                                                                                                                                                           | RW     |

### 12.4.6 DMA Re-config Trigger Source Selection Register (DMA\_RC\_TRGSEL)

After enabling DMA, use this register to select the trigger event source. When PLCHSEL[2:0] is set to 0b000, write the trigger event number to the TRGSEL[8:0] bits in this register. When PLCHSEL[2:0] is set to 0b001, 0b010, 0b011, or 0b100, write the trigger event number to the corresponding PLU Trigger Source Selection Register (AOS\_PLU\_TRGSELA/B/C/D), and configure the AOS\_PLU\_CR accordingly. When the peripheral event corresponding to the number occurs, DMA will be triggered to reset the channel.

DMA1 and DMA2 share this register.

| Offset | Bit Position |          |    |    |    |    |    |    |    |    |    |         |          |    |    |    |    |    |    |    |    |    |        |   |   |   |   |   |   |   |   |   |  |  |  |  |  |  |
|--------|--------------|----------|----|----|----|----|----|----|----|----|----|---------|----------|----|----|----|----|----|----|----|----|----|--------|---|---|---|---|---|---|---|---|---|--|--|--|--|--|--|
| 0x54   | 31           | 30       | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20      | 19       | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9      | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |  |  |  |  |  |  |
| Reset  | 0x000001FF   |          |    |    |    |    |    |    |    |    |    |         |          |    |    |    |    |    |    |    |    |    |        |   |   |   |   |   |   |   |   |   |  |  |  |  |  |  |
| Name   | COMEN        | Reserved |    |    |    |    |    |    |    |    |    | PLCHSEL | Reserved |    |    |    |    |    |    |    |    |    | TRGSEL |   |   |   |   |   |   |   |   |   |  |  |  |  |  |  |
| Access | RW           |          |    |    |    |    |    |    |    |    |    | RW      |          |    |    |    |    |    |    |    |    |    | RW     |   |   |   |   |   |   |   |   |   |  |  |  |  |  |  |

| Bit   | Name     | Function                 | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | Access |
|-------|----------|--------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31    | COMEN[1] | Common trigger enabled   | <ul style="list-style-type: none"> <li>0b0: Disable triggering of this AOS target by the common trigger event of AOS_COMTRGSEL2</li> <li>0b1: Enable triggering of this AOS target by the common trigger event of AOS_COMTRGSEL2</li> </ul>                                                                                                                                                                                                                                               | RW     |
| 30    | COMEN[0] | Common trigger enabled   | <ul style="list-style-type: none"> <li>0b0: Disable triggering of this AOS target by the common trigger event of AOS_COMTRGSEL1</li> <li>0b1: Enable triggering of this AOS target by the common trigger event of AOS_COMTRGSEL1</li> </ul>                                                                                                                                                                                                                                               | RW     |
| 29:19 | Reserved | Reserved                 | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | -      |
| 18:16 | PLCHSEL  | PLU selection bit        | <ul style="list-style-type: none"> <li>0b000: Trigger this AOS target using the event selected by the TRGSEL[8:0] bits</li> <li>0b001: Trigger this AOS target using the event after PLU0 operation</li> <li>0b010: Trigger this AOS target using the event after PLU1 operation</li> <li>0b011: Trigger this AOS target using the event after PLU2 operation</li> <li>0b100: Trigger this AOS target using the event after PLU3 operation</li> <li>Others: Setting prohibited</li> </ul> | RW     |
| 15:9  | Reserved | Reserved                 | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | -      |
| 8:0   | TRGSEL   | Trigger source selection | Write the number of the peripheral circuit event to be selected                                                                                                                                                                                                                                                                                                                                                                                                                           | RW     |

### 12.4.7 Timer6 Hardware Trigger Event Selection Register (TMR6\_TRGSELx) (x=0-3)

After selecting internal trigger hardware synchronization in Timer6, use this register to select the trigger event source. When PLCHSEL[2:0] is set to 0b000, write the trigger event number to the TRGSEL[8:0] bits in this register. When PLCHSEL[2:0] is set to 0b001, 0b010, 0b011, or 0b100, write the trigger event number to the corresponding PLU Trigger Source Selection Register (AOS\_PLU\_TRGSELA/B/C/D), and configure the AOS\_PLU\_CR Register accordingly. When the peripheral circuit event corresponding to the number occurs, Timer6 will be triggered for hardware synchronization (including start/stop/clear/refresh/increment/decrement/input capture) by this event.

All Timer6 units share these 4 registers.

Offset: 0x58+0x04\*x

| Offset | Bit Position |          |    |    |    |    |    |    |    |    |    |    |    |         |          |    |    |    |    |    |    |    |        |   |   |   |   |   |   |   |   |   |
|--------|--------------|----------|----|----|----|----|----|----|----|----|----|----|----|---------|----------|----|----|----|----|----|----|----|--------|---|---|---|---|---|---|---|---|---|
| -      | 31           | 30       | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18      | 17       | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9      | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x000001FF   |          |    |    |    |    |    |    |    |    |    |    |    |         |          |    |    |    |    |    |    |    |        |   |   |   |   |   |   |   |   |   |
| Name   | COMEN        | Reserved |    |    |    |    |    |    |    |    |    |    |    | PLCHSEL | Reserved |    |    |    |    |    |    |    | TRGSEL |   |   |   |   |   |   |   |   |   |
| Access | RW           |          |    |    |    |    |    |    |    |    |    |    |    | RW      |          |    |    |    |    |    |    |    | RW     |   |   |   |   |   |   |   |   |   |

| Bit   | Name     | Function                 | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | Access |
|-------|----------|--------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31    | COMEN[1] | Common trigger enabled   | <ul style="list-style-type: none"> <li>0b0: Disable triggering of this AOS target by the common trigger event of AOS_COMTRGSEL2</li> <li>0b1: Enable triggering of this AOS target by the common trigger event of AOS_COMTRGSEL2</li> </ul>                                                                                                                                                                                                                                               | RW     |
| 30    | COMEN[0] | Common trigger enabled   | <ul style="list-style-type: none"> <li>0b0: Disable triggering of this AOS target by the common trigger event of AOS_COMTRGSEL1</li> <li>0b1: Enable triggering of this AOS target by the common trigger event of AOS_COMTRGSEL1</li> </ul>                                                                                                                                                                                                                                               | RW     |
| 29:19 | Reserved | Reserved                 | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | -      |
| 18:16 | PLCHSEL  | PLU selection bit        | <ul style="list-style-type: none"> <li>0b000: Trigger this AOS target using the event selected by the TRGSEL[8:0] bits</li> <li>0b001: Trigger this AOS target using the event after PLU0 operation</li> <li>0b010: Trigger this AOS target using the event after PLU1 operation</li> <li>0b011: Trigger this AOS target using the event after PLU2 operation</li> <li>0b100: Trigger this AOS target using the event after PLU3 operation</li> <li>Others: Setting prohibited</li> </ul> | RW     |
| 15:9  | Reserved | Reserved                 | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | -      |
| 8:0   | TRGSEL   | Trigger source selection | Write the number of the peripheral circuit event to be selected                                                                                                                                                                                                                                                                                                                                                                                                                           | RW     |

### 12.4.8 Event Port Trigger Source Selection Register (PEVNT\_TRGSEL12/PEVNT\_TRGSEL34)

Write the trigger event number to this register to select the trigger event source. When PLCHSEL[2:0] is set to 0b000, write the trigger event number to the TRGSEL[8:0] bits in this register. When PLCHSEL[2:0] is set to 0b001, 0b010, 0b011, or 0b100, write the trigger event number to the corresponding PLU Trigger Source Selection Register (AOS\_PLU\_TRGSELA/B/C/D), and configure the AOS\_PLU\_CR Register accordingly. The Event Port is triggered to output the specified level or latch the port status.

Event Port1/2 share the PEVNT\_TRGSEL12 register, and Event Port3/4 share the PEVNT\_TRGSEL34 register.

Offset: 0x68 (PEVNT\_TRGSEL12), 0x6C (PEVNT\_TRGSEL34)

| Offset | Bit Position |          |    |    |    |    |    |    |    |    |    |    |    |         |          |    |    |    |    |    |    |    |        |   |   |   |   |   |   |   |   |   |
|--------|--------------|----------|----|----|----|----|----|----|----|----|----|----|----|---------|----------|----|----|----|----|----|----|----|--------|---|---|---|---|---|---|---|---|---|
| -      | 31           | 30       | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18      | 17       | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9      | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x000001FF   |          |    |    |    |    |    |    |    |    |    |    |    |         |          |    |    |    |    |    |    |    |        |   |   |   |   |   |   |   |   |   |
| Name   | COMEN        | Reserved |    |    |    |    |    |    |    |    |    |    |    | PLCHSEL | Reserved |    |    |    |    |    |    |    | TRGSEL |   |   |   |   |   |   |   |   |   |
| Access | RW           |          |    |    |    |    |    |    |    |    |    |    |    | RW      |          |    |    |    |    |    |    |    | RW     |   |   |   |   |   |   |   |   |   |

| Bit   | Name     | Function                 | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | Access |
|-------|----------|--------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31    | COMEN[1] | Common trigger enabled   | <ul style="list-style-type: none"> <li>0b0: Disable triggering of this AOS target by the common trigger event of AOS_COMTRGSEL2</li> <li>0b1: Enable triggering of this AOS target by the common trigger event of AOS_COMTRGSEL2</li> </ul>                                                                                                                                                                                                                                               | RW     |
| 30    | COMEN[0] | Common trigger enabled   | <ul style="list-style-type: none"> <li>0b0: Disable triggering of this AOS target by the common trigger event of AOS_COMTRGSEL1</li> <li>0b1: Enable triggering of this AOS target by the common trigger event of AOS_COMTRGSEL1</li> </ul>                                                                                                                                                                                                                                               | RW     |
| 29:19 | Reserved | Reserved                 | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | -      |
| 18:16 | PLCHSEL  | PLU selection bit        | <ul style="list-style-type: none"> <li>0b000: Trigger this AOS target using the event selected by the TRGSEL[8:0] bits</li> <li>0b001: Trigger this AOS target using the event after PLU0 operation</li> <li>0b010: Trigger this AOS target using the event after PLU1 operation</li> <li>0b011: Trigger this AOS target using the event after PLU2 operation</li> <li>0b100: Trigger this AOS target using the event after PLU3 operation</li> <li>Others: Setting prohibited</li> </ul> | RW     |
| 15:9  | Reserved | Reserved                 | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | -      |
| 8:0   | TRGSEL   | Trigger source selection | Write the number of the peripheral circuit event to be selected                                                                                                                                                                                                                                                                                                                                                                                                                           | RW     |

### 12.4.9 Timer0 Hardware Trigger Event Selection Register (TMR0\_TRGSEL)

After selecting hardware trigger in Timer0, use this register to select the trigger event source. When PLCHSEL[2:0] is set to 0b000, write the trigger event number to the TRGSEL[8:0] bits in this register. When PLCHSEL[2:0] is set to 0b001, 0b010, 0b011, or 0b0100, write the trigger event number to the corresponding PLU Trigger Source Selection Register (AOS\_PLU\_TRGSELA/B/C/D), and configure the PLU Control Register (AOS\_PLU\_CR) accordingly. When the peripheral event corresponding to the number occurs, Timer0 will be triggered to act (including start/stop/clear/input capture) by this event.

All Timer0 units and their channels share this register.

| Offset | Bit Position |          |    |    |    |    |    |    |    |    |    |    |    |         |    |    |    |          |    |    |    |    |   |   |   |        |   |   |   |   |   |   |  |  |  |
|--------|--------------|----------|----|----|----|----|----|----|----|----|----|----|----|---------|----|----|----|----------|----|----|----|----|---|---|---|--------|---|---|---|---|---|---|--|--|--|
| 0x70   | 31           | 30       | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18      | 17 | 16 | 15 | 14       | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6      | 5 | 4 | 3 | 2 | 1 | 0 |  |  |  |
| Reset  | 0x000001FF   |          |    |    |    |    |    |    |    |    |    |    |    |         |    |    |    |          |    |    |    |    |   |   |   |        |   |   |   |   |   |   |  |  |  |
| Name   | COMEN        | Reserved |    |    |    |    |    |    |    |    |    |    |    | PLCHSEL |    |    |    | Reserved |    |    |    |    |   |   |   | TRGSEL |   |   |   |   |   |   |  |  |  |
| Access | RW           |          |    |    |    |    |    |    |    |    |    |    |    | RW      |    |    |    |          |    |    |    |    |   |   |   | RW     |   |   |   |   |   |   |  |  |  |

| Bit   | Name     | Function                 | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | Access |
|-------|----------|--------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31    | COMEN[1] | Common trigger enabled   | <ul style="list-style-type: none"> <li>0b0: Disable triggering of this AOS target by the common trigger event of AOS_COMTRGSEL2</li> <li>0b1: Enable triggering of this AOS target by the common trigger event of AOS_COMTRGSEL2</li> </ul>                                                                                                                                                                                                                                               | RW     |
| 30    | COMEN[0] | Common trigger enabled   | <ul style="list-style-type: none"> <li>0b0: Disable triggering of this AOS target by the common trigger event of AOS_COMTRGSEL1</li> <li>0b1: Enable triggering of this AOS target by the common trigger event of AOS_COMTRGSEL1</li> </ul>                                                                                                                                                                                                                                               | RW     |
| 29:19 | Reserved | Reserved                 | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | -      |
| 18:16 | PLCHSEL  | PLU selection bit        | <ul style="list-style-type: none"> <li>0b000: Trigger this AOS target using the event selected by the TRGSEL[8:0] bits</li> <li>0b001: Trigger this AOS target using the event after PLU0 operation</li> <li>0b010: Trigger this AOS target using the event after PLU1 operation</li> <li>0b011: Trigger this AOS target using the event after PLU2 operation</li> <li>0b100: Trigger this AOS target using the event after PLU3 operation</li> <li>Others: Setting prohibited</li> </ul> | RW     |
| 15:9  | Reserved | Reserved                 | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | -      |
| 8:0   | TRGSEL   | Trigger source selection | Write the number of the peripheral circuit event to be selected                                                                                                                                                                                                                                                                                                                                                                                                                           | RW     |



### 12.4.10 Timer2 Hardware Trigger Event Selection Register (TMR2\_TRGSEL)

After selecting hardware trigger in Timer2, use this register to select the trigger event source. When PLCHSEL[2:0] is set to 0b000, write the trigger event number to the TRGSEL[8:0] bits in this register. When PLCHSEL[2:0] is set to 0b001, 0b010, 0b011, or 0b100, write the trigger event number to the corresponding PLU Trigger Source Selection Register (AOS\_PLU\_TRGSELA/B/C/D), and configure the AOS\_PLU\_CR Register accordingly. When the peripheral event corresponding to the number occurs, Timer2 will be triggered to act (including start/stop/clear/input capture) by this event.

All Timer2 units and their channels share this register.

| Offset | Bit Position |          |    |    |    |    |    |    |    |    |    |    |    |         |          |    |    |    |    |    |    |    |        |   |   |   |   |   |   |   |   |   |  |  |  |  |  |  |
|--------|--------------|----------|----|----|----|----|----|----|----|----|----|----|----|---------|----------|----|----|----|----|----|----|----|--------|---|---|---|---|---|---|---|---|---|--|--|--|--|--|--|
| 0x74   | 31           | 30       | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18      | 17       | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9      | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |  |  |  |  |  |  |
| Reset  | 0x000001FF   |          |    |    |    |    |    |    |    |    |    |    |    |         |          |    |    |    |    |    |    |    |        |   |   |   |   |   |   |   |   |   |  |  |  |  |  |  |
| Name   | COMEN        | Reserved |    |    |    |    |    |    |    |    |    |    |    | PLCHSEL | Reserved |    |    |    |    |    |    |    | TRGSEL |   |   |   |   |   |   |   |   |   |  |  |  |  |  |  |
| Access | RW           |          |    |    |    |    |    |    |    |    |    |    |    | RW      |          |    |    |    |    |    |    |    | RW     |   |   |   |   |   |   |   |   |   |  |  |  |  |  |  |

| Bit   | Name     | Function                 | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | Access |
|-------|----------|--------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31    | COMEN[1] | Common trigger enabled   | <ul style="list-style-type: none"> <li>0b0: Disable triggering of this AOS target by the common trigger event of AOS_COMTRGSEL2</li> <li>0b1: Enable triggering of this AOS target by the common trigger event of AOS_COMTRGSEL2</li> </ul>                                                                                                                                                                                                                                               | RW     |
| 30    | COMEN[0] | Common trigger enabled   | <ul style="list-style-type: none"> <li>0b0: Disable triggering of this AOS target by the common trigger event of AOS_COMTRGSEL1</li> <li>0b1: Enable triggering of this AOS target by the common trigger event of AOS_COMTRGSEL1</li> </ul>                                                                                                                                                                                                                                               | RW     |
| 29:19 | Reserved | Reserved                 | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | -      |
| 18:16 | PLCHSEL  | PLU selection bit        | <ul style="list-style-type: none"> <li>0b000: Trigger this AOS target using the event selected by the TRGSEL[8:0] bits</li> <li>0b001: Trigger this AOS target using the event after PLU0 operation</li> <li>0b010: Trigger this AOS target using the event after PLU1 operation</li> <li>0b011: Trigger this AOS target using the event after PLU2 operation</li> <li>0b100: Trigger this AOS target using the event after PLU3 operation</li> <li>Others: Setting prohibited</li> </ul> | RW     |
| 15:9  | Reserved | Reserved                 | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | -      |
| 8:0   | TRGSEL   | Trigger source selection | Write the number of the peripheral circuit event to be selected                                                                                                                                                                                                                                                                                                                                                                                                                           | RW     |



### 12.4.11 HASH Hardware Trigger Event Selection Register (HASH\_TRGSELx) (x=A/B)

After writing data to HASH\_DR, use the HASH\_TRGSELA register to select the trigger event source to start the HASH operation. After writing the last set of data to HASH\_DR, use the HASH\_TRGSELB register to select the trigger event source to notify HASH to perform the final operation. When PLCHSEL[2:0] is set to 0b000, write the trigger event number to the TRGSEL[8:0] bits in this register. When PLCHSEL[2:0] is set to 0b001, 0b010, 0b011, or 0b100, write the trigger event number to the corresponding PLU Trigger Source Selection Register (AOS\_PLU\_TRGSELA/B/C/D), and configure the AOS\_PLU\_CR Register accordingly. When the peripheral event corresponding to the number occurs, HASH will be triggered to start the operation by this event.



#### Notice

Do not select trigger signals other than DMA\_BTCx (x=0-7).

Offset: 0x78 (HASH\_TRGSELA), 0x7C (HASH\_TRGSELB)

| Offset | Bit Position |          |    |    |    |    |    |    |    |    |    |    |    |         |          |    |    |    |    |    |    |    |        |   |   |   |   |   |   |   |   |   |
|--------|--------------|----------|----|----|----|----|----|----|----|----|----|----|----|---------|----------|----|----|----|----|----|----|----|--------|---|---|---|---|---|---|---|---|---|
| -      | 31           | 30       | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18      | 17       | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9      | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x000001FF   |          |    |    |    |    |    |    |    |    |    |    |    |         |          |    |    |    |    |    |    |    |        |   |   |   |   |   |   |   |   |   |
| Name   | COMEN        | Reserved |    |    |    |    |    |    |    |    |    |    |    | PLCHSEL | Reserved |    |    |    |    |    |    |    | TRGSEL |   |   |   |   |   |   |   |   |   |
| Access | RW           |          |    |    |    |    |    |    |    |    |    |    |    | RW      |          |    |    |    |    |    |    |    | RW     |   |   |   |   |   |   |   |   |   |

| Bit   | Name     | Function               | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | Access |
|-------|----------|------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31    | COMEN[1] | Common trigger enabled | <ul style="list-style-type: none"> <li>0b0: Disable triggering of this AOS target by the common trigger event of AOS_COMTRGSEL2</li> <li>0b1: Enable triggering of this AOS target by the common trigger event of AOS_COMTRGSEL2</li> </ul>                                                                                                                                                                                                                                               | RW     |
| 30    | COMEN[0] | Common trigger enabled | <ul style="list-style-type: none"> <li>0b0: Disable triggering of this AOS target by the common trigger event of AOS_COMTRGSEL1</li> <li>0b1: Enable triggering of this AOS target by the common trigger event of AOS_COMTRGSEL1</li> </ul>                                                                                                                                                                                                                                               | RW     |
| 29:19 | Reserved | Reserved               | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | -      |
| 18:16 | PLCHSEL  | PLU selection bit      | <ul style="list-style-type: none"> <li>0b000: Trigger this AOS target using the event selected by the TRGSEL[8:0] bits</li> <li>0b001: Trigger this AOS target using the event after PLU0 operation</li> <li>0b010: Trigger this AOS target using the event after PLU1 operation</li> <li>0b011: Trigger this AOS target using the event after PLU2 operation</li> <li>0b100: Trigger this AOS target using the event after PLU3 operation</li> <li>Others: Setting prohibited</li> </ul> | RW     |

| Bit  | Name            | Function                 | Description                                                     | Access |
|------|-----------------|--------------------------|-----------------------------------------------------------------|--------|
| 15:9 | <i>Reserved</i> | <i>Reserved</i>          | -                                                               | -      |
| 8:0  | TRGSEL          | Trigger source selection | Write the number of the peripheral circuit event to be selected | RW     |

### 12.4.12 TimerA Hardware Trigger Event Selection Register (TMRA\_TRGSELx) (x=0-3)

After selecting internal hardware trigger in TimerA, use this register to select the trigger event source. When PLCHSEL[2:0] is set to 0b000, write the trigger event number to the TRGSEL[8:0] bits in this register. When PLCHSEL[2:0] is set to 0b001, 0b010, 0b011, or 0b100, write the trigger event number to the corresponding PLU Trigger Source Selection Register (AOS\_PLU\_TRGSELA/B/C/D), and configure the AOS\_PLU\_CR Register accordingly. When the peripheral event corresponding to the number occurs, TimerA will be triggered to act (including start/stop/clear/input capture) by this event.

All TimerA units share these 4 registers.

Offset: 0x80+0x04\*x

| Offset | Bit Position |          |    |    |    |    |    |    |    |    |    |    |    |         |          |    |    |    |    |    |    |    |        |   |   |   |   |   |   |   |   |   |  |  |  |  |  |  |
|--------|--------------|----------|----|----|----|----|----|----|----|----|----|----|----|---------|----------|----|----|----|----|----|----|----|--------|---|---|---|---|---|---|---|---|---|--|--|--|--|--|--|
| -      | 31           | 30       | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18      | 17       | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9      | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |  |  |  |  |  |  |
| Reset  | 0x000001FF   |          |    |    |    |    |    |    |    |    |    |    |    |         |          |    |    |    |    |    |    |    |        |   |   |   |   |   |   |   |   |   |  |  |  |  |  |  |
| Name   | COMEN        | Reserved |    |    |    |    |    |    |    |    |    |    |    | PLCHSEL | Reserved |    |    |    |    |    |    |    | TRGSEL |   |   |   |   |   |   |   |   |   |  |  |  |  |  |  |
| Access | RW           |          |    |    |    |    |    |    |    |    |    |    |    | RW      |          |    |    |    |    |    |    |    | RW     |   |   |   |   |   |   |   |   |   |  |  |  |  |  |  |

| Bit   | Name     | Function                 | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | Access |
|-------|----------|--------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31    | COMEN[1] | Common trigger enabled   | <ul style="list-style-type: none"> <li>0b0: Disable triggering of this AOS target by the common trigger event of AOS_COMTRGSEL2</li> <li>0b1: Enable triggering of this AOS target by the common trigger event of AOS_COMTRGSEL2</li> </ul>                                                                                                                                                                                                                                               | RW     |
| 30    | COMEN[0] | Common trigger enabled   | <ul style="list-style-type: none"> <li>0b0: Disable triggering of this AOS target by the common trigger event of AOS_COMTRGSEL1</li> <li>0b1: Enable triggering of this AOS target by the common trigger event of AOS_COMTRGSEL1</li> </ul>                                                                                                                                                                                                                                               | RW     |
| 29:19 | Reserved | Reserved                 | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | -      |
| 18:16 | PLCHSEL  | PLU selection bit        | <ul style="list-style-type: none"> <li>0b000: Trigger this AOS target using the event selected by the TRGSEL[8:0] bits</li> <li>0b001: Trigger this AOS target using the event after PLU0 operation</li> <li>0b010: Trigger this AOS target using the event after PLU1 operation</li> <li>0b011: Trigger this AOS target using the event after PLU2 operation</li> <li>0b100: Trigger this AOS target using the event after PLU3 operation</li> <li>Others: Setting prohibited</li> </ul> | RW     |
| 15:9  | Reserved | Reserved                 | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | -      |
| 8:0   | TRGSEL   | Trigger source selection | Write the number of the peripheral circuit event to be selected                                                                                                                                                                                                                                                                                                                                                                                                                           | RW     |

### 12.4.13 OTS Hardware Trigger Event Selection Register (OTS\_TRGSEL)

After selecting the hardware trigger start mode in the OTS, use this register to select the trigger event source. When PLCHSEL[2:0] is set to 0b000, write the trigger event number to the TRGSEL[8:0] bits in this register. When PLCHSEL[2:0] is set to 0b001, 0b010, 0b011, or 0b100, write the trigger event number to the corresponding PLU Trigger Source Selection Register (AOS\_PLU\_TRGSELA/B/C/D), and configure the AOS\_PLU\_CR Register accordingly. When the peripheral event corresponding to the number occurs, OTS will be triggered to start by this event.

| Offset | Bit Position |          |    |    |    |    |    |    |    |    |    |    |    |         |          |    |    |    |    |    |    |    |        |   |   |   |   |   |   |   |   |   |  |  |  |  |  |  |
|--------|--------------|----------|----|----|----|----|----|----|----|----|----|----|----|---------|----------|----|----|----|----|----|----|----|--------|---|---|---|---|---|---|---|---|---|--|--|--|--|--|--|
| 0x90   | 31           | 30       | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18      | 17       | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9      | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |  |  |  |  |  |  |
| Reset  | 0x000001FF   |          |    |    |    |    |    |    |    |    |    |    |    |         |          |    |    |    |    |    |    |    |        |   |   |   |   |   |   |   |   |   |  |  |  |  |  |  |
| Name   | COMEN        | Reserved |    |    |    |    |    |    |    |    |    |    |    | PLCHSEL | Reserved |    |    |    |    |    |    |    | TRGSEL |   |   |   |   |   |   |   |   |   |  |  |  |  |  |  |
| Access | RW           |          |    |    |    |    |    |    |    |    |    |    |    | RW      |          |    |    |    |    |    |    |    | RW     |   |   |   |   |   |   |   |   |   |  |  |  |  |  |  |

| Bit   | Name     | Function                 | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | Access |
|-------|----------|--------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31    | COMEN[1] | Common trigger enabled   | <ul style="list-style-type: none"> <li>0b0: Disable triggering of this AOS target by the common trigger event of AOS_COMTRGSEL2</li> <li>0b1: Enable triggering of this AOS target by the common trigger event of AOS_COMTRGSEL2</li> </ul>                                                                                                                                                                                                                                               | RW     |
| 30    | COMEN[0] | Common trigger enabled   | <ul style="list-style-type: none"> <li>0b0: Disable triggering of this AOS target by the common trigger event of AOS_COMTRGSEL1</li> <li>0b1: Enable triggering of this AOS target by the common trigger event of AOS_COMTRGSEL1</li> </ul>                                                                                                                                                                                                                                               | RW     |
| 29:19 | Reserved | Reserved                 | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | -      |
| 18:16 | PLCHSEL  | PLU selection bit        | <ul style="list-style-type: none"> <li>0b000: Trigger this AOS target using the event selected by the TRGSEL[8:0] bits</li> <li>0b001: Trigger this AOS target using the event after PLU0 operation</li> <li>0b010: Trigger this AOS target using the event after PLU1 operation</li> <li>0b011: Trigger this AOS target using the event after PLU2 operation</li> <li>0b100: Trigger this AOS target using the event after PLU3 operation</li> <li>Others: Setting prohibited</li> </ul> | RW     |
| 15:9  | Reserved | Reserved                 | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | -      |
| 8:0   | TRGSEL   | Trigger source selection | Write the number of the peripheral circuit event to be selected                                                                                                                                                                                                                                                                                                                                                                                                                           | RW     |

### 12.4.14 ADC1 Conversion Start On-Chip Trigger Source Selection Register (ADC1\_TRGSELx) (x=0-1)

After selecting the on-chip trigger mode in ADC1, use this register to select the trigger event source. When PLCHSEL[2:0] is set to 0b000, write the trigger event number to the TRGSEL[8:0] bits in this register. When PLCHSEL[2:0] is set to 0b001, 0b010, 0b011, or 0b100, write the trigger event number to the corresponding PLU Trigger Source Selection Register (AOS\_PLU\_TRGSELA/B/C/D), and configure the AOS\_PLU\_CR Register accordingly. When the peripheral event corresponding to the number occurs, ADC1 will be triggered to start ADC conversion by this event.

Sequence A and sequence B of ADC1 share these 2 registers.

Offset: 0x94+0x04\*x

| Offset | Bit Position |          |    |    |    |    |    |    |    |    |    |    |    |         |          |    |    |    |    |    |    |    |        |   |   |   |   |   |   |   |   |   |
|--------|--------------|----------|----|----|----|----|----|----|----|----|----|----|----|---------|----------|----|----|----|----|----|----|----|--------|---|---|---|---|---|---|---|---|---|
| -      | 31           | 30       | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18      | 17       | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9      | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x000001FF   |          |    |    |    |    |    |    |    |    |    |    |    |         |          |    |    |    |    |    |    |    |        |   |   |   |   |   |   |   |   |   |
| Name   | COMEN        | Reserved |    |    |    |    |    |    |    |    |    |    |    | PLCHSEL | Reserved |    |    |    |    |    |    |    | TRGSEL |   |   |   |   |   |   |   |   |   |
| Access | RW           |          |    |    |    |    |    |    |    |    |    |    |    | RW      |          |    |    |    |    |    |    |    | RW     |   |   |   |   |   |   |   |   |   |

| Bit   | Name     | Function                 | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | Access |
|-------|----------|--------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31    | COMEN[1] | Common trigger enabled   | <ul style="list-style-type: none"> <li>0b0: Disable triggering of this AOS target by the common trigger event of AOS_COMTRGSEL2</li> <li>0b1: Enable triggering of this AOS target by the common trigger event of AOS_COMTRGSEL2</li> </ul>                                                                                                                                                                                                                                               | RW     |
| 30    | COMEN[0] | Common trigger enabled   | <ul style="list-style-type: none"> <li>0b0: Disable triggering of this AOS target by the common trigger event of AOS_COMTRGSEL1</li> <li>0b1: Enable triggering of this AOS target by the common trigger event of AOS_COMTRGSEL1</li> </ul>                                                                                                                                                                                                                                               | RW     |
| 29:19 | Reserved | Reserved                 | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | -      |
| 18:16 | PLCHSEL  | PLU selection bit        | <ul style="list-style-type: none"> <li>0b000: Trigger this AOS target using the event selected by the TRGSEL[8:0] bits</li> <li>0b001: Trigger this AOS target using the event after PLU0 operation</li> <li>0b010: Trigger this AOS target using the event after PLU1 operation</li> <li>0b011: Trigger this AOS target using the event after PLU2 operation</li> <li>0b100: Trigger this AOS target using the event after PLU3 operation</li> <li>Others: Setting prohibited</li> </ul> | RW     |
| 15:9  | Reserved | Reserved                 | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | -      |
| 8:0   | TRGSEL   | Trigger source selection | Write the number of the peripheral circuit event to be selected                                                                                                                                                                                                                                                                                                                                                                                                                           | RW     |

### 12.4.15 ADC2 Conversion Start On-Chip Trigger Source Selection Register (ADC2\_TRGSELx) (x=0-1)

After selecting the on-chip trigger mode in ADC2, use this register to select the trigger event source. When PLCHSEL[2:0] is set to 0b000, write the trigger event number to the TRGSEL[8:0] bits in this register. When PLCHSEL[2:0] is set to 0b001, 0b010, 0b011, or 0b100, write the trigger event number to the corresponding PLU Trigger Source Selection Register (AOS\_PLU\_TRGSELA/B/C/D), and configure the AOS\_PLU\_CR Register accordingly. When the peripheral event corresponding to the number occurs, ADC2 will be triggered to start ADC conversion by this event.

Sequence A and sequence B of ADC2 share these 2 registers.

Offset: 0x9C+0x04\*x

| Offset | Bit Position |          |    |    |    |    |    |    |    |    |    |    |    |         |          |    |    |    |    |    |    |    |        |   |   |   |   |   |   |   |   |   |
|--------|--------------|----------|----|----|----|----|----|----|----|----|----|----|----|---------|----------|----|----|----|----|----|----|----|--------|---|---|---|---|---|---|---|---|---|
| -      | 31           | 30       | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18      | 17       | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9      | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x000001FF   |          |    |    |    |    |    |    |    |    |    |    |    |         |          |    |    |    |    |    |    |    |        |   |   |   |   |   |   |   |   |   |
| Name   | COMEN        | Reserved |    |    |    |    |    |    |    |    |    |    |    | PLCHSEL | Reserved |    |    |    |    |    |    |    | TRGSEL |   |   |   |   |   |   |   |   |   |
| Access | RW           |          |    |    |    |    |    |    |    |    |    |    |    | RW      |          |    |    |    |    |    |    |    | RW     |   |   |   |   |   |   |   |   |   |

| Bit   | Name     | Function                 | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | Access |
|-------|----------|--------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31    | COMEN[1] | Common trigger enabled   | <ul style="list-style-type: none"> <li>0b0: Disable triggering of this AOS target by the common trigger event of AOS_COMTRGSEL2</li> <li>0b1: Enable triggering of this AOS target by the common trigger event of AOS_COMTRGSEL2</li> </ul>                                                                                                                                                                                                                                               | RW     |
| 30    | COMEN[0] | Common trigger enabled   | <ul style="list-style-type: none"> <li>0b0: Disable triggering of this AOS target by the common trigger event of AOS_COMTRGSEL1</li> <li>0b1: Enable triggering of this AOS target by the common trigger event of AOS_COMTRGSEL1</li> </ul>                                                                                                                                                                                                                                               | RW     |
| 29:19 | Reserved | Reserved                 | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | -      |
| 18:16 | PLCHSEL  | PLU selection bit        | <ul style="list-style-type: none"> <li>0b000: Trigger this AOS target using the event selected by the TRGSEL[8:0] bits</li> <li>0b001: Trigger this AOS target using the event after PLU0 operation</li> <li>0b010: Trigger this AOS target using the event after PLU1 operation</li> <li>0b011: Trigger this AOS target using the event after PLU2 operation</li> <li>0b100: Trigger this AOS target using the event after PLU3 operation</li> <li>Others: Setting prohibited</li> </ul> | RW     |
| 15:9  | Reserved | Reserved                 | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | -      |
| 8:0   | TRGSEL   | Trigger source selection | Write the number of the peripheral circuit event to be selected                                                                                                                                                                                                                                                                                                                                                                                                                           | RW     |

### 12.4.16 ADC3 Conversion Start On-Chip Trigger Source Selection Register (ADC3\_TRGSELx) (x=0-1)

After selecting the on-chip trigger mode in ADC3, use this register to select the trigger event source. When PLCHSEL[2:0] is set to 0b000, write the trigger event number to the TRGSEL[8:0] bits in this register. When PLCHSEL[2:0] is set to 0b001, 0b010, 0b011, or 0b100, write the trigger event number to the corresponding PLU Trigger Source Selection Register (AOS\_PLU\_TRGSELA/B/C/D), and configure the AOS\_PLU\_CR Register accordingly. When the peripheral event corresponding to the number occurs, ADC3 will be triggered to start ADC conversion by this event.

Sequence A and sequence B of ADC3 share these 2 registers.

Offset: 0xA4+0x04\*x

| Offset | Bit Position |          |    |    |    |    |    |    |    |    |    |    |    |         |          |    |    |    |    |    |    |    |        |   |   |   |   |   |   |   |   |   |
|--------|--------------|----------|----|----|----|----|----|----|----|----|----|----|----|---------|----------|----|----|----|----|----|----|----|--------|---|---|---|---|---|---|---|---|---|
| -      | 31           | 30       | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18      | 17       | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9      | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x000001FF   |          |    |    |    |    |    |    |    |    |    |    |    |         |          |    |    |    |    |    |    |    |        |   |   |   |   |   |   |   |   |   |
| Name   | COMEN        | Reserved |    |    |    |    |    |    |    |    |    |    |    | PLCHSEL | Reserved |    |    |    |    |    |    |    | TRGSEL |   |   |   |   |   |   |   |   |   |
| Access | RW           |          |    |    |    |    |    |    |    |    |    |    |    | RW      |          |    |    |    |    |    |    |    | RW     |   |   |   |   |   |   |   |   |   |

| Bit   | Name     | Function                 | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | Access |
|-------|----------|--------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31    | COMEN[1] | Common trigger enabled   | <ul style="list-style-type: none"> <li>0b0: Disable triggering of this AOS target by the common trigger event of AOS_COMTRGSEL2</li> <li>0b1: Enable triggering of this AOS target by the common trigger event of AOS_COMTRGSEL2</li> </ul>                                                                                                                                                                                                                                               | RW     |
| 30    | COMEN[0] | Common trigger enabled   | <ul style="list-style-type: none"> <li>0b0: Disable triggering of this AOS target by the common trigger event of AOS_COMTRGSEL1</li> <li>0b1: Enable triggering of this AOS target by the common trigger event of AOS_COMTRGSEL1</li> </ul>                                                                                                                                                                                                                                               | RW     |
| 29:19 | Reserved | Reserved                 | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | -      |
| 18:16 | PLCHSEL  | PLU selection bit        | <ul style="list-style-type: none"> <li>0b000: Trigger this AOS target using the event selected by the TRGSEL[8:0] bits</li> <li>0b001: Trigger this AOS target using the event after PLU0 operation</li> <li>0b010: Trigger this AOS target using the event after PLU1 operation</li> <li>0b011: Trigger this AOS target using the event after PLU2 operation</li> <li>0b100: Trigger this AOS target using the event after PLU3 operation</li> <li>Others: Setting prohibited</li> </ul> | RW     |
| 15:9  | Reserved | Reserved                 | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | -      |
| 8:0   | TRGSEL   | Trigger source selection | Write the number of the peripheral circuit event to be selected                                                                                                                                                                                                                                                                                                                                                                                                                           | RW     |

12.4.17 Common Trigger Source Selection Register (AOS\_COMTRGSELx) (x=1-2)

Write the trigger event number to AOS\_COMTRGSEL1/2. When the peripheral circuit event corresponding to the number occurs, if the COMEN bit in the dedicated trigger source selection register of one or more AOS targets is 1, the peripheral event corresponding to the number will trigger the start of this one or more AOS targets.

Offset: 0xAC (AOS\_COMTRGSEL1), 0xB0 (AOS\_COMTRGSEL2)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |        |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|--------|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| '      | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16     | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x000001FF   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |        |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    | COMTRG |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access | RW           |    |    |    |    |    |    |    |    |    |    |    |    |    |    |        |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit  | Name     | Function                 | Description                                                     | Access |
|------|----------|--------------------------|-----------------------------------------------------------------|--------|
| 31:9 | Reserved | Reserved                 | -                                                               | -      |
| 8:0  | COMTRG   | Trigger source selection | Write the number of the peripheral circuit event to be selected | RW     |



### 12.4.18 PLU Control Register (AOS\_PLUx\_CR) (x=0-3)

The AOS is equipped with a total of 4 PLUs, each with 4 inputs (PLINA, PLINB, PLINC, PLIND) and 1 output (PLOUT).

The inputs for logical operations can be AOS trigger sources or PORT inputs. Select the input source or fixed value through the PLINySEL (y=A, B, C, D) bit. Select the desired logical operation through the PLMODE bit.

Offset: 0x200+0x004\*x

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |          |          |          |          |          |    |   |   |   |   |        |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----------|----------|----------|----------|----------|----|---|---|---|---|--------|---|---|---|---|---|
| -      | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15       | 14       | 13       | 12       | 11       | 10 | 9 | 8 | 7 | 6 | 5      | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |          |          |          |          |          |    |   |   |   |   |        |   |   |   |   |   |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | PLINDSEL | PLINCSEL | PLINBSEL | PLINASEL | Reserved |    |   |   |   |   | PLMODE |   |   |   |   |   |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RW       | RW       | RW       | RW       |          |    |   |   |   |   | RW     |   |   |   |   |   |

| Bit   | Name     | Function                     | Description                                                                                                                                                                                                                                        | Access |
|-------|----------|------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:16 | Reserved | Reserved                     | -                                                                                                                                                                                                                                                  | -      |
| 15:14 | PLINDSEL | PLIND Input Source Selection | <ul style="list-style-type: none"> <li>0b00: PLIND is fixed at 0</li> <li>0b01: PLIND is fixed at 1</li> <li>0b10: PLIND is the trigger source selected by AOS_PLU_TRGSELD</li> <li>0b11: PLIND is the input value of the PLUx_IND port</li> </ul> | RW     |
| 13:12 | PLINCSEL | PLINC Input Source Selection | <ul style="list-style-type: none"> <li>0b00: PLINC is fixed at 0</li> <li>0b01: PLINC is fixed at 1</li> <li>0b10: PLINC is the trigger source selected by AOS_PLU_TRGSELC</li> <li>0b11: PLINC is the PLUx_INC port input value</li> </ul>        | RW     |
| 11:10 | PLINBSEL | PLINB Input Source Selection | <ul style="list-style-type: none"> <li>0b00: PLINB is fixed at 0</li> <li>0b01: PLINB is fixed at 1</li> <li>0b10: PLINB is the trigger source selected by AOS_PLU_TRGSELB</li> <li>0b11: PLINB is the input value of the PLUx_INB port</li> </ul> | RW     |
| 9:8   | PLINASEL | PLINA Input Source Selection | <ul style="list-style-type: none"> <li>0b00: PLINA is fixed at 0</li> <li>0b01: PLINA is fixed at 1</li> <li>0b10: PLINA is the trigger source selected by AOS_PLU_TRGSELA</li> <li>0b11: PLINA is the input value of the PLUx_INA port</li> </ul> | RW     |
| 7:2   | Reserved | Reserved                     | -                                                                                                                                                                                                                                                  | -      |

| Bit | Name   | Function                    | Description                                                                                                                                                                                                                                                                                                                                                                                                      | Access |
|-----|--------|-----------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 1:0 | PLMODE | Logical operation selection | <ul style="list-style-type: none"> <li>● 0b00: OR-OR, i.e., <math>(PLINA \mid PLINB) \mid (PLINC \mid PLIND)</math></li> <li>● 0b01: AND-OR, i.e., <math>(PLINA \&amp; PLINB) \mid (PLINC \&amp; PLIND)</math></li> <li>● 0b10: OR-XOR, i.e., <math>(PLINA \mid PLINB) \wedge (PLINC \mid PLIND)</math></li> <li>● 0b11: AND-AND, i.e., <math>(PLINA \&amp; PLINB) \&amp; (PLINC \&amp; PLIND)</math></li> </ul> | RW     |

**Notice**

When using the PLU operation result as the AOS trigger source, set PLINySEL (y=A, B, C, D) to 0b00 or 0b10, and PLMODE to 0b00.

### 12.4.19 PLU Trigger Source Selection Register A (AOS\_PLUX\_TRGSELA) (x=0-3)

When PLINA of PLU selects the trigger source input, write the number of the corresponding trigger event to AOS\_PLUX\_TRGSELA.

Offset: 0x210+0x010\*x

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |          |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|----------|---|---|---|---|---|---|---|---|
| '      | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8        | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x000001FF   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |          |   |   |   |   |   |   |   |   |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   | PLTRGSEL |   |   |   |   |   |   |   |   |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   | RW       |   |   |   |   |   |   |   |   |

| Bit  | Name     | Function                 | Description                                                     | Access |
|------|----------|--------------------------|-----------------------------------------------------------------|--------|
| 31:9 | Reserved | Reserved                 | -                                                               | -      |
| 8:0  | PLTRGSEL | Trigger source selection | Write the number of the peripheral circuit event to be selected | RW     |

### 12.4.20 PLU Trigger Source Selection Register B (AOS\_PLUX\_TRGSELB) (x=0-3)

When PLINB of PLU selects the trigger source input, write the number of the corresponding trigger event to AOS\_PLUX\_TRGSELB.

Offset: 0x214+0x010\*x

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |          |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|----------|---|---|---|---|---|---|---|---|
| '      | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8        | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x000001FF   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |          |   |   |   |   |   |   |   |   |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   | PLTRGSEL |   |   |   |   |   |   |   |   |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   | RW       |   |   |   |   |   |   |   |   |

| Bit  | Name     | Function                 | Description                                                     | Access |
|------|----------|--------------------------|-----------------------------------------------------------------|--------|
| 31:9 | Reserved | Reserved                 | -                                                               | -      |
| 8:0  | PLTRGSEL | Trigger source selection | Write the number of the peripheral circuit event to be selected | RW     |

### 12.4.21 PLU Trigger Source Selection Register C (AOS\_PLUX\_TRGSELC) (x=0-3)

When PLINC of PLU selects the trigger source input, write the number of the corresponding trigger event to AOS\_PLUX\_TRGSELC.

Offset: 0x218+0x010\*x

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |          |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|----------|---|---|---|---|---|---|---|
| '      | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7        | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x000001FF   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |          |   |   |   |   |   |   |   |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   | PLTRGSEL |   |   |   |   |   |   |   |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   | RW       |   |   |   |   |   |   |   |

| Bit  | Name     | Function                 | Description                                                     | Access |
|------|----------|--------------------------|-----------------------------------------------------------------|--------|
| 31:9 | Reserved | Reserved                 | -                                                               | -      |
| 8:0  | PLTRGSEL | Trigger source selection | Write the number of the peripheral circuit event to be selected | RW     |

### 12.4.22 PLU Trigger Source Selection Register D (AOS\_PLUX\_TRGSELD) (x=0-3)

When PLIND of PLU selects the trigger source input, write the number of the corresponding trigger event to AOS\_PLUX\_TRGSELD.

Offset: 0x21C+0x010\*x

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |          |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|----------|---|---|---|---|---|---|---|
| '      | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7        | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x000001FF   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |          |   |   |   |   |   |   |   |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   | PLTRGSEL |   |   |   |   |   |   |   |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   | RW       |   |   |   |   |   |   |   |

| Bit  | Name     | Function                 | Description                                                     | Access |
|------|----------|--------------------------|-----------------------------------------------------------------|--------|
| 31:9 | Reserved | Reserved                 | -                                                               | -      |
| 8:0  | PLTRGSEL | Trigger source selection | Write the number of the peripheral circuit event to be selected | RW     |

## 13 Memory Protection Unit (MPU)

### 13.1 Introduction

The MPU can provide protection to the memory, and can improve the security of the system by preventing unauthorized access.

This chip has built-in 1 MPU unit for CPU, 5 MPU units for host and 1 MPU unit for IP.

**Table 13-1 MPU Module Introduction**

| Modules |                         | Content                                                                                                         |
|---------|-------------------------|-----------------------------------------------------------------------------------------------------------------|
| ARM MPU |                         | Memory Protection Unit of the CPU<br>8 areas, see ARM MPU description for details                               |
| DMPU    | System DMA_1 MPU: SMPU1 | Memory protection unit of system DMA_1<br>16 areas, 8 areas dedicated to system DMA, 8 areas shared by all DMAs |
|         | System DMA_2 MPU: SMPU2 | Memory protection unit of system DMA_2<br>16 areas, 8 areas dedicated to system DMA, 8 areas shared by all DMAs |
|         | USBFS-DMA MPU: FMPU     | Storage Protection Unit of USBFS-DMA<br>8 regions, shared by all DMAs                                           |
|         | USBHS-DMA MPU: HMPU     | Storage Protection Unit of USBHS-DMA<br>8 regions, shared by all DMAs                                           |
|         | ETH-DMA MPU: EMPU       | Storage Protection Unit of ETH-DMA<br>8 regions, shared by all DMAs                                             |
| IPMPU   |                         | Access protection unit for system IP and security-related IP                                                    |

Among them, the ARM MPU provides the access control of the CPU to the entire 4G address space.

DMPU includes SMPU1/SMPU2/FMPU/HMPU/EMPU, which respectively provide read and write access rights control of system DMA\_1/system DMA\_2/USBFS-DMA/USBHS-DMA/ETH-DMA to the entire 4G address space. When accessing the prohibited space, the MPU action can be set to ignore/bus error/non-maskable interrupt/reset.

The IPMPU provides access control to system IP and security-related IP in non-privileged mode.

## 13.2 Functional Description

### 13.2.1 Locale Setting

The MPU manages the authority of the storage space in units of regions. Each area can independently set the base address and area size, the range of which can be set is 32Bytes-4GBytes, the size must be  $2^N$  Bytes ( $N=5-32$ ), and the corresponding lower  $N$  bits of the base address are 0.

The address space not covered by any region is called the background region.

### 13.2.2 Permission Setting

Each area, including the background area, can be independently set for each DMA to allow reading/prohibit reading and allow writing/prohibit writing. If address overlap occurs between different areas, the set prohibition takes precedence.

### 13.2.3 MPU Action Selection

When a prohibited access occurs, the access has no effect (read access reads 0, write access has no effect), and the corresponding action can be set, which can be set as:

- Ignore
- Bus error
- Nomaskable interrupt
- Reset

### 13.2.4 Start MPU

SMPU1/SMPU2/FMPU/HMPU/EMPU can be enabled independently.

It is recommended to enable the MPU after setting the area range/permission setting/action selection.

## 13.3 Operation Example

### 13.3.1 Only Enable Partial Space Access

**Example:** Set the permission of the background area to read/write disable, set area 0 to read/write enable, set area 1 to read enable and disable write, and the range of area 0 covers area 1.

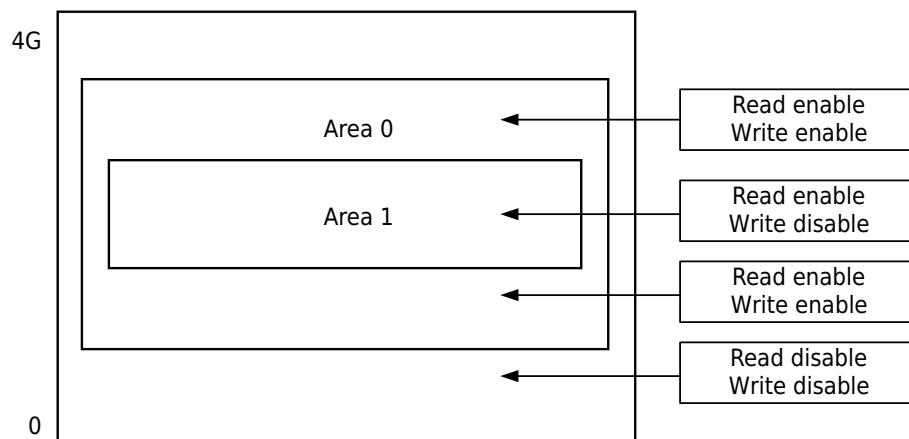


Figure 13-1 Schematic Diagram of Partial Memory Access Permissions

### 13.3.2 Only Disable Partial Space Access

**Example:** Set the permission of the background area to read enable/write disable, set area 0 to read/write enable, set area 1 to read enable/write disable, area 0 covers area 1, and set area 2 to read/write disable.

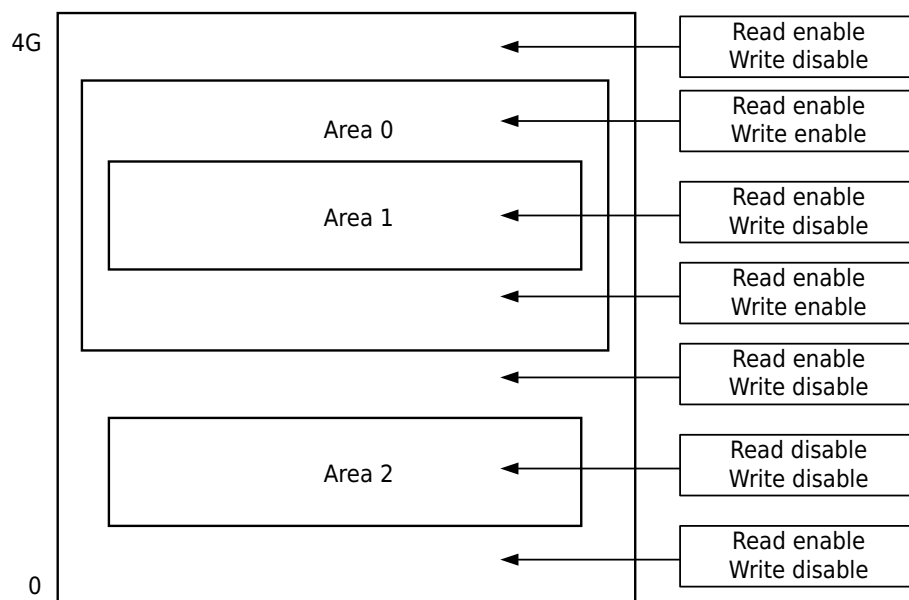


Figure 13-2 Schematic Diagram of only Prohibiting Access to Part of Space

## 13.4 Register Description

### 13.4.1 Register List

The registers of this module can only be set by the CPU.

Base address: 0x40050000

Table 13-2 Memory Protection Unit (MPU) Register Offset

| Offset      | Register   | Description                                           | Width |
|-------------|------------|-------------------------------------------------------|-------|
| 0x00+0x04*n | MPU_RGDn   | Area Region Description Register (n=0-15)<br>(1)      | 32    |
| 0x40        | MPU_SR     | Status Register                                       | 32    |
| 0x44        | MPU_ECLR   | Flag Clear Register                                   | 32    |
| 0x48        | MPU_WP     | Write Protection Register                             | 32    |
| 0x4C        | MPU_IPPR   | IP Access Protection Register <sup>(1)</sup>          | 32    |
| 0x50        | MPU_S1RGE  | SMPU1 Region Enable Register <sup>(1)</sup>           | 32    |
| 0x54        | MPU_S1RGWP | SMPU1 Region Write Permission Register <sup>(1)</sup> | 32    |
| 0x58        | MPU_S1RGRP | SMPU1 Region Read Permission Register <sup>(1)</sup>  | 32    |
| 0x5C        | MPU_S1CR   | SMPU1 Control Register <sup>(1)</sup>                 | 32    |
| 0x60        | MPU_S2RGE  | SMPU2 Region Enable Register <sup>(1)</sup>           | 32    |
| 0x64        | MPU_S2RGWP | SMPU2 Region Write Permission Register <sup>(1)</sup> | 32    |
| 0x68        | MPU_S2RGRP | SMPU2 Region Read Permission Register <sup>(1)</sup>  | 32    |
| 0x6C        | MPU_S2CR   | SMPU2 Control Register <sup>(1)</sup>                 | 32    |
| 0x70        | MPU_FRGE   | FMPU Region Enable Register <sup>(1)</sup>            | 32    |
| 0x74        | MPU_FRGWP  | FMPU Region Write Permission Register <sup>(1)</sup>  | 32    |
| 0x78        | MPU_FRGRP  | FMPU Region Read Permission Register <sup>(1)</sup>   | 32    |
| 0x7C        | MPU_FCR    | FMPU Control Register <sup>(1)</sup>                  | 32    |
| 0x80        | MPU_HRGE   | HMPU Region Enable Register <sup>(1)</sup>            | 32    |
| 0x84        | MPU_HRGWP  | HMPU Region Write Permission Register <sup>(1)</sup>  | 32    |
| 0x88        | MPU_HRGRP  | HMPU Region Read Permission Register <sup>(1)</sup>   | 32    |
| 0x8C        | MPU_HCR    | HMPU Control Register <sup>(1)</sup>                  | 32    |
| 0x90        | MPU_ERGE   | EMPU Region Enable Register <sup>(1)</sup>            | 32    |
| 0x94        | MPU_ERGWP  | EMPU Region Write Permission Register <sup>(1)</sup>  | 32    |
| 0x98        | MPU_ERGRP  | EMPU Region Read Permission Register <sup>(1)</sup>   | 32    |



| Offset | Register | Description                          | Width |
|--------|----------|--------------------------------------|-------|
| 0x9C   | MPU_ECR  | EMPU Control Register <sup>(1)</sup> | 32    |

**Note**

1. Writing is enabled when MPU\_WP.MPUWE is set to 1.

### 13.4.2 Area Region Description Register MPU\_RGDn (n=0-15)

Offset: 0x00+0x04\*n

| Offset | Bit Position     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |            |   |   |   |   |   |   |   |
|--------|------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|------------|---|---|---|---|---|---|---|
| -      | 31               | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7          | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000       |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |            |   |   |   |   |   |   |   |
| Name   | MPURGnADDR[31:5] |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   | MPURGnSIZE |   |   |   |   |   |   |   |
| Access | RW               |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   | RW         |   |   |   |   |   |   |   |

| Bit  | Name              | Function          | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | Access |
|------|-------------------|-------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:5 | MPURG-nADDR[31:5] | Area base address | Set the base address of area n, the active number of digits is related to the area size.<br>The total length of the area base address is 32 bits, of which MPURGnADDR[MPURGnSIZE:0] is fixed to 0.<br>For example:<br>When the region size is 32Bytes, the region base address is MPURGnADDR[31:5]+MPURGnADDR[4:0] (the lower 5 bits are fixed to 0).<br>When the region size is 64Bytes, the region base address is MPURGnADDR[31:6]+MPURGnADDR[5:0] (the lower 6 bits are fixed to 0). | RW     |
| 4:0  | MPURGn-SIZE       | Area size         | Set the size of the region n <ul style="list-style-type: none"> <li>● 0-3: Setting prohibited</li> <li>● 4: 32Bytes</li> <li>● 5: 64Bytes</li> <li>● .....</li> <li>● 30: 2GBytes</li> <li>● 31: 4GBytes</li> </ul>                                                                                                                                                                                                                                                                      | RW     |

## 13.4.3 Status Register (MPU\_SR)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |         |         |         |          |          |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|---------|---------|---------|----------|----------|
| 0x40   | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |         |         |         |          |          |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |         |         |         |          |          |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   | EMPUEAF | HMPUEAF | FMPUEAF | SMPU2EAF | SMPU1EAF |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   | RO      | RO      | RO      | RO       | RO       |

| Bit  | Name     | Function         | Description                                                                                                                                    | Access |
|------|----------|------------------|------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:5 | Reserved | Reserved         | -                                                                                                                                              | -      |
| 4    | EMPUEAF  | EMPU error flag  | <ul style="list-style-type: none"> <li>0b0: ETH-DMA has no error access</li> <li>0b1: ETH-DMA has a wrong access</li> </ul>                    | RO     |
| 3    | HMPUEAF  | HMPU error flag  | <ul style="list-style-type: none"> <li>0b0: USBHS-DMA has no error access</li> <li>0b1: USBHS-DMA has a wrong access</li> </ul>                | RO     |
| 2    | FMPUEAF  | FMPU error flag  | <ul style="list-style-type: none"> <li>0b0: USBFS-DMA has no error access</li> <li>0b1: USBFS-DMA has a wrong access</li> </ul>                | RO     |
| 1    | SMPU2EAF | SMPU2 error flag | <ul style="list-style-type: none"> <li>0b0: No access error occurred in system DMA_2</li> <li>0b1: System DMA_2 has an access error</li> </ul> | RO     |
| 0    | SMPU1EAF | SMPU1 error flag | <ul style="list-style-type: none"> <li>0b0: No access error occurred in system DMA_1</li> <li>0b1: System DMA_1 has an access error</li> </ul> | RO     |

**Notice**

The write operation to this register will be ignored, please use MPU\_ECLR to clear the error flag.

## 13.4.4 Flag Clear Register (MPU\_ECLR)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |          |          |          |           |           |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|----------|----------|----------|-----------|-----------|
| 0x44   | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |          |          |          |           |           |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |          |          |          |           |           |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   | EMPUECLR | HMPUECLR | FMPUECLR | SMPU2ECLR | SMPU1ECLR |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   | RW       | RW       | RW       | RW        | RW        |

| Bit  | Name      | Function                 | Description                                                                        | Access |
|------|-----------|--------------------------|------------------------------------------------------------------------------------|--------|
| 31:5 | Reserved  | Reserved                 | -                                                                                  | -      |
| 4    | EMPUECLR  | EMPU error flag cleared  | Writing 1 clears the error flag to 0, write 0 has no effect; read always returns 0 | RW     |
| 3    | HMPUECLR  | HMPU error flag cleared  | Writing 1 clears the error flag to 0, write 0 has no effect; read always returns 0 | RW     |
| 2    | FMPUECLR  | FMPU error flag cleared  | Writing 1 clears the error flag to 0, write 0 has no effect; read always returns 0 | RW     |
| 1    | SMPU2ECLR | SMPU2 error flag cleared | Writing 1 clears the error flag to 0, write 0 has no effect; read always returns 0 | RW     |
| 0    | SMPU1ECLR | SMPU1 error flag cleared | Writing 1 clears the error flag to 0, write 0 has no effect; read always returns 0 | RW     |

## 13.4.5 Write Protection Register (MPU\_WP)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |      |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |       |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|-------|
| 0x48   | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15   | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |       |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |      |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |       |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | WKEY |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   | MPUWE |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |      |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   | RW    |

| Bit   | Name     | Function                  | Description                                                                                                                                                                   | Access |
|-------|----------|---------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:16 | Reserved | Reserved                  | -                                                                                                                                                                             | -      |
| 15:1  | WKEY     | Write code                | Writing to MPUWE requires simultaneously writing 0x4B52 to the WKEY register                                                                                                  | WO     |
| 0     | MPUWE    | MPU register write enable | <ul style="list-style-type: none"> <li>0b0: MPU address registers/control registers write disable</li> <li>0b1: MPU address register/control register write enable</li> </ul> | RW     |

**Note**

Writing 0x96A5 to this register sets MPUWE to 1; writing 0x96A4 clears MPUWE to 0. It is prohibited to modify MPUWE by writing any other values.

## 13.4.6 IP Access Protection Register (MPU\_IPPR)

**Note**

1. In the privileged mode, the object IP can be read and written without being affected by this register.
2. When DMPU read/write protection is enabled, all registers except MPU\_IPPR are inaccessible for read or write operations.

| Offset | Bit Position |          |         |         |         |         |         |         |          |          |         |         |        |         |           |           |         |         |        |         |         |         |        |        |        |         |         |         |         |         |            |            |
|--------|--------------|----------|---------|---------|---------|---------|---------|---------|----------|----------|---------|---------|--------|---------|-----------|-----------|---------|---------|--------|---------|---------|---------|--------|--------|--------|---------|---------|---------|---------|---------|------------|------------|
| 0x4C   | 31           | 30       | 29      | 28      | 27      | 26      | 25      | 24      | 23       | 22       | 21      | 20      | 19     | 18      | 17        | 16        | 15      | 14      | 13     | 12      | 11      | 10      | 9      | 8      | 7      | 6       | 5       | 4       | 3       | 2       | 1          | 0          |
| Reset  | 0x00000000   |          |         |         |         |         |         |         |          |          |         |         |        |         |           |           |         |         |        |         |         |         |        |        |        |         |         |         |         |         |            |            |
| Name   | BUSERRE      | Reserved | MSTPWRP | MSTPRDP | SYSCWRP | SYSCRDP | INTCWRP | INTCRDP | SRAMCWRP | SRAMCRDP | DMPUWRP | DMPURDP | RTCWRP | RTCURDP | BKSRAMWRP | BKSRAMRDP | SWDTWRP | SWDTRDP | WDTWRP | WDTURDP | ERMUWRP | ERMURDP | EFMWRP | EFMRDP | CRCWRP | CRCURDP | TRNGWRP | TRNGRDP | HASHWRP | HASHRDP | ENCRYPTWRP | ENCRYPTRDP |
| Access | RW           |          | RW      | RW      | RW      | RW      | RW      | RW      | RW       | RW       | RW      | RW      | RW     | RW      | RW        | RW        | RW      | RW      | RW     | RW      | RW      | RW      | RW     | RW     | RW     | RW      | RW      | RW      | RW      | RW      | RW         | RW         |

| Bit | Name     | Function              | Description                                                                                                                                                                                                                                                        | Access |
|-----|----------|-----------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31  | BUSERRE  | Bus error enabled     | <ul style="list-style-type: none"> <li>0b0: Ignore access to protected IP</li> <li>0b1: When an access to the protected IP occurs, a bus error is returned</li> </ul>                                                                                              | RW     |
| 30  | Reserved | Reserved              | -                                                                                                                                                                                                                                                                  | -      |
| 29  | MSTPWRP  | MSTP write protection | <ul style="list-style-type: none"> <li>0b0: Enable write operations on PWC_FCG0/1/2/3 and PWC_FCG0PC</li> <li>0b1: Disable write operations on PWC_FCG0/1/2/3 and PWC_FCG0PC</li> </ul>                                                                            | RW     |
| 28  | MSTPRDP  | MSTP read protection  | <ul style="list-style-type: none"> <li>0b0: Enable read operations on PWC_FCG0/1/2/3 and PWC_FCG0PC</li> <li>0b1: Disable read operations on PWC_FCG0/1/2/3 and PWC_FCG0PC</li> </ul>                                                                              | RW     |
| 27  | SYSCWRP  | SYSC write protection | <ul style="list-style-type: none"> <li>0b0: Enable write operations on RMU/CMU/PWC</li> <li>0b1: Disable write operations on RMU/CMU/PWC</li> </ul> <div> <b>Note</b><br/> When BUSERRE is set to 1, write operations on RMU/PWC will not cause bus errors. </div> | RW     |
| 26  | SYSCRDP  | SYSC read protection  | <ul style="list-style-type: none"> <li>0b0: Enable read operations on RMU/CMU/PWC</li> <li>0b1: Disable read operations on RMU/CMU/PWC</li> </ul> <div> <b>Note</b><br/> When BUSERRE is set to 1, write operations on RMU/PWC will not cause bus errors. </div>   | RW     |

| Bit | Name      | Function                            | Description                                                                                                                                       | Access |
|-----|-----------|-------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 25  | INTCWRP   | INTC write protection               | <ul style="list-style-type: none"> <li>● 0b0: Enable write operations on INTC</li> <li>● 0b1: Disable write operations on INTC</li> </ul>         | RW     |
| 24  | INTCRDP   | INTC read protection                | <ul style="list-style-type: none"> <li>● 0b0: Enable read operations on INTC</li> <li>● 0b1: Disable read operations on INTC</li> </ul>           | RW     |
| 23  | SRAMCWRP  | SRAMC write protection              | <ul style="list-style-type: none"> <li>● 0b0: Enable write operations on SRAMC</li> <li>● 0b1: Disable write operations on SRAMC</li> </ul>       | RW     |
| 22  | SRAMCRDP  | SRAMC read protection               | <ul style="list-style-type: none"> <li>● 0b0: Enable read operations on SRAMC</li> <li>● 0b1: Disable read operations on SRAMC</li> </ul>         | RW     |
| 21  | DMPUWRP   | DMPU write protection <sup>*2</sup> | <ul style="list-style-type: none"> <li>● 0b0: Enable write operations on DMPU</li> <li>● 0b1: Disable write operations on DMPU</li> </ul>         | RW     |
| 20  | DMPURDP   | DMPU read protection <sup>*2</sup>  | <ul style="list-style-type: none"> <li>● 0b0: Enable read operations on DMPU</li> <li>● 0b1: Disable read operations on DMPU</li> </ul>           | RW     |
| 19  | RTCWRP    | RTC write protection                | <ul style="list-style-type: none"> <li>● 0b0: Enable write operations on RTC</li> <li>● 0b1: Disable write operations on RTC</li> </ul>           | RW     |
| 18  | RTCRDP    | RTC read protection                 | <ul style="list-style-type: none"> <li>● 0b0: Enable read operations on RTC</li> <li>● 0b1: Disable read operations on RTC</li> </ul>             | RW     |
| 17  | BKSRAMWRP | BKSRAM write protection             | <ul style="list-style-type: none"> <li>● 0b0: Enable write operations on Ret SRAM</li> <li>● 0b1: Disable write operations on Ret SRAM</li> </ul> | RW     |
| 16  | BKSRAMRDP | BKSRAM read protection              | <ul style="list-style-type: none"> <li>● 0b0: Enable read operations on Ret SRAM</li> <li>● 0b1: Disable read operations on Ret SRAM</li> </ul>   | RW     |
| 15  | SWDTWRP   | SWDT write protection               | <ul style="list-style-type: none"> <li>● 0b0: Enable write operations on SWDT</li> <li>● 0b1: Disable write operations on SWDT</li> </ul>         | RW     |
| 14  | SWDTRDP   | SWDT read protection                | <ul style="list-style-type: none"> <li>● 0b0: Enable read operations on SWDT</li> <li>● 0b1: Disable read operations on SWDT</li> </ul>           | RW     |
| 13  | WDTWRP    | WDT write protection                | <ul style="list-style-type: none"> <li>● 0b0: Enable write operations on WDT</li> <li>● 0b1: Disable write operations on WDT</li> </ul>           | RW     |
| 12  | WDTRDP    | WDT read protection                 | <ul style="list-style-type: none"> <li>● 0b0: Enable read operations on WDT</li> <li>● 0b1: Disable read operations on WDT</li> </ul>             | RW     |
| 11  | ERMUWRP   | ERMU write protection               | <ul style="list-style-type: none"> <li>● 0b0: Enable write operations on ERMU</li> <li>● 0b1: Disable write operations on ERMU</li> </ul>         | RW     |
| 10  | ERMURDP   | ERMU read protection                | <ul style="list-style-type: none"> <li>● 0b0: Enable read operations on ERMU</li> <li>● 0b1: Disable read operations on ERMU</li> </ul>           | RW     |
| 9   | EFMWRP    | EFM write protection                | <ul style="list-style-type: none"> <li>● 0b0: Enable write operations on EFM</li> <li>● 0b1: Disable write operations on EFM</li> </ul>           | RW     |
| 8   | EFMRDP    | EFM read protection                 | <ul style="list-style-type: none"> <li>● 0b0: Enable read operations on EFM</li> <li>● 0b1: Disable read operations on EFM</li> </ul>             | RW     |
| 7   | CRCWRP    | CRC write protection                | <ul style="list-style-type: none"> <li>● 0b0: Enable write operations on CRC</li> <li>● 0b1: Disable write operations on CRC</li> </ul>           | RW     |

| Bit | Name       | Function                 | Description                                                                                                                                     | Access |
|-----|------------|--------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 6   | CRCRDP     | CRC read protection      | <ul style="list-style-type: none"> <li>● 0b0: Enable read operations on CRC</li> <li>● 0b1: Disable read operations on CRC</li> </ul>           | RW     |
| 5   | TRNGWRP    | TRNG write protection    | <ul style="list-style-type: none"> <li>● 0b0: Enable write operations on TRNG</li> <li>● 0b1: Disable write operations on TRNG</li> </ul>       | RW     |
| 4   | TRNGRDP    | TRNG read protection     | <ul style="list-style-type: none"> <li>● 0b0: Enable read operations on TRNG</li> <li>● 0b1: Disable read operations on TRNG</li> </ul>         | RW     |
| 3   | HASHWRP    | HASH write protection    | <ul style="list-style-type: none"> <li>● 0b0: Enable write operations on HASH</li> <li>● 0b1: Disable write operations on HASH</li> </ul>       | RW     |
| 2   | HASHRDP    | HASH read protection     | <ul style="list-style-type: none"> <li>● 0b0: Enable read operations on HASH</li> <li>● 0b1: Disable read operations on HASH</li> </ul>         | RW     |
| 1   | ENCRYPTWRP | SKE/PKE write protection | <ul style="list-style-type: none"> <li>● 0b0: Enable write operations on SKE/PKE</li> <li>● 0b1: Disable write operations on SKE/PKE</li> </ul> | RW     |
| 0   | ENCRYPTRDP | SKE/PKE read protection  | <ul style="list-style-type: none"> <li>● 0b0: Enable read operations on SKE/PKE</li> <li>● 0b1: Disable read operations on SKE/PKE</li> </ul>   | RW     |



## 13.4.7 SMPU1 Region Enable Register (MPU\_S1RGE)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |         |         |         |         |         |         |        |        |        |        |        |        |        |        |        |        |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---------|---------|---------|---------|---------|---------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|
| 0x50   | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15      | 14      | 13      | 12      | 11      | 10      | 9      | 8      | 7      | 6      | 5      | 4      | 3      | 2      | 1      | 0      |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |         |         |         |         |         |         |        |        |        |        |        |        |        |        |        |        |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | S1RG15E | S1RG14E | S1RG13E | S1RG12E | S1RG11E | S1RG10E | S1RG9E | S1RG8E | S1RG7E | S1RG6E | S1RG5E | S1RG4E | S1RG3E | S1RG2E | S1RG1E | S1RG0E |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RW      | RW      | RW      | RW      | RW      | RW      | RW     | RW     | RW     | RW     | RW     | RW     | RW     | RW     | RW     | RW     |

| Bit   | Name     | Function              | Description                                                                                                       | Access |
|-------|----------|-----------------------|-------------------------------------------------------------------------------------------------------------------|--------|
| 31:16 | Reserved | Reserved              | -                                                                                                                 | -      |
| 15    | S1RG15E  | SMPU1 zone 15 enabled | <ul style="list-style-type: none"> <li>0b0: SMPU1 zone 15 disabled</li> <li>0b1: SMPU1 zone 15 enabled</li> </ul> | RW     |
| 14    | S1RG14E  | SMPU1 zone 14 enabled | <ul style="list-style-type: none"> <li>0b0: SMPU1 zone 14 disabled</li> <li>0b1: SMPU1 zone 14 enabled</li> </ul> | RW     |
| 13    | S1RG13E  | SMPU1 zone 13 enabled | <ul style="list-style-type: none"> <li>0b0: SMPU1 zone 13 disabled</li> <li>0b1: SMPU1 zone 13 enabled</li> </ul> | RW     |
| 12    | S1RG12E  | SMPU1 zone 12 enabled | <ul style="list-style-type: none"> <li>0b0: SMPU1 zone 12 disabled</li> <li>0b1: SMPU1 zone 12 enabled</li> </ul> | RW     |
| 11    | S1RG11E  | SMPU1 zone 11 enabled | <ul style="list-style-type: none"> <li>0b0: SMPU1 zone 11 disabled</li> <li>0b1: SMPU1 zone 11 enabled</li> </ul> | RW     |
| 10    | S1RG10E  | SMPU1 zone 10 enabled | <ul style="list-style-type: none"> <li>0b0: SMPU1 zone 10 disabled</li> <li>0b1: SMPU1 zone 10 enabled</li> </ul> | RW     |
| 9     | S1RG9E   | SMPU1 zone 9 enabled  | <ul style="list-style-type: none"> <li>0b0: SMPU1 zone 9 disabled</li> <li>0b1: SMPU1 zone 9 enabled</li> </ul>   | RW     |
| 8     | S1RG8E   | SMPU1 zone 8 enabled  | <ul style="list-style-type: none"> <li>0b0: SMPU1 zone 8 disabled</li> <li>0b1: SMPU1 zone 8 enabled</li> </ul>   | RW     |
| 7     | S1RG7E   | SMPU1 zone 7 enabled  | <ul style="list-style-type: none"> <li>0b0: SMPU1 zone 7 disabled</li> <li>0b1: SMPU1 zone 7 enabled</li> </ul>   | RW     |
| 6     | S1RG6E   | SMPU1 zone 6 enabled  | <ul style="list-style-type: none"> <li>0b0: SMPU1 zone 6 disabled</li> <li>0b1: SMPU1 zone 6 enabled</li> </ul>   | RW     |
| 5     | S1RG5E   | SMPU1 zone 5 enabled  | <ul style="list-style-type: none"> <li>0b0: SMPU1 zone 5 disabled</li> <li>0b1: SMPU1 zone 5 enabled</li> </ul>   | RW     |
| 4     | S1RG4E   | SMPU1 zone 4 enabled  | <ul style="list-style-type: none"> <li>0b0: SMPU1 zone 4 disabled</li> <li>0b1: SMPU1 zone 4 enabled</li> </ul>   | RW     |
| 3     | S1RG3E   | SMPU1 zone 3 enabled  | <ul style="list-style-type: none"> <li>0b0: SMPU1 zone 3 disabled</li> <li>0b1: SMPU1 zone 3 enabled</li> </ul>   | RW     |

| Bit | Name   | Function             | Description                                                                                                      | Access |
|-----|--------|----------------------|------------------------------------------------------------------------------------------------------------------|--------|
| 2   | S1RG2E | SMPU1 zone 2 enabled | <ul style="list-style-type: none"><li>● 0b0: SMPU1 zone 2 disabled</li><li>● 0b1: SMPU1 zone 2 enabled</li></ul> | RW     |
| 1   | S1RG1E | SMPU1 zone 1 enabled | <ul style="list-style-type: none"><li>● 0b0: SMPU1 zone 1 disabled</li><li>● 0b1: SMPU1 zone 1 enabled</li></ul> | RW     |
| 0   | S1RG0E | SMPU1 zone 0 enabled | <ul style="list-style-type: none"><li>● 0b0: SMPU1 zone 0 disabled</li><li>● 0b1: SMPU1 zone 0 enabled</li></ul> | RW     |

## 13.4.8 SMPU1 Region Write Permission Register (MPU\_S1RGWP)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |          |          |          |          |          |          |         |         |         |         |         |         |         |         |         |         |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----------|----------|----------|----------|----------|----------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|
| 0x54   | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15       | 14       | 13       | 12       | 11       | 10       | 9       | 8       | 7       | 6       | 5       | 4       | 3       | 2       | 1       | 0       |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |          |          |          |          |          |          |         |         |         |         |         |         |         |         |         |         |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | S1RG15WP | S1RG14WP | S1RG13WP | S1RG12WP | S1RG11WP | S1RG10WP | S1RG9WP | S1RG8WP | S1RG7WP | S1RG6WP | S1RG5WP | S1RG4WP | S1RG3WP | S1RG2WP | S1RG1WP | S1RG0WP |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RW       | RW       | RW       | RW       | RW       | RW       | RW      | RW      | RW      | RW      | RW      | RW      | RW      | RW      | RW      | RW      |

| Bit   | Name     | Function                       | Description                                                                                                                                                 | Access |
|-------|----------|--------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:16 | Reserved | Reserved                       | -                                                                                                                                                           | -      |
| 15    | S1RG15WP | SMPU1 zone 15 write permission | <ul style="list-style-type: none"> <li>0b0: SMPU1 zone 15 enable system DMA_1 to write</li> <li>0b1: SMPU1 zone 15 disable system DMA_1 to write</li> </ul> | RW     |
| 14    | S1RG14WP | SMPU1 zone 14 write permission | <ul style="list-style-type: none"> <li>0b0: SMPU1 zone 14 enable system DMA_1 to write</li> <li>0b1: SMPU1 zone 14 disable system DMA_1 to write</li> </ul> | RW     |
| 13    | S1RG13WP | SMPU1 zone 13 write permission | <ul style="list-style-type: none"> <li>0b0: SMPU1 zone 13 enable system DMA_1 to write</li> <li>0b1: SMPU1 zone 13 disable system DMA_1 to write</li> </ul> | RW     |
| 12    | S1RG12WP | SMPU1 zone 12 write permission | <ul style="list-style-type: none"> <li>0b0: SMPU1 zone 12 enable system DMA_1 to write</li> <li>0b1: SMPU1 zone 12 disable system DMA_1 to write</li> </ul> | RW     |
| 11    | S1RG11WP | SMPU1 zone 11 write permission | <ul style="list-style-type: none"> <li>0b0: SMPU1 zone 11 enable system DMA_1 to write</li> <li>0b1: SMPU1 zone 11 disable system DMA_1 to write</li> </ul> | RW     |
| 10    | S1RG10WP | SMPU1 zone 10 write permission | <ul style="list-style-type: none"> <li>0b0: SMPU1 zone 10 enable system DMA_1 to write</li> <li>0b1: SMPU1 zone 10 disable system DMA_1 to write</li> </ul> | RW     |
| 9     | S1RG9WP  | SMPU1 zone 9 write permission  | <ul style="list-style-type: none"> <li>0b0: SMPU1 zone 9 enable system DMA_1 to write</li> <li>0b1: SMPU1 zone 9 disable system DMA_1 to write</li> </ul>   | RW     |
| 8     | S1RG8WP  | SMPU1 zone 8 write permission  | <ul style="list-style-type: none"> <li>0b0: SMPU1 zone 8 enable system DMA_1 to write</li> <li>0b1: SMPU1 zone 8 disable system DMA_1 to write</li> </ul>   | RW     |

| Bit | Name    | Function                      | Description                                                                                                                                                   | Access |
|-----|---------|-------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 7   | S1RG7WP | SMPU1 zone 7 write permission | <ul style="list-style-type: none"> <li>● 0b0: SMPU1 zone 7 enable system DMA_1 to write</li> <li>● 0b1: SMPU1 zone 7 disable system DMA_1 to write</li> </ul> | RW     |
| 6   | S1RG6WP | SMPU1 zone 6 write permission | <ul style="list-style-type: none"> <li>● 0b0: SMPU1 zone 6 enable system DMA_1 to write</li> <li>● 0b1: SMPU1 zone 6 disable system DMA_1 to write</li> </ul> | RW     |
| 5   | S1RG5WP | SMPU1 zone 5 write permission | <ul style="list-style-type: none"> <li>● 0b0: SMPU1 zone 5 enable system DMA_1 to write</li> <li>● 0b1: SMPU1 zone 5 disable system DMA_1 to write</li> </ul> | RW     |
| 4   | S1RG4WP | SMPU1 zone 4 write permission | <ul style="list-style-type: none"> <li>● 0b0: SMPU1 zone 4 enable system DMA_1 to write</li> <li>● 0b1: SMPU1 zone 4 disable system DMA_1 to write</li> </ul> | RW     |
| 3   | S1RG3WP | SMPU1 zone 3 write permission | <ul style="list-style-type: none"> <li>● 0b0: SMPU1 zone 3 enable system DMA_1 to write</li> <li>● 0b1: SMPU1 zone 3 disable system DMA_1 to write</li> </ul> | RW     |
| 2   | S1RG2WP | SMPU1 zone 2 write permission | <ul style="list-style-type: none"> <li>● 0b0: SMPU1 zone 2 enable system DMA_1 to write</li> <li>● 0b1: SMPU1 zone 2 disable system DMA_1 to write</li> </ul> | RW     |
| 1   | S1RG1WP | SMPU1 zone 1 write permission | <ul style="list-style-type: none"> <li>● 0b0: SMPU1 zone 1 enable system DMA_1 to write</li> <li>● 0b1: SMPU1 zone 1 disable system DMA_1 to write</li> </ul> | RW     |
| 0   | S1RG0WP | SMPU1 zone 0 write permission | <ul style="list-style-type: none"> <li>● 0b0: SMPU1 zone 0 enable system DMA_1 to write</li> <li>● 0b1: SMPU1 zone 0 disable system DMA_1 to write</li> </ul> | RW     |

## 13.4.9 SMPU1 Region Read Permission Register (MPU\_S1RGRP)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |          |          |          |          |          |          |         |         |         |         |         |         |         |         |         |         |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----------|----------|----------|----------|----------|----------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|
| 0x58   | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15       | 14       | 13       | 12       | 11       | 10       | 9       | 8       | 7       | 6       | 5       | 4       | 3       | 2       | 1       | 0       |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |          |          |          |          |          |          |         |         |         |         |         |         |         |         |         |         |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | S1RG15RP | S1RG14RP | S1RG13RP | S1RG12RP | S1RG11RP | S1RG10RP | S1RG9RP | S1RG8RP | S1RG7RP | S1RG6RP | S1RG5RP | S1RG4RP | S1RG3RP | S1RG2RP | S1RG1RP | S1RG0RP |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RW       | RW       | RW       | RW       | RW       | RW       | RW      | RW      | RW      | RW      | RW      | RW      | RW      | RW      | RW      | RW      |

| Bit   | Name     | Function                      | Description                                                                                                                                               | Access |
|-------|----------|-------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:16 | Reserved | Reserved                      | -                                                                                                                                                         | -      |
| 15    | S1RG15RP | SMPU1 zone 15 read permission | <ul style="list-style-type: none"> <li>0b0: SMPU1 zone 15 enable system DMA_1 to read</li> <li>0b1: SMPU1 zone 15 disable system DMA_1 to read</li> </ul> | RW     |
| 14    | S1RG14RP | SMPU1 zone 14 read permission | <ul style="list-style-type: none"> <li>0b0: SMPU1 zone 14 enable system DMA_1 to read</li> <li>0b1: SMPU1 zone 14 disable system DMA_1 to read</li> </ul> | RW     |
| 13    | S1RG13RP | SMPU1 zone 13 read permission | <ul style="list-style-type: none"> <li>0b0: SMPU1 zone 13 enable system DMA_1 to read</li> <li>0b1: SMPU1 zone 13 disable system DMA_1 to read</li> </ul> | RW     |
| 12    | S1RG12RP | SMPU1 zone 12 read permission | <ul style="list-style-type: none"> <li>0b0: SMPU1 zone 12 enable system DMA_1 to read</li> <li>0b1: SMPU1 zone 12 disable system DMA_1 to read</li> </ul> | RW     |
| 11    | S1RG11RP | SMPU1 zone 11 read permission | <ul style="list-style-type: none"> <li>0b0: SMPU1 zone 11 enable system DMA_1 to read</li> <li>0b1: SMPU1 zone 11 disable system DMA_1 to read</li> </ul> | RW     |
| 10    | S1RG10RP | SMPU1 zone 10 read permission | <ul style="list-style-type: none"> <li>0b0: SMPU1 zone 10 enable system DMA_1 to read</li> <li>0b1: SMPU1 zone 10 disable system DMA_1 to read</li> </ul> | RW     |
| 9     | S1RG9RP  | SMPU1 zone 9 read permission  | <ul style="list-style-type: none"> <li>0b0: SMPU1 zone 9 enable system DMA_1 to read</li> <li>0b1: SMPU1 zone 9 disable system DMA_1 to read</li> </ul>   | RW     |
| 8     | S1RG8RP  | SMPU1 zone 8 read permission  | <ul style="list-style-type: none"> <li>0b0: SMPU1 zone 8 enable system DMA_1 to read</li> <li>0b1: SMPU1 zone 8 disable system DMA_1 to read</li> </ul>   | RW     |

| Bit | Name    | Function                     | Description                                                                                                                                                 | Access |
|-----|---------|------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 7   | S1RG7RP | SMPU1 zone 7 read permission | <ul style="list-style-type: none"> <li>● 0b0: SMPU1 zone 7 enable system DMA_1 to read</li> <li>● 0b1: SMPU1 zone 7 disable system DMA_1 to read</li> </ul> | RW     |
| 6   | S1RG6RP | SMPU1 zone 6 read permission | <ul style="list-style-type: none"> <li>● 0b0: SMPU1 zone 6 enable system DMA_1 to read</li> <li>● 0b1: SMPU1 zone 6 disable system DMA_1 to read</li> </ul> | RW     |
| 5   | S1RG5RP | SMPU1 zone 5 read permission | <ul style="list-style-type: none"> <li>● 0b0: SMPU1 zone 5 enable system DMA_1 to read</li> <li>● 0b1: SMPU1 zone 5 disable system DMA_1 to read</li> </ul> | RW     |
| 4   | S1RG4RP | SMPU1 zone 4 read permission | <ul style="list-style-type: none"> <li>● 0b0: SMPU1 zone 4 enable system DMA_1 to read</li> <li>● 0b1: SMPU1 zone 4 disable system DMA_1 to read</li> </ul> | RW     |
| 3   | S1RG3RP | SMPU1 zone 3 read permission | <ul style="list-style-type: none"> <li>● 0b0: SMPU1 zone 3 enable system DMA_1 to read</li> <li>● 0b1: SMPU1 zone 3 disable system DMA_1 to read</li> </ul> | RW     |
| 2   | S1RG2RP | SMPU1 zone 2 read permission | <ul style="list-style-type: none"> <li>● 0b0: SMPU1 zone 2 enable system DMA_1 to read</li> <li>● 0b1: SMPU1 zone 2 disable system DMA_1 to read</li> </ul> | RW     |
| 1   | S1RG1RP | SMPU1 zone 1 read permission | <ul style="list-style-type: none"> <li>● 0b0: SMPU1 zone 1 enable system DMA_1 to read</li> <li>● 0b1: SMPU1 zone 1 disable system DMA_1 to read</li> </ul> | RW     |
| 0   | S1RG0RP | SMPU1 zone 0 read permission | <ul style="list-style-type: none"> <li>● 0b0: SMPU1 zone 0 enable system DMA_1 to read</li> <li>● 0b1: SMPU1 zone 0 disable system DMA_1 to read</li> </ul> | RW     |

## 13.4.10 SMPU1 Control Register (MPU\_S1CR)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |        |   |          |   |          |   |          |   |          |  |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|--------|---|----------|---|----------|---|----------|---|----------|--|
| 0x5C   | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7      | 6 | 5        | 4 | 3        | 2 | 1        | 0 |          |  |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |        |   |          |   |          |   |          |   |          |  |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   | SMPUIE |   | Reserved |   | SMPUIACT |   | SMPUIBWP |   | SMPUIBRP |  |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   | RW     |   |          |   | RW       |   | RW       |   | RW       |  |

| Bit  | Name     | Function                          | Description                                                                                                                                                                                                                                                                                                                                                | Access |
|------|----------|-----------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:8 | Reserved | Reserved                          | -                                                                                                                                                                                                                                                                                                                                                          | -      |
| 7    | SMPU1E   | SMPU1 enable                      | <ul style="list-style-type: none"> <li>0b0: SMPU1 disabled</li> <li>0b1: SMPU1 enabled</li> </ul>                                                                                                                                                                                                                                                          | RW     |
| 6:4  | Reserved | Reserved                          | -                                                                                                                                                                                                                                                                                                                                                          | -      |
| 3:2  | SMPU1ACT | SMPU1 action selection            | Set the action when the system DMA_1 is disabled from accessing <ul style="list-style-type: none"> <li>0b00: Read access reads 0, write access ignore</li> <li>0b01: Read access reads 0, write access ignore; bus error generate</li> <li>0b10: Read access reads 0, write access ignore; non-maskable interrupt generate</li> <li>0b11: Reset</li> </ul> | RW     |
| 1    | SMPU1BWP | SMPU1 background write permission | <ul style="list-style-type: none"> <li>0b0: Enable system DMA_1 write access in SMPU1 background space</li> <li>0b1: Disable system DMA_1 write access in SMPU1 background space</li> </ul>                                                                                                                                                                | RW     |
| 0    | SMPU1BRP | SMPU1 background read permission  | <ul style="list-style-type: none"> <li>0b0: Enable system DMA_1 read access in SMPU1 background space</li> <li>0b1: Disable system DMA_1 read access in SMPU1 background space</li> </ul>                                                                                                                                                                  | RW     |

**Note**

When multiple area settings overlap, the priority is: Setting prohibited > setting enabled.

## 13.4.11 SMPU2 Region Enable Register (MPU\_S2RGE)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |         |         |         |         |         |         |        |        |        |        |        |        |        |        |        |        |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---------|---------|---------|---------|---------|---------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|
| 0x60   | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15      | 14      | 13      | 12      | 11      | 10      | 9      | 8      | 7      | 6      | 5      | 4      | 3      | 2      | 1      | 0      |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |         |         |         |         |         |         |        |        |        |        |        |        |        |        |        |        |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | S2RG15E | S2RG14E | S2RG13E | S2RG12E | S2RG11E | S2RG10E | S2RG9E | S2RG8E | S2RG7E | S2RG6E | S2RG5E | S2RG4E | S2RG3E | S2RG2E | S2RG1E | S2RG0E |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RW      | RW      | RW      | RW      | RW      | RW      | RW     | RW     | RW     | RW     | RW     | RW     | RW     | RW     | RW     | RW     |

| Bit   | Name     | Function              | Description                                                                                                       | Access |
|-------|----------|-----------------------|-------------------------------------------------------------------------------------------------------------------|--------|
| 31:16 | Reserved | Reserved              | -                                                                                                                 | -      |
| 15    | S2RG15E  | SMPU2 zone 15 enabled | <ul style="list-style-type: none"> <li>0b0: SMPU2 zone 15 disabled</li> <li>0b1: SMPU2 zone 15 enabled</li> </ul> | RW     |
| 14    | S2RG14E  | SMPU2 zone 14 enabled | <ul style="list-style-type: none"> <li>0b0: SMPU2 zone 14 disabled</li> <li>0b1: SMPU2 zone 14 enabled</li> </ul> | RW     |
| 13    | S2RG13E  | SMPU2 zone 13 enabled | <ul style="list-style-type: none"> <li>0b0: SMPU2 zone 13 disabled</li> <li>0b1: SMPU2 zone 13 enabled</li> </ul> | RW     |
| 12    | S2RG12E  | SMPU2 zone 12 enabled | <ul style="list-style-type: none"> <li>0b0: SMPU2 zone 12 disabled</li> <li>0b1: SMPU2 zone 12 enabled</li> </ul> | RW     |
| 11    | S2RG11E  | SMPU2 zone 11 enabled | <ul style="list-style-type: none"> <li>0b0: SMPU2 zone 11 disabled</li> <li>0b1: SMPU2 zone 11 enabled</li> </ul> | RW     |
| 10    | S2RG10E  | SMPU2 zone 10 enabled | <ul style="list-style-type: none"> <li>0b0: SMPU2 zone 10 disabled</li> <li>0b1: SMPU2 zone 10 enabled</li> </ul> | RW     |
| 9     | S2RG9E   | SMPU2 zone 9 enabled  | <ul style="list-style-type: none"> <li>0b0: SMPU2 zone 9 disabled</li> <li>0b1: SMPU2 zone 9 enabled</li> </ul>   | RW     |
| 8     | S2RG8E   | SMPU2 zone 8 enabled  | <ul style="list-style-type: none"> <li>0b0: SMPU2 zone 8 disabled</li> <li>0b1: SMPU2 zone 8 enabled</li> </ul>   | RW     |
| 7     | S2RG7E   | SMPU2 zone 7 enabled  | <ul style="list-style-type: none"> <li>0b0: SMPU2 zone 7 disabled</li> <li>0b1: SMPU2 zone 7 enabled</li> </ul>   | RW     |
| 6     | S2RG6E   | SMPU2 zone 6 enabled  | <ul style="list-style-type: none"> <li>0b0: SMPU2 zone 6 disabled</li> <li>0b1: SMPU2 zone 6 enabled</li> </ul>   | RW     |
| 5     | S2RG5E   | SMPU2 zone 5 enabled  | <ul style="list-style-type: none"> <li>0b0: SMPU2 zone 5 disabled</li> <li>0b1: SMPU2 zone 5 enabled</li> </ul>   | RW     |
| 4     | S2RG4E   | SMPU2 zone 4 enabled  | <ul style="list-style-type: none"> <li>0b0: SMPU2 zone 4 disabled</li> <li>0b1: SMPU2 zone 4 enabled</li> </ul>   | RW     |
| 3     | S2RG3E   | SMPU2 zone 3 enabled  | <ul style="list-style-type: none"> <li>0b0: SMPU2 zone 3 disabled</li> <li>0b1: SMPU2 zone 3 enabled</li> </ul>   | RW     |



| Bit | Name   | Function             | Description                                                                                                      | Access |
|-----|--------|----------------------|------------------------------------------------------------------------------------------------------------------|--------|
| 2   | S2RG2E | SMPU2 zone 2 enabled | <ul style="list-style-type: none"><li>● 0b0: SMPU2 zone 2 disabled</li><li>● 0b1: SMPU2 zone 2 enabled</li></ul> | RW     |
| 1   | S2RG1E | SMPU2 zone 1 enabled | <ul style="list-style-type: none"><li>● 0b0: SMPU2 zone 1 disabled</li><li>● 0b1: SMPU2 zone 1 enabled</li></ul> | RW     |
| 0   | S2RG0E | SMPU2 zone 0 enabled | <ul style="list-style-type: none"><li>● 0b0: SMPU2 zone 0 disabled</li><li>● 0b1: SMPU2 zone 0 enabled</li></ul> | RW     |

## 13.4.12 SMPU2 Region Write Permission Register (MPU\_S2RGWP)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |          |          |          |          |          |          |         |         |         |         |         |         |         |         |         |         |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----------|----------|----------|----------|----------|----------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|
| 0x64   | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15       | 14       | 13       | 12       | 11       | 10       | 9       | 8       | 7       | 6       | 5       | 4       | 3       | 2       | 1       | 0       |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |          |          |          |          |          |          |         |         |         |         |         |         |         |         |         |         |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | S2RG15WP | S2RG14WP | S2RG13WP | S2RG12WP | S2RG11WP | S2RG10WP | S2RG9WP | S2RG8WP | S2RG7WP | S2RG6WP | S2RG5WP | S2RG4WP | S2RG3WP | S2RG2WP | S2RG1WP | S2RG0WP |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RW       | RW       | RW       | RW       | RW       | RW       | RW      | RW      | RW      | RW      | RW      | RW      | RW      | RW      | RW      | RW      |

| Bit   | Name     | Function                       | Description                                                                                                                                                 | Access |
|-------|----------|--------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:16 | Reserved | Reserved                       | -                                                                                                                                                           | -      |
| 15    | S2RG15WP | SMPU2 zone 15 write permission | <ul style="list-style-type: none"> <li>0b0: SMPU2 zone 15 enable system DMA_2 to write</li> <li>0b1: SMPU2 zone 15 disable system DMA_2 to write</li> </ul> | RW     |
| 14    | S2RG14WP | SMPU2 zone 14 write permission | <ul style="list-style-type: none"> <li>0b0: SMPU2 zone 14 enable system DMA_2 to write</li> <li>0b1: SMPU2 zone 14 disable system DMA_2 to write</li> </ul> | RW     |
| 13    | S2RG13WP | SMPU2 zone 13 write permission | <ul style="list-style-type: none"> <li>0b0: SMPU2 zone 13 enable system DMA_2 to write</li> <li>0b1: SMPU2 zone 13 disable system DMA_2 to write</li> </ul> | RW     |
| 12    | S2RG12WP | SMPU2 zone 12 write permission | <ul style="list-style-type: none"> <li>0b0: SMPU2 zone 12 enable system DMA_2 to write</li> <li>0b1: SMPU2 zone 12 disable system DMA_2 to write</li> </ul> | RW     |
| 11    | S2RG11WP | SMPU2 zone 11 write permission | <ul style="list-style-type: none"> <li>0b0: SMPU2 zone 11 enable system DMA_2 to write</li> <li>0b1: SMPU2 zone 11 disable system DMA_2 to write</li> </ul> | RW     |
| 10    | S2RG10WP | SMPU2 zone 10 write permission | <ul style="list-style-type: none"> <li>0b0: SMPU2 zone 10 enable system DMA_2 to write</li> <li>0b1: SMPU2 zone 10 disable system DMA_2 to write</li> </ul> | RW     |
| 9     | S2RG9WP  | SMPU2 zone 9 write permission  | <ul style="list-style-type: none"> <li>0b0: SMPU2 zone 9 enable system DMA_2 to write</li> <li>0b1: SMPU2 zone 9 disable system DMA_2 to write</li> </ul>   | RW     |
| 8     | S2RG8WP  | SMPU2 zone 8 write permission  | <ul style="list-style-type: none"> <li>0b0: SMPU2 zone 8 enable system DMA_2 to write</li> <li>0b1: SMPU2 zone 8 disable system DMA_2 to write</li> </ul>   | RW     |

| Bit | Name    | Function                      | Description                                                                                                                                                   | Access |
|-----|---------|-------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 7   | S2RG7WP | SMPU2 zone 7 write permission | <ul style="list-style-type: none"> <li>● 0b0: SMPU2 zone 7 enable system DMA_2 to write</li> <li>● 0b1: SMPU2 zone 7 disable system DMA_2 to write</li> </ul> | RW     |
| 6   | S2RG6WP | SMPU2 zone 6 write permission | <ul style="list-style-type: none"> <li>● 0b0: SMPU2 zone 6 enable system DMA_2 to write</li> <li>● 0b1: SMPU2 zone 6 disable system DMA_2 to write</li> </ul> | RW     |
| 5   | S2RG5WP | SMPU2 zone 5 write permission | <ul style="list-style-type: none"> <li>● 0b0: SMPU2 zone 5 enable system DMA_2 to write</li> <li>● 0b1: SMPU2 zone 5 disable system DMA_2 to write</li> </ul> | RW     |
| 4   | S2RG4WP | SMPU2 zone 4 write permission | <ul style="list-style-type: none"> <li>● 0b0: SMPU2 zone 4 enable system DMA_2 to write</li> <li>● 0b1: SMPU2 zone 4 disable system DMA_2 to write</li> </ul> | RW     |
| 3   | S2RG3WP | SMPU2 zone 3 write permission | <ul style="list-style-type: none"> <li>● 0b0: SMPU2 zone 3 enable system DMA_2 to write</li> <li>● 0b1: SMPU2 zone 3 disable system DMA_2 to write</li> </ul> | RW     |
| 2   | S2RG2WP | SMPU2 zone 2 write permission | <ul style="list-style-type: none"> <li>● 0b0: SMPU2 zone 2 enable system DMA_2 to write</li> <li>● 0b1: SMPU2 zone 2 disable system DMA_2 to write</li> </ul> | RW     |
| 1   | S2RG1WP | SMPU2 zone 1 write permission | <ul style="list-style-type: none"> <li>● 0b0: SMPU2 zone 1 enable system DMA_2 to write</li> <li>● 0b1: SMPU2 zone 1 disable system DMA_2 to write</li> </ul> | RW     |
| 0   | S2RG0WP | SMPU2 zone 0 write permission | <ul style="list-style-type: none"> <li>● 0b0: SMPU2 zone 0 enable system DMA_2 to write</li> <li>● 0b1: SMPU2 zone 0 disable system DMA_2 to write</li> </ul> | RW     |

## 13.4.13 SMPU2 Region Read Permission Register (MPU\_S2RGRP)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |          |          |          |          |          |          |         |         |         |         |         |         |         |         |         |         |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----------|----------|----------|----------|----------|----------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|
| 0x68   | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15       | 14       | 13       | 12       | 11       | 10       | 9       | 8       | 7       | 6       | 5       | 4       | 3       | 2       | 1       | 0       |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |          |          |          |          |          |          |         |         |         |         |         |         |         |         |         |         |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | S2RG15RP | S2RG14RP | S2RG13RP | S2RG12RP | S2RG11RP | S2RG10RP | S2RG9RP | S2RG8RP | S2RG7RP | S2RG6RP | S2RG5RP | S2RG4RP | S2RG3RP | S2RG2RP | S2RG1RP | S2RG0RP |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RW       | RW       | RW       | RW       | RW       | RW       | RW      | RW      | RW      | RW      | RW      | RW      | RW      | RW      | RW      | RW      |

| Bit   | Name     | Function                      | Description                                                                                                                                               | Access |
|-------|----------|-------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:16 | Reserved | Reserved                      | -                                                                                                                                                         | -      |
| 15    | S2RG15RP | SMPU2 zone 15 read permission | <ul style="list-style-type: none"> <li>0b0: SMPU2 zone 15 enable system DMA_2 to read</li> <li>0b1: SMPU2 zone 15 disable system DMA_2 to read</li> </ul> | RW     |
| 14    | S2RG14RP | SMPU2 zone 14 read permission | <ul style="list-style-type: none"> <li>0b0: SMPU2 zone 14 enable system DMA_2 to read</li> <li>0b1: SMPU2 zone 14 disable system DMA_2 to read</li> </ul> | RW     |
| 13    | S2RG13RP | SMPU2 zone 13 read permission | <ul style="list-style-type: none"> <li>0b0: SMPU2 zone 13 enable system DMA_2 to read</li> <li>0b1: SMPU2 zone 13 disable system DMA_2 to read</li> </ul> | RW     |
| 12    | S2RG12RP | SMPU2 zone 12 read permission | <ul style="list-style-type: none"> <li>0b0: SMPU2 zone 12 enable system DMA_2 to read</li> <li>0b1: SMPU2 zone 12 disable system DMA_2 to read</li> </ul> | RW     |
| 11    | S2RG11RP | SMPU2 zone 11 read permission | <ul style="list-style-type: none"> <li>0b0: SMPU2 zone 11 enable system DMA_2 to read</li> <li>0b1: SMPU2 zone 11 disable system DMA_2 to read</li> </ul> | RW     |
| 10    | S2RG10RP | SMPU2 zone 10 read permission | <ul style="list-style-type: none"> <li>0b0: SMPU2 zone 10 enable system DMA_2 to read</li> <li>0b1: SMPU2 zone 10 disable system DMA_2 to read</li> </ul> | RW     |
| 9     | S2RG9RP  | SMPU2 zone 9 read permission  | <ul style="list-style-type: none"> <li>0b0: SMPU2 zone 9 enable system DMA_2 to read</li> <li>0b1: SMPU2 zone 9 disable system DMA_2 to read</li> </ul>   | RW     |
| 8     | S2RG8RP  | SMPU2 zone 8 read permission  | <ul style="list-style-type: none"> <li>0b0: SMPU2 zone 8 enable system DMA_2 to read</li> <li>0b1: SMPU2 zone 8 disable system DMA_2 to read</li> </ul>   | RW     |

| Bit | Name    | Function                     | Description                                                                                                                                                 | Access |
|-----|---------|------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 7   | S2RG7RP | SMPU2 zone 7 read permission | <ul style="list-style-type: none"> <li>● 0b0: SMPU2 zone 7 enable system DMA_2 to read</li> <li>● 0b1: SMPU2 zone 7 disable system DMA_2 to read</li> </ul> | RW     |
| 6   | S2RG6RP | SMPU2 zone 6 read permission | <ul style="list-style-type: none"> <li>● 0b0: SMPU2 zone 6 enable system DMA_2 to read</li> <li>● 0b1: SMPU2 zone 6 disable system DMA_2 to read</li> </ul> | RW     |
| 5   | S2RG5RP | SMPU2 zone 5 read permission | <ul style="list-style-type: none"> <li>● 0b0: SMPU2 zone 5 enable system DMA_2 to read</li> <li>● 0b1: SMPU2 zone 5 disable system DMA_2 to read</li> </ul> | RW     |
| 4   | S2RG4RP | SMPU2 zone 4 read permission | <ul style="list-style-type: none"> <li>● 0b0: SMPU2 zone 4 enable system DMA_2 to read</li> <li>● 0b1: SMPU2 zone 4 disable system DMA_2 to read</li> </ul> | RW     |
| 3   | S2RG3RP | SMPU2 zone 3 read permission | <ul style="list-style-type: none"> <li>● 0b0: SMPU2 zone 3 enable system DMA_2 to read</li> <li>● 0b1: SMPU2 zone 3 disable system DMA_2 to read</li> </ul> | RW     |
| 2   | S2RG2RP | SMPU2 zone 2 read permission | <ul style="list-style-type: none"> <li>● 0b0: SMPU2 zone 2 enable system DMA_2 to read</li> <li>● 0b1: SMPU2 zone 2 disable system DMA_2 to read</li> </ul> | RW     |
| 1   | S2RG1RP | SMPU2 zone 1 read permission | <ul style="list-style-type: none"> <li>● 0b0: SMPU2 zone 1 enable system DMA_2 to read</li> <li>● 0b1: SMPU2 zone 1 disable system DMA_2 to read</li> </ul> | RW     |
| 0   | S2RG0RP | SMPU2 zone 0 read permission | <ul style="list-style-type: none"> <li>● 0b0: SMPU2 zone 0 enable system DMA_2 to read</li> <li>● 0b1: SMPU2 zone 0 disable system DMA_2 to read</li> </ul> | RW     |

## 13.4.14 SMPU2 Control Register (MPU\_S2CR)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |        |   |          |   |          |   |          |   |          |  |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|--------|---|----------|---|----------|---|----------|---|----------|--|
| 0x6C   | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7      | 6 | 5        | 4 | 3        | 2 | 1        | 0 |          |  |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |        |   |          |   |          |   |          |   |          |  |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   | SMPU2E |   | Reserved |   | SMPU2ACT |   | SMPU2BWP |   | SMPU2BRP |  |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   | RW     |   |          |   | RW       |   | RW       |   | RW       |  |

| Bit  | Name     | Function                          | Description                                                                                                                                                                                                                                                                                                                                        | Access |
|------|----------|-----------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:8 | Reserved | Reserved                          | -                                                                                                                                                                                                                                                                                                                                                  | -      |
| 7    | SMPU2E   | SMPU2 enable                      | <ul style="list-style-type: none"> <li>0b0: SMPU2 disabled</li> <li>0b1: SMPU2 enabled</li> </ul>                                                                                                                                                                                                                                                  | RW     |
| 6:4  | Reserved | Reserved                          | -                                                                                                                                                                                                                                                                                                                                                  | -      |
| 3:2  | SMPU2ACT | SMPU2 action selection            | Set the action when the system DMA_2 access is disabled <ul style="list-style-type: none"> <li>0b00: Read access reads 0, write access ignore</li> <li>0b01: Read access reads 0, write access ignore; bus error generate</li> <li>0b10: Read access reads 0, write access ignore; non-maskable interrupt generate</li> <li>0b11: Reset</li> </ul> | RW     |
| 1    | SMPU2BWP | SMPU2 background write permission | <ul style="list-style-type: none"> <li>0b0: Enable system DMA_2 write access in SMPU2 background space</li> <li>0b1: Disable system DMA_2 write access in SMPU2 background space</li> </ul>                                                                                                                                                        | RW     |
| 0    | SMPU2BRP | SMPU2 background read permission  | <ul style="list-style-type: none"> <li>0b0: Enable system DMA_2 read access in SMPU2 background space</li> <li>0b1: Disable system DMA_2 read access in SMPU2 background space</li> </ul>                                                                                                                                                          | RW     |

**Note**

When multiple area settings overlap, the priority is: Setting prohibited > setting enabled.

## 13.4.15 FMPU Region Enable Register (MPU\_FRGE)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |       |       |       |       |       |       |       |       |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|-------|-------|-------|-------|-------|-------|-------|-------|
| 0x70   | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7     | 6     | 5     | 4     | 3     | 2     | 1     | 0     |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |       |       |       |       |       |       |       |       |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   | FRG7E | FRG6E | FRG5E | FRG4E | FRG3E | FRG2E | FRG1E | FRG0E |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   | RW    | RW    | RW    | RW    | RW    | RW    | RW    | RW    |

| Bit  | Name     | Function            | Description                                                                                                   | Access |
|------|----------|---------------------|---------------------------------------------------------------------------------------------------------------|--------|
| 31:8 | Reserved | Reserved            | -                                                                                                             | -      |
| 7    | FRG7E    | FMPU zone 7 enabled | <ul style="list-style-type: none"> <li>0b0: FMPU zone 7 disabled</li> <li>0b1: FMPU zone 7 enabled</li> </ul> | RW     |
| 6    | FRG6E    | FMPU zone 6 enabled | <ul style="list-style-type: none"> <li>0b0: FMPU zone 6 disabled</li> <li>0b1: FMPU zone 6 enabled</li> </ul> | RW     |
| 5    | FRG5E    | FMPU zone 5 enabled | <ul style="list-style-type: none"> <li>0b0: FMPU zone 5 disabled</li> <li>0b1: FMPU zone 5 enabled</li> </ul> | RW     |
| 4    | FRG4E    | FMPU zone 4 enabled | <ul style="list-style-type: none"> <li>0b0: FMPU zone 4 disabled</li> <li>0b1: FMPU zone 4 enabled</li> </ul> | RW     |
| 3    | FRG3E    | FMPU zone 3 enabled | <ul style="list-style-type: none"> <li>0b0: FMPU zone 3 disabled</li> <li>0b1: FMPU zone 3 enabled</li> </ul> | RW     |
| 2    | FRG2E    | FMPU zone 2 enabled | <ul style="list-style-type: none"> <li>0b0: FMPU zone 2 disabled</li> <li>0b1: FMPU zone 2 enabled</li> </ul> | RW     |
| 1    | FRG1E    | FMPU zone 1 enabled | <ul style="list-style-type: none"> <li>0b0: FMPU zone 1 disabled</li> <li>0b1: FMPU zone 1 enabled</li> </ul> | RW     |
| 0    | FRG0E    | FMPU zone 0 enabled | <ul style="list-style-type: none"> <li>0b0: FMPU zone 0 disabled</li> <li>0b1: FMPU zone 0 enabled</li> </ul> | RW     |

## 13.4.16 FMPU Region Write Permission Register (MPU\_FRGWP)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |        |        |        |        |        |        |        |        |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|--------|--------|--------|--------|--------|--------|--------|--------|
| 0x74   | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7      | 6      | 5      | 4      | 3      | 2      | 1      | 0      |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |        |        |        |        |        |        |        |        |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   | FRG7WP | FRG6WP | FRG5WP | FRG4WP | FRG3WP | FRG2WP | FRG1WP | FRG0WP |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   | RW     | RW     | RW     | RW     | RW     | RW     | RW     | RW     |

| Bit  | Name     | Function                     | Description                                                                                                                                       | Access |
|------|----------|------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:8 | Reserved | Reserved                     | -                                                                                                                                                 | -      |
| 7    | FRG7WP   | FMPU zone 7 write permission | <ul style="list-style-type: none"> <li>0b0: FMPU zone 7 enable USBFS-DMA to write</li> <li>0b1: FMPU zone 7 disable USBFS-DMA to write</li> </ul> | RW     |
| 6    | FRG6WP   | FMPU zone 6 write permission | <ul style="list-style-type: none"> <li>0b0: FMPU zone 6 enable USBFS-DMA to write</li> <li>0b1: FMPU zone 6 disable USBFS-DMA to write</li> </ul> | RW     |
| 5    | FRG5WP   | FMPU zone 5 write permission | <ul style="list-style-type: none"> <li>0b0: FMPU zone 5 enable USBFS-DMA to write</li> <li>0b1: FMPU zone 5 disable USBFS-DMA to write</li> </ul> | RW     |
| 4    | FRG4WP   | FMPU zone 4 write permission | <ul style="list-style-type: none"> <li>0b0: FMPU zone 4 enable USBFS-DMA to write</li> <li>0b1: FMPU zone 4 disable USBFS-DMA to write</li> </ul> | RW     |
| 3    | FRG3WP   | FMPU zone 3 write permission | <ul style="list-style-type: none"> <li>0b0: FMPU zone 3 enable USBFS-DMA to write</li> <li>0b1: FMPU zone 3 disable USBFS-DMA to write</li> </ul> | RW     |
| 2    | FRG2WP   | FMPU zone 2 write permission | <ul style="list-style-type: none"> <li>0b0: FMPU zone 2 enable USBFS-DMA to write</li> <li>0b1: FMPU zone 2 disable USBFS-DMA to write</li> </ul> | RW     |
| 1    | FRG1WP   | FMPU zone 1 write permission | <ul style="list-style-type: none"> <li>0b0: FMPU zone 1 enable USBFS-DMA to write</li> <li>0b1: FMPU zone 1 disable USBFS-DMA to write</li> </ul> | RW     |
| 0    | FRG0WP   | FMPU zone 0 write permission | <ul style="list-style-type: none"> <li>0b0: FMPU zone 0 enable USBFS-DMA to write</li> <li>0b1: FMPU zone 0 disable USBFS-DMA to write</li> </ul> | RW     |



## 13.4.17 FMPU Region Read Permission Register (MPU\_FRGRP)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |        |        |        |        |        |        |        |        |    |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|--------|--------|--------|--------|--------|--------|--------|--------|----|
| 0x78   | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7      | 6      | 5      | 4      | 3      | 2      | 1      | 0      |    |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |        |        |        |        |        |        |        |        |    |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   | FRG7RP | FRG6RP | FRG5RP | FRG4RP | FRG3RP | FRG2RP | FRG1RP | FRG0RP |    |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   | RW     | RW     | RW     | RW     | RW     | RW     | RW     | RW     | RW |

| Bit  | Name     | Function                    | Description                                                                                                                                     | Access |
|------|----------|-----------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:8 | Reserved | Reserved                    | -                                                                                                                                               | -      |
| 7    | FRG7RP   | FMPU zone 7 read permission | <ul style="list-style-type: none"> <li>0b0: FMPU zone 7 enable USBFS-DMA to read</li> <li>0b1: FMPU zone 7 disable USBFS-DMA to read</li> </ul> | RW     |
| 6    | FRG6RP   | FMPU zone 6 read permission | <ul style="list-style-type: none"> <li>0b0: FMPU zone 6 enable USBFS-DMA to read</li> <li>0b1: FMPU zone 6 disable USBFS-DMA to read</li> </ul> | RW     |
| 5    | FRG5RP   | FMPU zone 5 read permission | <ul style="list-style-type: none"> <li>0b0: FMPU zone 5 enable USBFS-DMA to read</li> <li>0b1: FMPU zone 5 disable USBFS-DMA to read</li> </ul> | RW     |
| 4    | FRG4RP   | FMPU zone 4 read permission | <ul style="list-style-type: none"> <li>0b0: FMPU zone 4 enable USBFS-DMA to read</li> <li>0b1: FMPU zone 4 disable USBFS-DMA to read</li> </ul> | RW     |
| 3    | FRG3RP   | FMPU zone 3 read permission | <ul style="list-style-type: none"> <li>0b0: FMPU zone 3 enable USBFS-DMA to read</li> <li>0b1: FMPU zone 3 disable USBFS-DMA to read</li> </ul> | RW     |
| 2    | FRG2RP   | FMPU zone 2 read permission | <ul style="list-style-type: none"> <li>0b0: FMPU zone 2 enable USBFS-DMA to read</li> <li>0b1: FMPU zone 2 disable USBFS-DMA to read</li> </ul> | RW     |
| 1    | FRG1RP   | FMPU zone 1 read permission | <ul style="list-style-type: none"> <li>0b0: FMPU zone 1 enable USBFS-DMA to read</li> <li>0b1: FMPU zone 1 disable USBFS-DMA to read</li> </ul> | RW     |
| 0    | FRG0RP   | FMPU zone 0 read permission | <ul style="list-style-type: none"> <li>0b0: FMPU zone 0 enable USBFS-DMA to read</li> <li>0b1: FMPU zone 0 disable USBFS-DMA to read</li> </ul> | RW     |

## 13.4.18 FMPU Control Register (MPU\_FCR)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |       |          |   |         |         |         |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|-------|----------|---|---------|---------|---------|---|---|
| 0x7C   | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7     | 6        | 5 | 4       | 3       | 2       | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |       |          |   |         |         |         |   |   |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   | FMPUE | Reserved |   | FMPUACT | FMPUBWP | FMPUBRP |   |   |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   | RW    |          |   | RW      | RW      | RW      |   |   |

| Bit  | Name     | Function                         | Description                                                                                                                                                                                                                                                                                                                                 | Access |
|------|----------|----------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:8 | Reserved | Reserved                         | -                                                                                                                                                                                                                                                                                                                                           | -      |
| 7    | FMPUE    | FMPU enable                      | <ul style="list-style-type: none"> <li>0b0: FMPU disabled</li> <li>0b1: FMPU enabled</li> </ul>                                                                                                                                                                                                                                             | RW     |
| 6:4  | Reserved | Reserved                         | -                                                                                                                                                                                                                                                                                                                                           | -      |
| 3:2  | FMPUACT  | FMPU action selection            | Set the action when USBFS-DMA access is disabled <ul style="list-style-type: none"> <li>0b00: Read access reads 0, write access ignore</li> <li>0b01: Read access reads 0, write access ignore; bus error generate</li> <li>0b10: Read access reads 0, write access ignore; non-maskable interrupt generate</li> <li>0b11: Reset</li> </ul> | RW     |
| 1    | FMPUBWP  | FMPU background write permission | <ul style="list-style-type: none"> <li>0b0: Enable USBFS-DMA write access in FMPU background space</li> <li>0b1: Disable USBFS-DMA write access in FMPU background space</li> </ul>                                                                                                                                                         | RW     |
| 0    | FMPUBRP  | FMPU background read permission  | <ul style="list-style-type: none"> <li>0b0: Enable USBFS-DMA read access in FMPU background space</li> <li>0b1: Disable USBFS-DMA read access in FMPU background space</li> </ul>                                                                                                                                                           | RW     |

**Note**

When multiple area settings overlap, the priority is: Setting prohibited > setting enabled.

## 13.4.19 HMPU Region Enable Register (MPU\_HRGE)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |       |       |       |       |       |       |       |       |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|-------|-------|-------|-------|-------|-------|-------|-------|
| 0x80   | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7     | 6     | 5     | 4     | 3     | 2     | 1     | 0     |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |       |       |       |       |       |       |       |       |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   | HRG7E | HRG6E | HRG5E | HRG4E | HRG3E | HRG2E | HRG1E | HRG0E |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   | RW    | RW    | RW    | RW    | RW    | RW    | RW    | RW    |

| Bit  | Name     | Function            | Description                                                                                                   | Access |
|------|----------|---------------------|---------------------------------------------------------------------------------------------------------------|--------|
| 31:8 | Reserved | Reserved            | -                                                                                                             | -      |
| 7    | HRG7E    | HMPU zone 7 enabled | <ul style="list-style-type: none"> <li>0b0: HMPU zone 7 disabled</li> <li>0b1: HMPU zone 7 enabled</li> </ul> | RW     |
| 6    | HRG6E    | HMPU zone 6 enabled | <ul style="list-style-type: none"> <li>0b0: HMPU zone 6 disabled</li> <li>0b1: HMPU zone 6 enabled</li> </ul> | RW     |
| 5    | HRG5E    | HMPU zone 5 enabled | <ul style="list-style-type: none"> <li>0b0: HMPU zone 5 disabled</li> <li>0b1: HMPU zone 5 enabled</li> </ul> | RW     |
| 4    | HRG4E    | HMPU zone 4 enabled | <ul style="list-style-type: none"> <li>0b0: HMPU zone 4 disabled</li> <li>0b1: HMPU zone 4 enabled</li> </ul> | RW     |
| 3    | HRG3E    | HMPU zone 3 enabled | <ul style="list-style-type: none"> <li>0b0: HMPU zone 3 disabled</li> <li>0b1: HMPU zone 3 enabled</li> </ul> | RW     |
| 2    | HRG2E    | HMPU zone 2 enabled | <ul style="list-style-type: none"> <li>0b0: HMPU zone 2 disabled</li> <li>0b1: HMPU zone 2 enabled</li> </ul> | RW     |
| 1    | HRG1E    | HMPU zone 1 enabled | <ul style="list-style-type: none"> <li>0b0: HMPU zone 1 disabled</li> <li>0b1: HMPU zone 1 enabled</li> </ul> | RW     |
| 0    | HRG0E    | HMPU zone 0 enabled | <ul style="list-style-type: none"> <li>0b0: HMPU zone 0 disabled</li> <li>0b1: HMPU zone 0 enabled</li> </ul> | RW     |

## 13.4.20 HMPU Region Write Permission Register (MPU\_HRGWP)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |        |        |        |        |        |        |        |        |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|--------|--------|--------|--------|--------|--------|--------|--------|
| 0x84   | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7      | 6      | 5      | 4      | 3      | 2      | 1      | 0      |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |        |        |        |        |        |        |        |        |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   | HRG7WP | HRG6WP | HRG5WP | HRG4WP | HRG3WP | HRG2WP | HRG1WP | HRG0WP |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   | RW     | RW     | RW     | RW     | RW     | RW     | RW     | RW     |

| Bit  | Name     | Function                     | Description                                                                                                                                       | Access |
|------|----------|------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:8 | Reserved | Reserved                     | -                                                                                                                                                 | -      |
| 7    | HRG7WP   | HMPU zone 7 write permission | <ul style="list-style-type: none"> <li>0b0: HMPU zone 7 enable USBHS-DMA to write</li> <li>0b1: HMPU zone 7 disable USBHS-DMA to write</li> </ul> | RW     |
| 6    | HRG6WP   | HMPU zone 6 write permission | <ul style="list-style-type: none"> <li>0b0: HMPU zone 6 enable USBHS-DMA to write</li> <li>0b1: HMPU zone 6 disable USBHS-DMA to write</li> </ul> | RW     |
| 5    | HRG5WP   | HMPU zone 5 write permission | <ul style="list-style-type: none"> <li>0b0: HMPU zone 5 enable USBHS-DMA to write</li> <li>0b1: HMPU zone 5 disable USBHS-DMA to write</li> </ul> | RW     |
| 4    | HRG4WP   | HMPU zone 4 write permission | <ul style="list-style-type: none"> <li>0b0: HMPU zone 4 enable USBHS-DMA to write</li> <li>0b1: HMPU zone 4 disable USBHS-DMA to write</li> </ul> | RW     |
| 3    | HRG3WP   | HMPU zone 3 write permission | <ul style="list-style-type: none"> <li>0b0: HMPU zone 3 enable USBHS-DMA to write</li> <li>0b1: HMPU zone 3 disable USBHS-DMA to write</li> </ul> | RW     |
| 2    | HRG2WP   | HMPU zone 2 write permission | <ul style="list-style-type: none"> <li>0b0: HMPU zone 2 enable USBHS-DMA to write</li> <li>0b1: HMPU zone 2 disable USBHS-DMA to write</li> </ul> | RW     |
| 1    | HRG1WP   | HMPU zone 1 write permission | <ul style="list-style-type: none"> <li>0b0: HMPU zone 1 enable USBHS-DMA to write</li> <li>0b1: HMPU zone 1 disable USBHS-DMA to write</li> </ul> | RW     |
| 0    | HRG0WP   | HMPU zone 0 write permission | <ul style="list-style-type: none"> <li>0b0: HMPU zone 0 enable USBHS-DMA to write</li> <li>0b1: HMPU zone 0 disable USBHS-DMA to write</li> </ul> | RW     |

## 13.4.21 HMPU Region Read Permission Register (MPU\_HRGRP)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |        |        |        |        |        |        |        |        |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|--------|--------|--------|--------|--------|--------|--------|--------|
| 0x88   | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7      | 6      | 5      | 4      | 3      | 2      | 1      | 0      |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |        |        |        |        |        |        |        |        |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   | HRG7RP | HRG6RP | HRG5RP | HRG4RP | HRG3RP | HRG2RP | HRG1RP | HRG0RP |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   | RW     | RW     | RW     | RW     | RW     | RW     | RW     | RW     |

| Bit  | Name     | Function                    | Description                                                                                                                                     | Access |
|------|----------|-----------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:8 | Reserved | Reserved                    | -                                                                                                                                               | -      |
| 7    | HRG7RP   | HMPU zone 7 read permission | <ul style="list-style-type: none"> <li>0b0: HMPU zone 7 enable USBHS-DMA to read</li> <li>0b1: HMPU zone 7 disable USBHS-DMA to read</li> </ul> | RW     |
| 6    | HRG6RP   | HMPU zone 6 read permission | <ul style="list-style-type: none"> <li>0b0: HMPU zone 6 enable USBHS-DMA to read</li> <li>0b1: HMPU zone 6 disable USBHS-DMA to read</li> </ul> | RW     |
| 5    | HRG5RP   | HMPU zone 5 read permission | <ul style="list-style-type: none"> <li>0b0: HMPU zone 5 enable USBHS-DMA to read</li> <li>0b1: HMPU zone 5 disable USBHS-DMA to read</li> </ul> | RW     |
| 4    | HRG4RP   | HMPU zone 4 read permission | <ul style="list-style-type: none"> <li>0b0: HMPU zone 4 enable USBHS-DMA to read</li> <li>0b1: HMPU zone 4 disable USBHS-DMA to read</li> </ul> | RW     |
| 3    | HRG3RP   | HMPU zone 3 read permission | <ul style="list-style-type: none"> <li>0b0: HMPU zone 3 enable USBHS-DMA to read</li> <li>0b1: HMPU zone 3 disable USBHS-DMA to read</li> </ul> | RW     |
| 2    | HRG2RP   | HMPU zone 2 read permission | <ul style="list-style-type: none"> <li>0b0: HMPU zone 2 enable USBHS-DMA to read</li> <li>0b1: HMPU zone 2 disable USBHS-DMA to read</li> </ul> | RW     |
| 1    | HRG1RP   | HMPU zone 1 read permission | <ul style="list-style-type: none"> <li>0b0: HMPU zone 1 enable USBHS-DMA to read</li> <li>0b1: HMPU zone 1 disable USBHS-DMA to read</li> </ul> | RW     |
| 0    | HRG0RP   | HMPU zone 0 read permission | <ul style="list-style-type: none"> <li>0b0: HMPU zone 0 enable USBHS-DMA to read</li> <li>0b1: HMPU zone 0 disable USBHS-DMA to read</li> </ul> | RW     |

## 13.4.22 HMPU Control Register (MPU\_HCR)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |       |          |         |         |         |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|-------|----------|---------|---------|---------|---|---|---|
| 0x8C   | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7     | 6        | 5       | 4       | 3       | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |       |          |         |         |         |   |   |   |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   | HMPUE | Reserved | HMPUACT | HMPUBWP | HMPUBRP |   |   |   |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   | RW    |          | RW      | RW      | RW      |   |   |   |

| Bit  | Name     | Function                         | Description                                                                                                                                                                                                                                                                                                                                 | Access |
|------|----------|----------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:8 | Reserved | Reserved                         | -                                                                                                                                                                                                                                                                                                                                           | -      |
| 7    | HMPUE    | HMPU enable                      | <ul style="list-style-type: none"> <li>0b0: HMPU disabled</li> <li>0b1: HMPU enabled</li> </ul>                                                                                                                                                                                                                                             | RW     |
| 6:4  | Reserved | Reserved                         | -                                                                                                                                                                                                                                                                                                                                           | -      |
| 3:2  | HMPUACT  | HMPU action selection            | Set the action when USBHS-DMA access is disabled <ul style="list-style-type: none"> <li>0b00: Read access reads 0, write access ignore</li> <li>0b01: Read access reads 0, write access ignore; bus error generate</li> <li>0b10: Read access reads 0, write access ignore; non-maskable interrupt generate</li> <li>0b11: Reset</li> </ul> | RW     |
| 1    | HMPUBWP  | HMPU background write permission | <ul style="list-style-type: none"> <li>0b0: Enable USBHS-DMA write access in HMPU background space</li> <li>0b1: Disable USBHS-DMA write access in HMPU background space</li> </ul>                                                                                                                                                         | RW     |
| 0    | HMPUBRP  | HMPU background read permission  | <ul style="list-style-type: none"> <li>0b0: Enable USBHS-DMA read access in HMPU background space</li> <li>0b1: Disable USBHS-DMA read access in HMPU background space</li> </ul>                                                                                                                                                           | RW     |

**Note**

When multiple area settings overlap, the priority is: Setting prohibited > setting enabled.

## 13.4.23 EMPU Region Enable Register (MPU\_ERGE)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |       |       |       |       |       |       |       |       |    |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|-------|-------|-------|-------|-------|-------|-------|-------|----|
| 0x90   | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7     | 6     | 5     | 4     | 3     | 2     | 1     | 0     |    |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |       |       |       |       |       |       |       |       |    |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   | ERG7E | ERG6E | ERG5E | ERG4E | ERG3E | ERG2E | ERG1E | ERG0E |    |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   | RW    | RW    | RW    | RW    | RW    | RW    | RW    | RW    | RW |

| Bit  | Name     | Function            | Description                                                                                                   | Access |
|------|----------|---------------------|---------------------------------------------------------------------------------------------------------------|--------|
| 31:8 | Reserved | Reserved            | -                                                                                                             | -      |
| 7    | ERG7E    | EMPU zone 7 enabled | <ul style="list-style-type: none"> <li>0b0: EMPU zone 7 disabled</li> <li>0b1: EMPU zone 7 enabled</li> </ul> | RW     |
| 6    | ERG6E    | EMPU zone 6 enabled | <ul style="list-style-type: none"> <li>0b0: EMPU zone 6 disabled</li> <li>0b1: EMPU zone 6 enabled</li> </ul> | RW     |
| 5    | ERG5E    | EMPU zone 5 enabled | <ul style="list-style-type: none"> <li>0b0: EMPU zone 5 disabled</li> <li>0b1: EMPU zone 5 enabled</li> </ul> | RW     |
| 4    | ERG4E    | EMPU zone 4 enabled | <ul style="list-style-type: none"> <li>0b0: EMPU zone 4 disabled</li> <li>0b1: EMPU zone 4 enabled</li> </ul> | RW     |
| 3    | ERG3E    | EMPU zone 3 enabled | <ul style="list-style-type: none"> <li>0b0: EMPU zone 3 disabled</li> <li>0b1: EMPU zone 3 enabled</li> </ul> | RW     |
| 2    | ERG2E    | EMPU zone 2 enabled | <ul style="list-style-type: none"> <li>0b0: EMPU zone 2 disabled</li> <li>0b1: EMPU zone 2 enabled</li> </ul> | RW     |
| 1    | ERG1E    | EMPU zone 1 enabled | <ul style="list-style-type: none"> <li>0b0: EMPU zone 1 disabled</li> <li>0b1: EMPU zone 1 enabled</li> </ul> | RW     |
| 0    | ERG0E    | EMPU zone 0 enabled | <ul style="list-style-type: none"> <li>0b0: EMPU zone 0 disabled</li> <li>0b1: EMPU zone 0 enabled</li> </ul> | RW     |

## 13.4.24 EMPU Region Write Permission Register (MPU\_ERGWP)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |        |        |        |        |        |        |        |        |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|--------|--------|--------|--------|--------|--------|--------|--------|
| 0x94   | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7      | 6      | 5      | 4      | 3      | 2      | 1      | 0      |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |        |        |        |        |        |        |        |        |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   | ERG7WP | ERG6WP | ERG5WP | ERG4WP | ERG3WP | ERG2WP | ERG1WP | ERG0WP |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   | RW     | RW     | RW     | RW     | RW     | RW     | RW     | RW     |

| Bit  | Name     | Function                     | Description                                                                                                                                   | Access |
|------|----------|------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:8 | Reserved | Reserved                     | -                                                                                                                                             | -      |
| 7    | ERG7WP   | EMPU zone 7 write permission | <ul style="list-style-type: none"> <li>0b0: EMPU zone 7 enable ETH-DMA to write</li> <li>0b1: EMPU zone 7 disable ETH-DMA to write</li> </ul> | RW     |
| 6    | ERG6WP   | EMPU zone 6 write permission | <ul style="list-style-type: none"> <li>0b0: EMPU zone 6 enable ETH-DMA to write</li> <li>0b1: EMPU zone 6 disable ETH-DMA to write</li> </ul> | RW     |
| 5    | ERG5WP   | EMPU zone 5 write permission | <ul style="list-style-type: none"> <li>0b0: EMPU zone 5 enable ETH-DMA to write</li> <li>0b1: EMPU zone 5 disable ETH-DMA to write</li> </ul> | RW     |
| 4    | ERG4WP   | EMPU zone 4 write permission | <ul style="list-style-type: none"> <li>0b0: EMPU zone 4 enable ETH-DMA to write</li> <li>0b1: EMPU zone 4 disable ETH-DMA to write</li> </ul> | RW     |
| 3    | ERG3WP   | EMPU zone 3 write permission | <ul style="list-style-type: none"> <li>0b0: EMPU zone 3 enable ETH-DMA to write</li> <li>0b1: EMPU zone 3 disable ETH-DMA to write</li> </ul> | RW     |
| 2    | ERG2WP   | EMPU zone 2 write permission | <ul style="list-style-type: none"> <li>0b0: EMPU zone 2 enable ETH-DMA to write</li> <li>0b1: EMPU zone 2 disable ETH-DMA to write</li> </ul> | RW     |
| 1    | ERG1WP   | EMPU zone 1 write permission | <ul style="list-style-type: none"> <li>0b0: EMPU zone 1 enable ETH-DMA to write</li> <li>0b1: EMPU zone 1 disable ETH-DMA to write</li> </ul> | RW     |
| 0    | ERG0WP   | EMPU zone 0 write permission | <ul style="list-style-type: none"> <li>0b0: EMPU zone 0 enable ETH-DMA to write</li> <li>0b1: EMPU zone 0 disable ETH-DMA to write</li> </ul> | RW     |



## 13.4.25 EMPU Region Read Permission Register (MPU\_ERGRP)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |        |        |        |        |        |        |        |        |    |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|--------|--------|--------|--------|--------|--------|--------|--------|----|
| 0x98   | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7      | 6      | 5      | 4      | 3      | 2      | 1      | 0      |    |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |        |        |        |        |        |        |        |        |    |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   | ERG7RP | ERG6RP | ERG5RP | ERG4RP | ERG3RP | ERG2RP | ERG1RP | ERG0RP |    |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   | RW     | RW     | RW     | RW     | RW     | RW     | RW     | RW     | RW |

| Bit  | Name     | Function                    | Description                                                                                                                                 | Access |
|------|----------|-----------------------------|---------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:8 | Reserved | Reserved                    | -                                                                                                                                           | -      |
| 7    | ERG7RP   | EMPU zone 7 read permission | <ul style="list-style-type: none"> <li>0b0: EMPU zone 7 enable ETH-DMA to read</li> <li>0b1: EMPU zone 7 disable ETH-DMA to read</li> </ul> | RW     |
| 6    | ERG6RP   | EMPU zone 6 read permission | <ul style="list-style-type: none"> <li>0b0: EMPU zone 6 enable ETH-DMA to read</li> <li>0b1: EMPU zone 6 disable ETH-DMA to read</li> </ul> | RW     |
| 5    | ERG5RP   | EMPU zone 5 read permission | <ul style="list-style-type: none"> <li>0b0: EMPU zone 5 enable ETH-DMA to read</li> <li>0b1: EMPU zone 5 disable ETH-DMA to read</li> </ul> | RW     |
| 4    | ERG4RP   | EMPU zone 4 read permission | <ul style="list-style-type: none"> <li>0b0: EMPU zone 4 enable ETH-DMA to read</li> <li>0b1: EMPU zone 4 disable ETH-DMA to read</li> </ul> | RW     |
| 3    | ERG3RP   | EMPU zone 3 read permission | <ul style="list-style-type: none"> <li>0b0: EMPU zone 3 enable ETH-DMA to read</li> <li>0b1: EMPU zone 3 disable ETH-DMA to read</li> </ul> | RW     |
| 2    | ERG2RP   | EMPU zone 2 read permission | <ul style="list-style-type: none"> <li>0b0: EMPU zone 2 enable ETH-DMA to read</li> <li>0b1: EMPU zone 2 disable ETH-DMA to read</li> </ul> | RW     |
| 1    | ERG1RP   | EMPU zone 1 read permission | <ul style="list-style-type: none"> <li>0b0: EMPU zone 1 enable ETH-DMA to read</li> <li>0b1: EMPU zone 1 disable ETH-DMA to read</li> </ul> | RW     |
| 0    | ERG0RP   | EMPU zone 0 read permission | <ul style="list-style-type: none"> <li>0b0: EMPU zone 0 enable ETH-DMA to read</li> <li>0b1: EMPU zone 0 disable ETH-DMA to read</li> </ul> | RW     |

## 13.4.26 EMPU Control Register (MPU\_ECR)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |       |   |          |   |         |   |         |   |         |  |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|-------|---|----------|---|---------|---|---------|---|---------|--|
| 0x9C   | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7     | 6 | 5        | 4 | 3       | 2 | 1       | 0 |         |  |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |       |   |          |   |         |   |         |   |         |  |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   | EMPUE |   | Reserved |   | EMPUACT |   | EMPUBWP |   | EMPUBRP |  |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   | RW    |   |          |   | RW      |   | RW      |   | RW      |  |

| Bit  | Name     | Function                         | Description                                                                                                                                                                                                                                                                                                                               | Access |
|------|----------|----------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:8 | Reserved | Reserved                         | -                                                                                                                                                                                                                                                                                                                                         | -      |
| 7    | EMPUE    | EMPU enable                      | <ul style="list-style-type: none"> <li>0b0: EMPU disabled</li> <li>0b1: EMPU enabled</li> </ul>                                                                                                                                                                                                                                           | RW     |
| 6:4  | Reserved | Reserved                         | -                                                                                                                                                                                                                                                                                                                                         | -      |
| 3:2  | EMPUACT  | EMPU action selection            | Set the action when ETH-DMA access is disabled <ul style="list-style-type: none"> <li>0b00: Read access reads 0, write access ignore</li> <li>0b01: Read access reads 0, write access ignore; bus error generate</li> <li>0b10: Read access reads 0, write access ignore; non-maskable interrupt generate</li> <li>0b11: Reset</li> </ul> | RW     |
| 1    | EMPUBWP  | EMPU background write permission | <ul style="list-style-type: none"> <li>0b0: Enable ETH-DMA write access in EMPU background space</li> <li>0b1: Disable ETH-DMA write access in EMPU background space</li> </ul>                                                                                                                                                           | RW     |
| 0    | EMPUBRP  | EMPU background read permission  | <ul style="list-style-type: none"> <li>0b0: Enable ETH-DMA read access in EMPU background space</li> <li>0b1: Disable ETH-DMA read access in EMPU background space</li> </ul>                                                                                                                                                             | RW     |

**Note**

When multiple area settings overlap, the priority is: Setting prohibited > setting enabled.

## 14 Keyboard Scan Control Module (KEYSCAN)

### 14.1 Introduction

This product is equipped with a keyboard scan control module (KEYSCAN) 1 unit. The KEYSCAN module supports keyboard array (row and column) scanning, the column is driven by an independent scan output KEYOUT<sub>m</sub> (m=0-7), and the row KEYIN<sub>n</sub> (n=0-15) is input as EIRQ<sub>n</sub> (n=0-15) is detected. This module realizes the key recognition function through the line scan query method.

### 14.2 Features

- EIRQ0-EIRQ15 can be independently selected as the row input of the keyboard array.
- KEYOUT can be selected by register.
- Output low level sequentially at regular intervals to scan the keyboard array.
- Scan time can be set.
- Stop scanning when the EIRQ interrupt is detected, and locate the pressed key according to the SSR.INDEX value and the EIRQ interrupt flag (INTC\_EIFR.EIF).

### 14.3.1 Function Block Diagram

**Figure 14-1 KEYSKAN System Block Diagram**

| Pin Name        | Direction | Description                   |
|-----------------|-----------|-------------------------------|
| KEYINn (n=0-15) | Input     | Keyboard row input signal     |
| KEYOUTm (m=0-7) | Output    | Keyboard column output signal |

Through the register SCR.KEYINSEL[15:0], KEYIN can be independently selected from EIRQ0-EIRQ15, and through the register SCR.KEYOUTSEL[2:0], the KEYOUT pin can be selected, so that the number of rows and columns of the keyboard can be flexibly selected. It can support a keyboard array of 16 rows\*8 columns.

### 14.3.3 Keyboard Scan Function

The keyboard scan function is: output low level to the columns of the keyboard array in a continuous period, so that when a key is pressed, a corresponding EIRQ interrupt flag will be generated.

When KEYSCAN\_SER.SEN is set to 1, KEYOUT0 outputs a low level, KEYOUT1-KEYOUTn (n is set by KEYSCAN\_SCR.KEYOUTSEL[2:0]) are Hi-Z, and after the time set by KEYSCAN\_SCR.T\_LLEVEL[4:0], all KEYOUT0-KEYOUTn pins are Hi-Z. After the time set by KEYSCAN\_SCR.T\_HIZ[2:0], KEYOUT1 outputs a low level, and the other KEYOUT pins are Hi-Z, and so on. When a key is pressed and an EIRQ interrupt flag is generated, the keyboard scanning function stops, and the scanning automatically restarts after the corresponding interrupt flag is cleared.

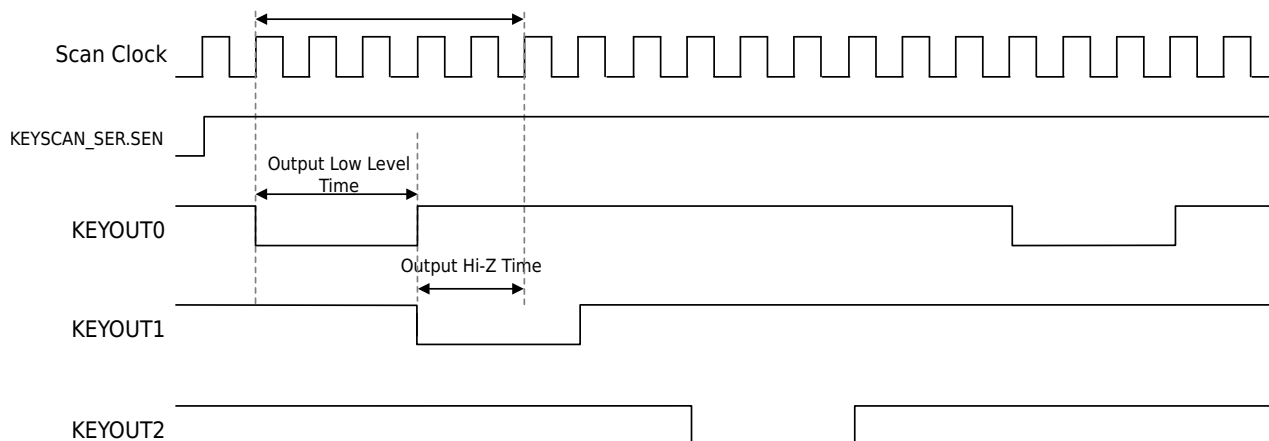


Figure 14-2 Keyboard Scan Function Diagram

## 14.4 Register Description

### 14.4.1 Register List

Base address: 0x40050C00

Table 14-2 Keyboard Scan Control Module (KEYSCAN) Register Offset

| Offset | Register    | Description           | Width |
|--------|-------------|-----------------------|-------|
| 0x00   | KEYSCAN_SCR | Scan Control Register | 32    |
| 0x04   | KEYSCAN_SER | Scan Enable Register  | 32    |
| 0x08   | KEYSCAN_SSR | Scan Status Register  | 32    |

## 14.4.2 Scan Control Register (KEYSCAN\_SCR)

**Note**

This register is only enabled when KEYCAN\_SER.SEN=0.

| Offset | Bit Position |    |          |    |    |    |          |    |       |    |          |    |           |    |          |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----------|----|----|----|----------|----|-------|----|----------|----|-----------|----|----------|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0x00   | 31           | 30 | 29       | 28 | 27 | 26 | 25       | 24 | 23    | 22 | 21       | 20 | 19        | 18 | 17       | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |          |    |    |    |          |    |       |    |          |    |           |    |          |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | T_HIZ        |    | T_LLEVEL |    |    |    | Reserved |    | CKSEL |    | Reserved |    | KEYOUTSEL |    | KEYINSEL |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|        |              |    |          |    |    |    |          |    |       |    |          |    |           |    |          |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access | RW           |    | RW       |    |    |    |          |    | RW    |    |          |    | RW        |    | RW       |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit   | Name     | Function                        | Description                                                                                                                                                                                                                                                                                                                                                                                            | Access |
|-------|----------|---------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:29 | T_HIZ    | Output Hi-Z time                | KEYOUT output Hi-Z time (number of scan clock) <ul style="list-style-type: none"> <li>0b000: 4 scan clocks</li> <li>0b001: 8 scan clocks</li> <li>0b010: 16 scan clocks</li> <li>0b011: 32 scan clocks</li> <li>0b100: 64 scan clocks</li> <li>0b101: 256 scan clocks</li> <li>0b110: 512 scan clocks</li> <li>0b111: 1024 scan clocks</li> </ul> Scan period=Output low level time + output Hi-Z time | RW     |
| 28:24 | T_LLEVEL | Output low level time           | KEYOUT output low level time (number of scan clocks) <p>Output low level time=2<sup>T_LLEVEL</sup> scan clocks</p> <p>Scan period=Output low level time + output Hi-Z time</p> <div> <b>Note</b><br/> 0b00000 and 0b00001 are set prohibited, and the maximum value that can be set is 0b11000. </div>                                                                                                 | RW     |
| 23:22 | Reserved | Reserved                        | -                                                                                                                                                                                                                                                                                                                                                                                                      | -      |
| 21:20 | CKSEL    | Scan clock source selection bit | <ul style="list-style-type: none"> <li>0b00: System clock HCLK</li> <li>0b01: Internal low-speed oscillator LRC</li> <li>0b10: External low-speed oscillator XTAL32</li> <li>0b11: Setting prohibited</li> </ul>                                                                                                                                                                                       | RW     |
| 19    | Reserved | Reserved                        | -                                                                                                                                                                                                                                                                                                                                                                                                      | -      |

| Bit   | Name      | Function                    | Description                                                                                                                                                                                                                                                                                                                 | Access |
|-------|-----------|-----------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 18:16 | KEYOUTSEL | KEYOUT output selection bit | <ul style="list-style-type: none"> <li>● 0b000: Setting prohibited</li> <li>● 0b001: KEYOUT0-KEYOUT1</li> <li>● 0b010: KEYOUT0-KEYOUT2</li> <li>● 0b011: KEYOUT0-KEYOUT3</li> <li>● 0b100: KEYOUT0-KEYOUT4</li> <li>● 0b101: KEYOUT0-KEYOUT5</li> <li>● 0b110: KEYOUT0-KEYOUT6</li> <li>● 0b111: KEYOUT0-KEYOUT7</li> </ul> | RW     |
| 15:0  | KEYINSEL  | Row input selection bit     | Corresponding to KEYINSEL15-KEYINSEL0, the selected row is used as the row of the keyboard array and is detected as EIRQn (n=15-0) : <ul style="list-style-type: none"> <li>● 0b0: Not used as rows of the keyboard array</li> <li>● 0b1: Used as a row of the keyboard array</li> </ul>                                    | RW     |

### 14.4.3 Scan Enable Register (KEYSCAN\_SER)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |     |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|-----|
| 0x04   | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |     |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |     |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   | SEN |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   | RW  |

| Bit  | Name     | Function        | Description                                                                                         | Access |
|------|----------|-----------------|-----------------------------------------------------------------------------------------------------|--------|
| 31:1 | Reserved | Reserved        | -                                                                                                   | -      |
| 0    | SEN      | Scan enable bit | <ul style="list-style-type: none"> <li>● 0b0: Scan disabled</li> <li>● 0b1: Scan enabled</li> </ul> | RW     |



#### 14.4.4 Scan Status Register (KEYSCAN\_SSR)


**Note**

The data read from this register is meaningful only when KEYCAN\_SER.SEN=1.

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |       |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|-------|
| 0x08   | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0     |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |       |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   | INDEX |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   | RO    |

| Bit  | Name     | Function                           | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | Access |
|------|----------|------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:3 | Reserved | Reserved                           | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | -      |
| 2:0  | INDEX    | Current working SCAN pin index bit | <ul style="list-style-type: none"> <li>0b000: The current working SCAN pin is KEY-OUT0</li> <li>0b001: The current working SCAN pin is KEY-OUT1</li> <li>0b010: The current working SCAN pin is KEY-OUT2</li> <li>0b011: The current working SCAN pin is KEY-OUT3</li> <li>0b100: The current working SCAN pin is KEY-OUT4</li> <li>0b101: The current working SCAN pin is KEY-OUT5</li> <li>0b110: The current working SCAN pin is KEY-OUT6</li> <li>0b111: The current working SCAN pin is KEY-OUT7</li> </ul> | RO     |

## 14.5 Precautions

This module drives the keyboard column, and the keyboard row detection is realized by the external EIRQ function of the interrupt control module (INTC). EIRQ needs to select the falling edge detection, and open the digital filter function, and set the appropriate filter time.

If you use this function in STOP mode, you need to set the scan-related parameters and select the internal low-speed oscillator LRC or the external low-speed oscillator XTAL32 clock as the scan clock.

If the internal pull-up resistor of the chip is used, please refer to the relevant description in the "Electrical Characteristics-I/O Port Characteristics" section of the "Datasheet" to select the appropriate scan time and filter time.

## 15 Clock Trimming Controller (CTC)

### 15.1 Introduction

Clock Trimming Controller (CTC) can automatically calibrate internal high-speed oscillator (HRC). Due to the influence of the working environment, the frequency of HRC may deviate. Based on the external high-precision reference clock, CTC is used to automatically adjust the frequency of HRC by hardware to get an accurate HRC clock.

### 15.2 Features

- Three external reference clock sources: XTAL, XTAL32, CTCREF
- 16-bit calibration counter for frequency measurement with reload function
- 8-bit calibration deviation value and 6-bit calibration value for frequency calibration
- Error interrupt for indicating calibration failure

### 15.3 Functional Description

#### 15.3.1 Function Block Diagram

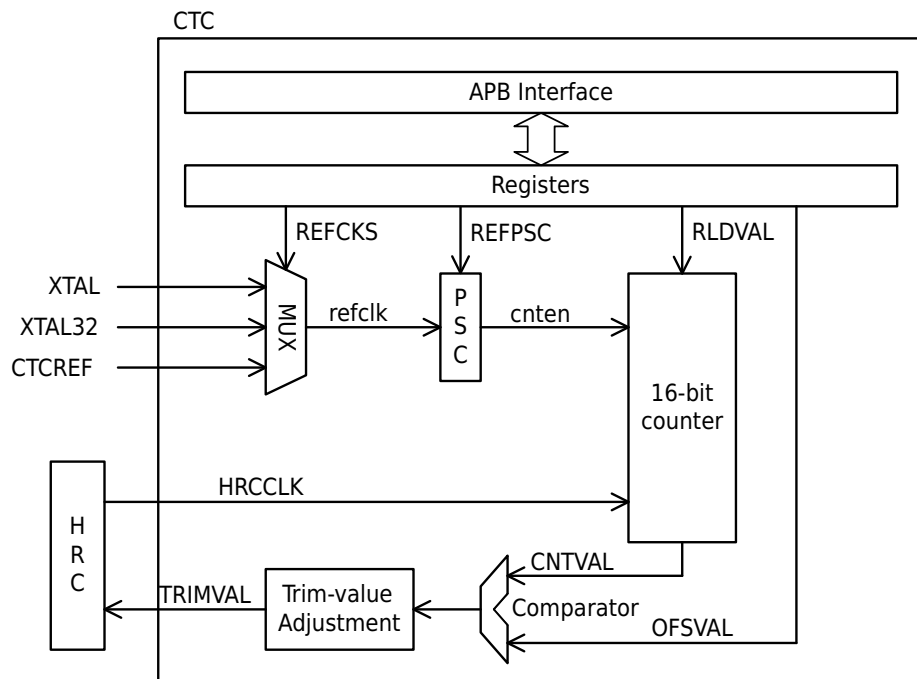


Figure 15-1 CTC Function Block Diagram

#### 15.3.2 Reference Clock

CTC has the following three reference clocks, selected through the CTC\_CR1.REFCKS bit.

- External high-speed clock (XTAL)
- External low-speed clock (XTAL32)
- External reference-speed clock (CTCREF)

Both the jitter of HRC and the counting error during calibration can affect the calibration accuracy. This can be reduced by setting the CTC\_CR1.REFPSC bit for the reference clock to select an appropriate prescaler. The error calculation formula is as follows:

$$TE = \frac{0.099\%}{\sqrt{(F_{hrc} \div F_{ref}) \times PSC_{ref} \div 10^5}} + \frac{4}{(F_{hrc} \div F_{ref}) \times PSC_{ref}}$$

Where:

- $F_{hrc}$  represents the target frequency of HRC, in Hz;
- $F_{ref}$  represents the frequency of the reference clock, in Hz;
- $PSC_{ref}$  represents the reference clock frequency division number.

The following two tables list the error values for different reference clock frequencies and prescaler conditions when the target frequency of HRC is 16MHz and 20MHz, respectively, for quick reference.



#### Note

- "Cannot be set" indicates that the setting exceeds the range of the calibration counter and calibration cannot be completed.
- "Not Recommended" indicates that the setting error exceeds 2%, so it is not recommended.

Table 15-1 Measurement Error for HRC Target Frequency of 20MHz

| $PSC_{ref}$ | $F_{ref}$     |                 |                 |                 |                 |
|-------------|---------------|-----------------|-----------------|-----------------|-----------------|
|             | 32.768kHz     | 4MHz            | 8MHz            | 12MHz           | 24MHz           |
| 8           | 0.530%        | Not Recommended | Not Recommended | Not Recommended | Not Recommended |
| 32          | 0.244%        | Not Recommended | Not Recommended | Not Recommended | Not Recommended |
| 128         | Cannot be set | 1.863%          | 3.000%          | Not Recommended | Not Recommended |
| 256         | Cannot be set | 1.188%          | 1.863%          | 2.453%          | Not Recommended |
| 512         | Cannot be set | 0.775%          | 1.188%          | 1.540%          | 2.453%          |
| 1024        | Cannot be set | 0.516%          | 0.775%          | 0.992%          | 1.540%          |
| 2048        | Cannot be set | 0.348%          | 0.516%          | 0.653%          | 0.992%          |
| 4096        | Cannot be set | 0.238%          | 0.348%          | 0.437%          | 0.653%          |

Table 15-2 Measurement Error for HRC Target Frequency of 16MHz

| $PSC_{ref}$ | $F_{ref}$     |                 |                 |                 |                 |
|-------------|---------------|-----------------|-----------------|-----------------|-----------------|
|             | 32.768kHz     | 4MHz            | 8MHz            | 12MHz           | 24MHz           |
| 8           | 0.603%        | Not Recommended | Not Recommended | Not Recommended | Not Recommended |
| 32          | 0.276%        | Not Recommended | Not Recommended | Not Recommended | Not Recommended |
| 128         | Cannot be set | 2.165%          | 3.519%          | Not Recommended | Not Recommended |

| PSC <sub>ref</sub> | F <sub>ref</sub> |        |        |        |                 |
|--------------------|------------------|--------|--------|--------|-----------------|
|                    | 32.768kHz        | 4MHz   | 8MHz   | 12MHz  | 24MHz           |
| 256                | Cannot be set    | 1.369% | 2.165% | 2.866% | Not Recommended |
| 512                | Cannot be set    | 0.887% | 1.369% | 1.784% | 2.866%          |
| 1024               | Cannot be set    | 0.587% | 0.887% | 1.140% | 1.784%          |
| 2048               | Cannot be set    | 0.395% | 0.587% | 0.746% | 1.140%          |
| 4096               | Cannot be set    | 0.269% | 0.395% | 0.497% | 0.746%          |

### 15.3.3 Frequency Calibration

After writing 1 to the CTCEN bit in the CTC\_CR1 register, the 16-bit calibration counter reloads RLDVAL from the CTC\_CR2 register and starts down-counting under the trigger of the reference clock, and the count clock is provided by HRC. After counting starts, if the reference clock triggers the count to stop, the 16-bit calibration counter will stop counting. Otherwise, the 16-bit calibration counter will continue counting and stop when the count value underflows to 0xFFFF.

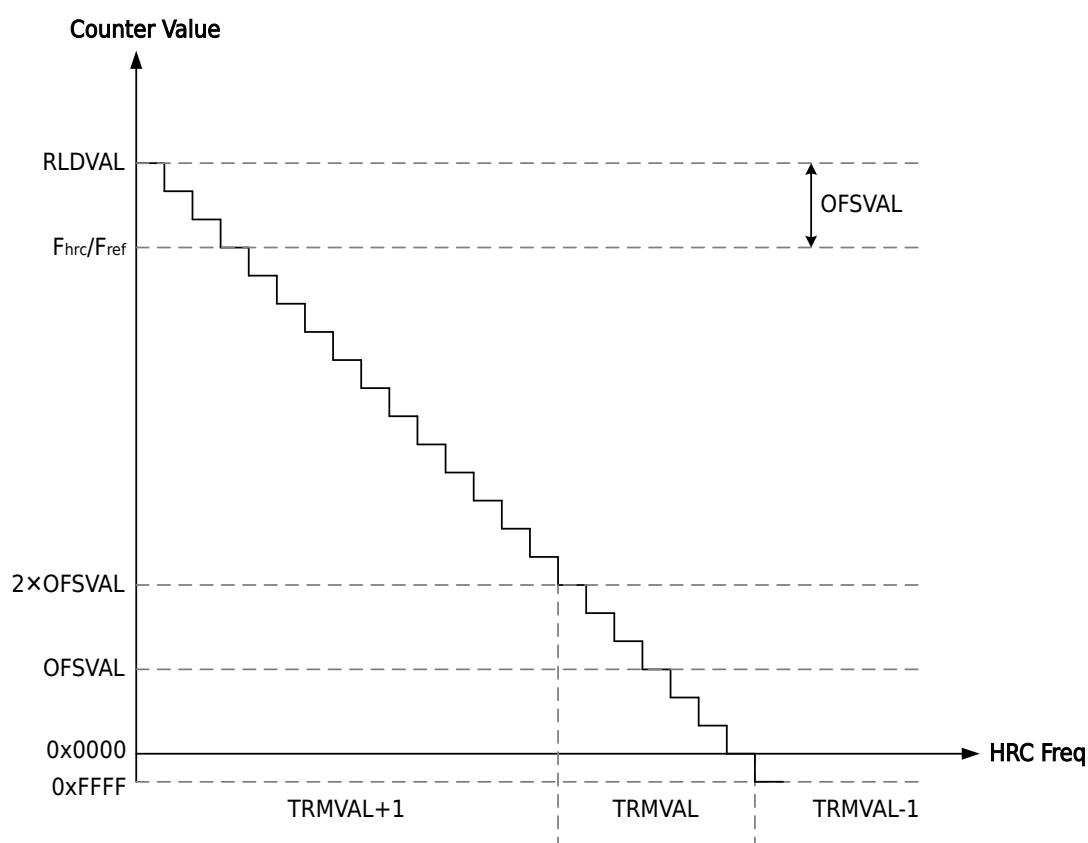


Figure 15-2 CTC Calibration Diagram

After the 16-bit calibration counter stops counting, HRC frequency calibration starts. Since the calibration target deviation value  $OFSVAL$  is added when setting the calibration count reload value RLDVAL, if the value of the 16-bit calibration counter is greater than  $2 \times OFSVAL$  and no underflow occurs, it indicates that the current frequency is slower than the desired frequency, and the value of  $TRMVAL$  in the CTC\_CR1 register needs to be increased to raise the frequency. Conversely, if the value of the 16-bit calibration counter underflows, it indicates that the current frequency is faster than the desired

frequency, and the value of TRMVAL needs to be decreased to lower the frequency. If the value of the 16-bit calibration counter is less than  $2 \times \text{OFSVAL}$ , it indicates that the current frequency is within the desired frequency range, and no adjustment to TRMVAL is required.

- When Counter > (OFSVAL $\times$ 2), counting stops, and the value of TRMVAL is automatically incremented by 1.
- When  $0 \leq \text{Counter} \leq (\text{OFSVAL} \times 2)$ , counting stops, the value of TRMVAL remains unchanged, and the CTC\_STR.TRIMOK bit automatically becomes 1.
- When Counter=0xFFFF, the value of TRMVAL in the CTC\_CR1 register is automatically decremented by 1.

After each adjustment of TRMVAL, the next calibration starts automatically. Calibration stops when CTCEN is cleared by software or a calibration error occurs.

### 15.3.4 Interrupt

The mid-value of TRMVAL is 0x00. If the result of automatically adding 1 exceeds 0x1F, the value of TRMVAL remains 0x1F, and a calibration overflow error occurs. At this time, the TRMOVF bit in the CTC\_STR register automatically becomes 1. Conversely, if the result of automatic subtraction by 1 is less than 0x20, the value of TRMVAL remains 0x20, and a calibration underflow error occurs. At this time, the TRMUFD bit in the CTC\_STR register automatically becomes 1. When a calibration overflow or underflow error occurs, calibration stops automatically (CTCEN is automatically cleared). If the ERRIE bit in the CTC\_CR1 register is 1, a calibration error interrupt request is generated. For specific setting methods, please refer to the "Interrupt Controller (INTC)" chapter.

## 15.4 Operation Example

### 15.4.1 Use XTAL32 as the Reference Clock to Calibrate HRC to 20MHz with a Calibration Target Deviation of Less Than 0.5%

Set XTAL32 to a 32-division prescaler. According to [Table 15-1 : Measurement Error for HRC Target Frequency of 20MHz](#), the calibration error is 0.244%, which is less than the 0.5% calibration target deviation, so the scheme is feasible.

1. Set register CTC\_CR2=0x4CAD0062, where:
  - $\text{OFSVAL} = (20 \div 0.032768) \times 32 \times 0.5\% \approx 98$  (0x62)
  - $\text{RLDVAL} = (20 \div 0.032768) \times 32 + 98 \approx 19629$  (0x4CAD)
2. Set register CTC\_CR1=0x000000E1, where:
  - TRMVAL=0x00 (Start calibration from the mid-value)
  - CTCEN=1 (Start CTC)
  - ERRIE=1 (Enable error interrupt)
  - REFCKS=0b10 (Select XTAL32 as the reference clock)
  - REFPSC=0b001 (Select a 32-division prescaler for the reference clock)
3. After a while, confirm the status of the CTC\_STR.TRIMOK bit:
  - If TRIMOK=1, write 0 to CTCEN to stop CTC;
  - If TRIMOK=0, continue waiting for a while and repeat this step.
4. Confirm that CTC\_STR.CTCBSY=0, CTC stops, and calibration is complete.

If an error interrupt occurs during the process, it indicates that a calibration overflow or underflow has occurred due to exceeding the calibratable range, CTCEN is automatically cleared, and calibration stops. At this time, please adjust OFSVAL, RLDVAL, and the reference clock settings and try calibration again.

**Note**

Frequency calibration is started or stopped through the CTC\_CR1.CTCEN bit. Therefore, after starting frequency calibration, do not change the settings of any other registers. To modify the settings, write the CTCEN bit to 0 and confirm that the CTC\_STR.CTCBSY bit is 0 before making the modification.

## 15.5 Register Description

### 15.5.1 Register List

Base Address: 0x40049C00

Table 15-3 Clock Trimming Controller (CTC) Register Offset

| Offset | Register | Description            | Width |
|--------|----------|------------------------|-------|
| 0x00   | CTC_CR1  | CTC Control Register 1 | 32    |
| 0x04   | CTC_CR2  | CTC Control Register 2 | 32    |
| 0x08   | CTC_STR  | CTC Status Register    | 32    |



## 15.5.2 CTC Control Register 1 (CTC\_CR1)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |        |    |    |    |    |    |          |    |    |    |    |    |   |   |   |   |       |       |        |          |        |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|--------|----|----|----|----|----|----------|----|----|----|----|----|---|---|---|---|-------|-------|--------|----------|--------|---|
| 0x00   | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21     | 20 | 19 | 18 | 17 | 16 | 15       | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5     | 4     | 3      | 2        | 1      | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |        |    |    |    |    |    |          |    |    |    |    |    |   |   |   |   |       |       |        |          |        |   |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    | TRMVAL |    |    |    |    |    | Reserved |    |    |    |    |    |   |   |   |   | CTCEN | ERRIE | REFCKS | Reserved | REFPSC |   |
| Access |              |    |    |    |    |    |    |    |    |    | RW     |    |    |    |    |    |          |    |    |    |    |    |   |   |   |   | RW    | RW    | RW     |          | RW     |   |

| Bit   | Name     | Function              | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | Access |
|-------|----------|-----------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:22 | Reserved | Reserved              | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | -      |
| 21:16 | TRMVAL   | HRC Calibration Value | <p>TRMVAL is a 6-bit signed calibration value with a median value of 0.</p> <ul style="list-style-type: none"> <li>● 0x20: -32</li> <li>● 0x21: -31</li> <li>● .....</li> <li>● 0x3F: -1</li> <li>● 0x00: 0 (Median value)</li> <li>● 0x01: +1</li> <li>● .....</li> <li>● 0x1E: +30</li> <li>● 0x1F: +31</li> </ul> <p>When the CTCEN bit is written to 1, TRMVAL becomes read-only and is automatically modified by hardware. To modify, write CTCEN to 0 and confirm CTC_STR.CTCBSY is 0 before making changes.</p> <p>In all cases, the final frequency calibration value of the HRC is the factory calibration value plus TRMVAL. Therefore, if the frequency under factory calibration meets requirements and no calibration is required, set TRMVAL to 0x00.</p> | RW     |
| 15:8  | Reserved | Reserved              | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | -      |
| 7     | CTCEN    | Calibration Enable    | <p>Used to start frequency calibration.</p> <ul style="list-style-type: none"> <li>● 0b0: Stops frequency calibration.</li> <li>● 0b1: Starts frequency calibration.</li> </ul> <div>  <b>Note</b> <ul style="list-style-type: none"> <li>● Setting Conditions: Set by software.</li> <li>● Clearing Conditions: <ul style="list-style-type: none"> <li>▶ Software Clear</li> <li>▶ Automatically clears on calibration overflow or underflow.</li> </ul> </li> </ul> </div>                                                                                                                                                                                                         | RW     |

| Bit | Name            | Function                            | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | Access |
|-----|-----------------|-------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 6   | ERRIE           | Error Interrupt Enable              | Used to select whether an error interrupt is generated on calibration overflow or underflow.<br><ul style="list-style-type: none"> <li>● 0b0: Disables error interrupt.</li> <li>● 0b1: Enables error interrupt.</li> </ul>                                                                                                                                                                                                                                                                                        | RW     |
| 5:4 | REFCKS          | Reference Clock Selection           | Used to select the reference clock source for frequency calibration.<br><ul style="list-style-type: none"> <li>● 0b00/0b01: Selects CTCREF as the reference clock.</li> <li>● 0b10: Selects XTAL32 as the reference clock .</li> <li>● 0b11: Selects XTAL as the reference clock .</li> </ul> When CTCEN is 1, modifying the value of REFCKS is prohibited. To modify, write CTCEN to 0 and confirm CTC_STR.CTCBSY is 0 before making changes.                                                                     | RW     |
| 3   | <i>Reserved</i> | <i>Reserved</i>                     | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | -      |
| 2:0 | REFPSC          | Reference Clock Prescaler Selection | Used to select the prescaler for the reference clock.<br><ul style="list-style-type: none"> <li>● 0b000: Divide by 8</li> <li>● 0b001: Divide by 32</li> <li>● 0b010: Divide by 128</li> <li>● 0b011: Divide by 256</li> <li>● 0b100: Divide by 512</li> <li>● 0b101: Divide by 1024</li> <li>● 0b110: Divide by 2048</li> <li>● 0b111: Divide by 4096</li> </ul> When CTCEN is 1, modifying the value of REFPSC is prohibited. To modify, write CTCEN to 0 and confirm CTC_STR.CTCBSY is 0 before making changes. | RW     |

## 15.5.3 CTC Control Register 2 (CTC\_CR2)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |          |    |    |    |    |    |   |   |        |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----------|----|----|----|----|----|---|---|--------|---|---|---|---|---|---|---|
| 0x04   | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15       | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7      | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |          |    |    |    |    |    |   |   |        |   |   |   |   |   |   |   |
| Name   | RLDVAL       |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | Reserved |    |    |    |    |    |   |   | OFSVAL |   |   |   |   |   |   |   |
| Access | RW           |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |          |    |    |    |    |    |   |   | RW     |   |   |   |   |   |   |   |

| Bit   | Name     | Function                          | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | Access |
|-------|----------|-----------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:16 | RLDVAL   | Calibration Count Reload Value    | <p>RLDVAL sets the reload value for the 16-bit calibration counter.</p> $RLDVAL = (F_{hrc} \div (F_{ref} \div PSC_{ref})) + OFSVAL$ <ul style="list-style-type: none"> <li>● <math>F_{hrc}</math>: HRC target frequency;</li> <li>● <math>F_{ref}</math>: Reference clock frequency;</li> <li>● OFSVAL: Calibration count deviation value;</li> <li>● <math>PSC_{ref}</math>: Reference clock prescaler.</li> </ul> <div>  <b>Note</b><br/>           When the CTCEN bit is 1, modifying RLDVAL is prohibited. To modify, write CTCEN to 0 and confirm CTC_STR.CTCBSY is 0 before making changes.         </div> | RW     |
| 15:8  | Reserved | Reserved                          | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | -      |
| 7:0   | OFSVAL   | Calibration Count Deviation Value | <p>OFSVAL sets the allowed deviation value for HRC during calibration.</p> $OFSVAL = (F_{hrc} \div (F_{ref} \div PSC_{ref})) \times TA$ <ul style="list-style-type: none"> <li>● <math>F_{hrc}</math>: HRC target frequency;</li> <li>● <math>F_{ref}</math>: Reference clock frequency;</li> <li>● TA: Calibration accuracy, e.g., 0.5%;</li> <li>● <math>PSC_{ref}</math>: Reference clock prescaler.</li> </ul> <div>  <b>Note</b><br/>           When the CTCEN bit is 1, modifying OFSVAL is prohibited. To modify, write CTCEN to 0 and confirm CTC_STR.CTCBSY is 0 before making changes.         </div> | RW     |

## 15.5.4 CTC Status Register (CTC\_STR)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |        |        |        |        |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|--------|--------|--------|--------|
| 0x08   | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |        |        |        |        |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |        |        |        |        |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   | CTCBSY | TRMUDF | TRMOVF | TRIMOK |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   | RO     | RO     | RO     | RO     |

| Bit  | Name     | Function                   | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                               | Access |
|------|----------|----------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:4 | Reserved | Reserved                   | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | -      |
| 3    | CTCBSY   | CTC Status Flag            | <ul style="list-style-type: none"> <li>0b0: CTC is stopped</li> <li>0b1: CTC is operating</li> </ul> When CTC_CR1.CTCEN=1, this bit is set and cleared when CTCEN is cleared by software or a calibration error occurs.                                                                                                                                                                                                                                                   | RO     |
| 2    | TRMUDF   | Calibration Underflow Flag | Calibration underflow indicates that the TRMVAL has been decremented to 0x20, yet the calibration has still not achieved the target range. When calibration underflow occurs, the calibration process is automatically stopped, and an error interrupt will be triggered if ERRIE is set to 1. <ul style="list-style-type: none"> <li>0b0: No calibration underflow.</li> <li>0b1: Calibration underflow occurred.</li> </ul> Cleared when CTC_CR1.CTCEN is written to 1. | RO     |
| 1    | TRMOVF   | Calibration Overflow Flag  | Calibration overflow indicates that the TRMVAL has been incremented to 0x1F, yet the calibration has still not achieved the target range. When calibration overflow occurs, the calibration process is automatically stopped, and an error interrupt will be triggered if ERRIE is set to 1. <ul style="list-style-type: none"> <li>0b0: No calibration overflow.</li> <li>0b1: Calibration overflow occurred.</li> </ul> Cleared when CTC_CR1.CTCEN is written to 1.     | RO     |
| 0    | TRIMOK   | Calibration Success Flag   | <ul style="list-style-type: none"> <li>0b0: HRC frequency has not been calibrated to the target range.</li> <li>0b1: HRC frequency has been calibrated to the target range.</li> </ul> Cleared when CTC_CR1.CTCEN is written to 1.                                                                                                                                                                                                                                        | RO     |

## 16 DMA Controller (DMA)

### 16.1 Introduction

DMA is used to transfer data between memory and peripheral function modules, enabling data exchange between memories, between memory and peripheral function modules, and between peripheral function modules without CPU involvement.

### 16.2 Features

- The DMA bus is independent of the CPU bus and transmits data according to the AMBA AHB-Lite bus protocol.
- With 2 DMA control units, a total of 16 independent channels, which can independently operate different DMA transfer functions.
- The start source of each channel is configured through an independent trigger source selection register.
- 1 block per request.
- The data block is a minimum of one data and can be up to 1024 data.
- The width of each data item can be configured as 8-bit, 16-bit, or 32-bit.
- It can be configured for 1-65535 transfers or infinite transfers.
- The source address and destination address can be independently configured as fixed, incrementing, decrementing, repeated jump, or specified offset jump.
- 3 types of interrupts can be generated: block transfer completed interrupt, transfer completed interrupt, and transfer error interrupt. Each interrupt can be configured to be masked or unmasked. Among them, block transfer completed and transfer completed can be used as event outputs, serving as trigger sources for other peripheral modules.
- Support chain transfer function, enabling the transfer of multiple data blocks in a single request.
- Support channel reset triggered by external events.
- Support software-triggered start of transfer and software-triggered channel reset.
- When not in use, it can be set to stop state to reduce power consumption.
- The HPROT value in the AHB bus during DMA access can be set through a register.

## 16.3 Functional Description

### 16.3.1 DMA Block Diagram

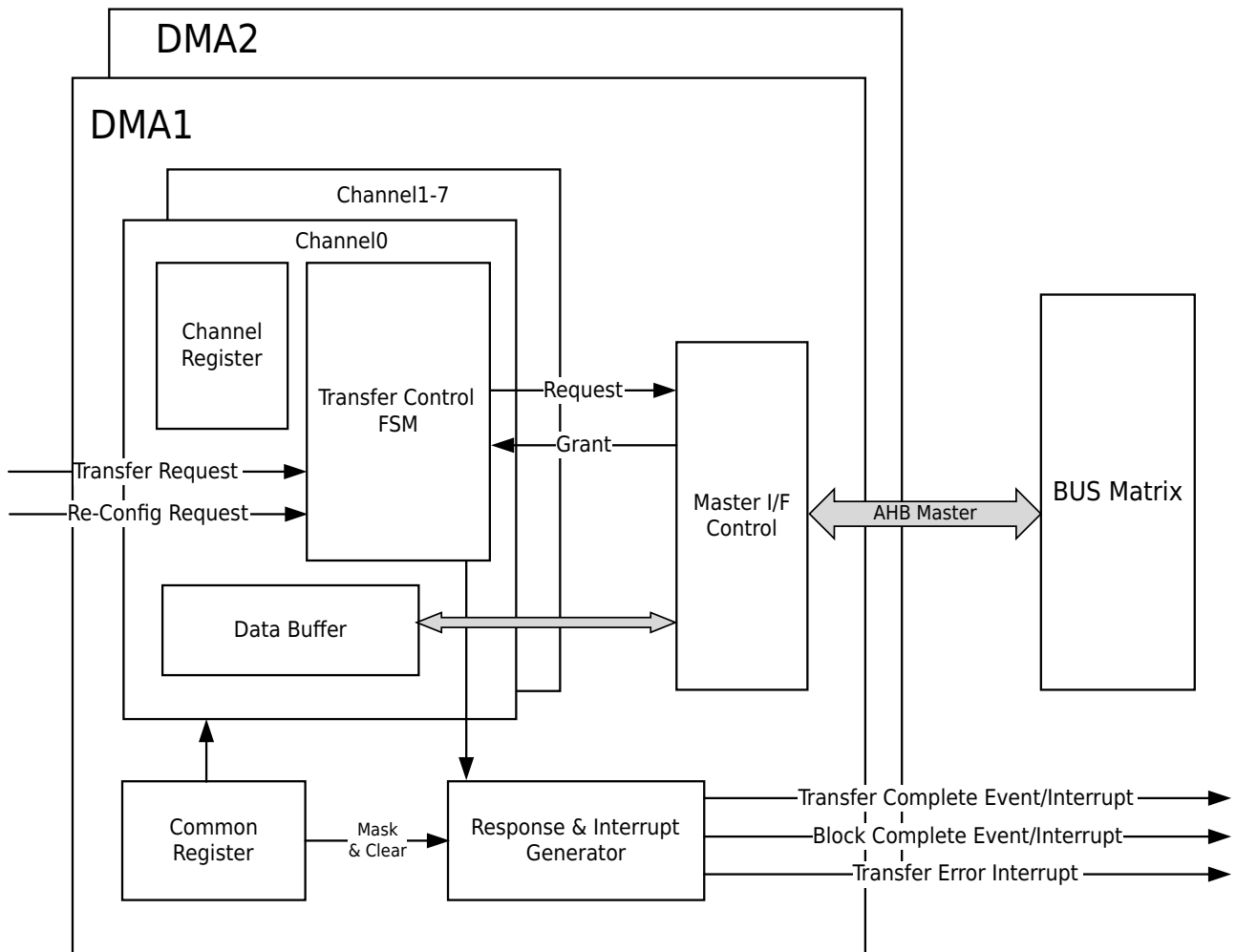


Figure 16-1 DMA Architecture Diagram

### 16.3.2 Enabling the DMA Controller

To use DMA, you need to first write 0 to the FCG0.DMAx bit to allow DMA clock supply, and then write 1 to the DMA\_EN.EN bit. At the same time, DMA requires peripheral circuits to initiate transfers, so the FCG0.AOS peripheral circuit trigger function bit also needs to be written to 0.

When DMA is not in use or the chip needs to enter STOP mode, please set DMA\_EN.EN to 0 and write to FCG0 to turn off the DMA clock. Before setting EN to 0, please confirm that DMA\_CHSTAT.DMAACT is 0 to ensure that DMA has completed all transfers.

### 16.3.3 Channel Selection and Channel Priority

Each DMA control unit contains 8 channels, and each channel can configure the transfer function independently.

The channel priority order is: Channel 0 > Channel 1 >... > Channel 7.

When multiple channels in a DMA unit request transfer, they will be executed in priority order. But the channel that is already in transit will not be interrupted, the high priority channel will not start until the current channel transmission is complete.

### 16.3.4 Start DMA

DMA is started by requests generated by peripheral circuits, which are configured through the trigger source selection register (DMA\_TRGSELx, where x=0-7) for each channel independently. DMA can be started by writing to the DMA\_SWREQ.SWREQ register for software-triggered transfer. When a peripheral circuit generates a start request or software writes to a register to generate a start request, and DMA transfer enable is active (DMA\_EN.EN=1), and the transfer channel is in the permitted state (DMA\_CHEN.CHEN[x]=1), channel x transfer starts.

Before use, you need to set the peripheral circuit trigger function and DMA function enable bits in FCGO register to active.

### 16.3.5 Data Block

The amount of data transferred each time DMA is started is expressed in blocks. The size of the block is set by register DMA\_DTCTLx.BLKSIZE, and the maximum size that can be set is 1024 data. The data width of each is determined by DMA\_CHCTLx.HSIZE and can be 8-bit, 16-bit, or 32-bit.

### 16.3.6 Transfer Address Control

The source address and destination address of the transfer can be set to fixed, incrementing, decrementing, repeated jump, or nonSequence jump through registers.

- **Fixed:** The source address and destination address will be fixed during the transmission process.
- **Incrementing and Decrementing:** The source address and destination address increase or decrease by the value of HSIZE after each data item transfer. For example, when HSIZE is 8-bit, the address increases/decreases by 1 each time; for 16-bit, it increases/decreases by 2; for 32-bit, it increases/decreases by 4.
- **Repeating Jump:** After transferring a specified number of data items, the source and destination addresses return to their initial values. The amount of data that needs to be transferred before the address repeats the jump, that is, the size of the repeated area is set by the size of the register DMA\_RPT.
- **NonSequence Jump:** After transferring a specified number of data items, the source and destination addresses skip a specified offset. The offset for address jump and the amount of data to be transferred before jump, i.e., the size of the nonSequence zone, are set by the DMA\_SNSEQCTL/DMA\_DNSEQCTL registers. When the conditions for address repeated jump and nonSequence jump are met simultaneously, address repeated jump is executed.

### 16.3.7 Transfer Count

The total number of data blocks transferred by DMA is set by the DMA\_DTCTLx.CNT bit. The maximum number of transmissions can be set to 65535. Each time a data block is transferred, the register value decrements by 1. When the register value becomes 0, it indicates that all data transfer for the channel is completed, and the channel transfer permission bit (DMA\_CHEN.CHEN[x]) is automatically cleared, generating a transfer completed interrupt. If DMA\_DTCTLx.CNT is set to 0 at the beginning of the transfer, it means unlimited transmission, and each start requests to transmit a data block, but the channel transmission permission bit is not cleared, and the transmission completion interrupt will not be generated.

**Note**

After CHEN[x] is automatically cleared, to restart the channel, you need to reconfigure the channel configuration register and then write 1 to CHEN[x] to enable the channel. Otherwise, the subsequent transfer will start from the state at the end of the last transfer, i.e., the source/destination address will be the next address corresponding to the last transfer, and CNT will be 0 for unlimited transfer.

### 16.3.8 Chain Transmission

DMA controller has the function of linkage transmission. The chain transmission needs to configure the following 8 registers, a total of 8 words, called a descriptor which contains the source address, destination address, data control information, address control information, chain pointer and transmission control information of the chain transmission.

- DMA\_SARx
- DMA\_DARx
- DMA\_DTCTLx
- DMA\_RPTx
- DMA\_SNSEQCTLx
- DMA\_DNSEQCTLx
- DMA\_LL Px
- DMA\_CHCTLx

Where LLP is called a Linked-List Pointer where the value represents the next descriptor's first address in the memory. When using chained transfer, first write the register DMA\_CHCTLx.LLPEN to enable chained transfer, and write the descriptor information of the first transfer into the corresponding register. Then initialize the descriptors for subsequent transfers sequentially in memory. To end chain transfer, set DMA\_CHCTLx.LLPEN to inactive in the last descriptor, and the DMA controller will end chain transfer after transfer completion.

When the last transfer of a descriptor completes, the next descriptor specified by LLP is loaded into the channel configuration register. Wait for the next transfer request input to start the first transfer of the new descriptor. Or according to register DMA\_CHCTLx.LLPRUN settings, starting the first transfer directly after loading the new descriptor.

- When DMA\_CHCTLx.LLPRUN=0, after the last block transfer of the current descriptor, the new channel descriptor is loaded into the channel register. BTC and TC interrupts are generated as usual, and then the DMA waits for the next transfer request to start transferring the newly loaded descriptor.
- When DMA\_CHCTLx.LLPRUN=1, the transfer of the new descriptor starts immediately after loading, and neither BTC nor TC interrupt is generated for the last block transfer of the old descriptor.



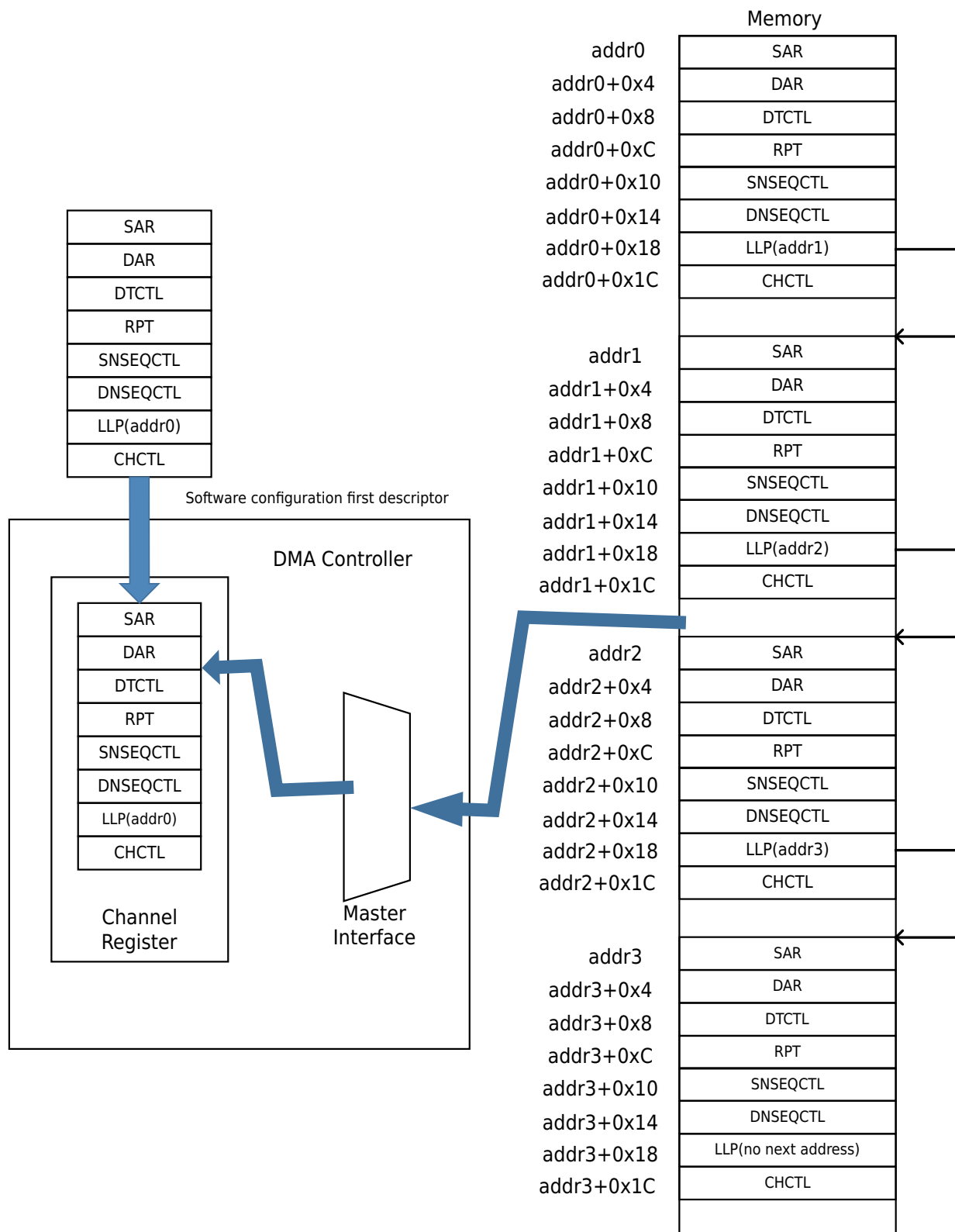


Figure 16-2 Chain Transmission Diagram

### 16.3.9 NonSequence Address Transfer

Using nonSequence address transmission can realize source address and destination address to jump according to a certain offset after a certain amount of data is transmitted. Taking source address NonSequence as an example: First, set DMA\_CHCTLx.SNSEQEN to 1 to enable nonSequence address transfer. The jump direction is set forward or backward according to DMA\_CHCTLx.SINC, and then configure DMA\_SNSEQCTLx to set the specific jump mode. The control of the destination address is

similar to that of the source address, with corresponding control registers being DMA\_CHCTLx.DNSEQEN, DMA\_CHCTLx.DINC, and DMA\_DNSEQCTLx. The transfer process proceeds as shown in the following diagram.

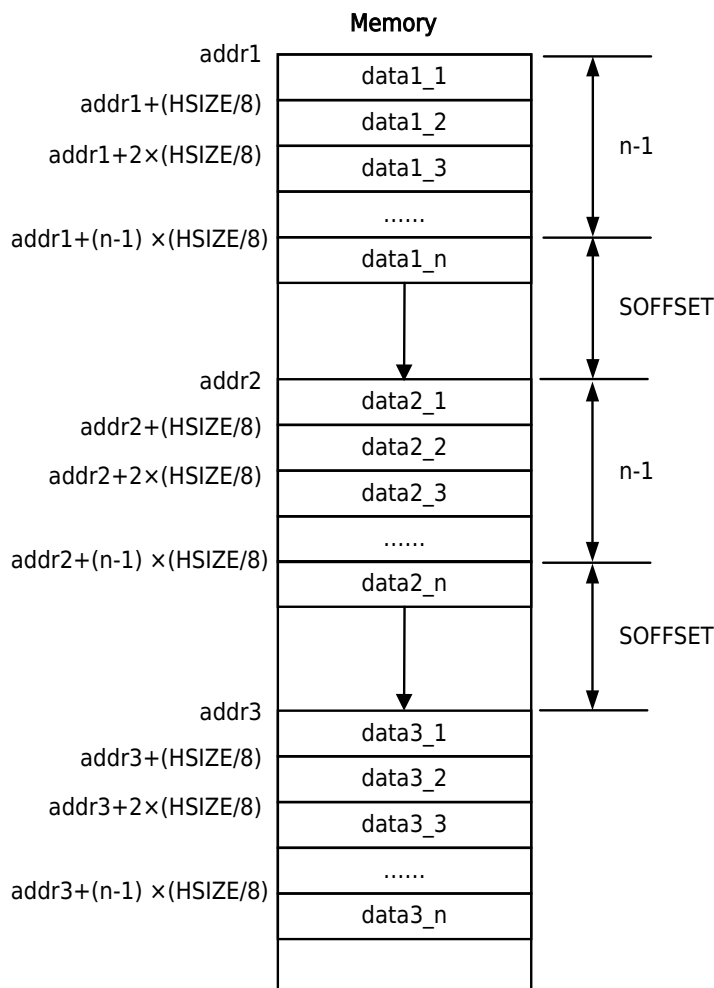


Figure 16-3 Schematic Diagram of NonSequence Address Transfer (Taking source address as an example)

$n = SNSCNT$

$addr2 = addr1 + (n-1) \times (HSIZE/8) + SOFFSET \times (HSIZE/8)$

$addr3 = addr2 + (n-1) \times (HSIZE/8) + SOFFSET \times (HSIZE/8)$

Where HSIZE is the data width (bit), which can be 8 bits (byte), 16 bits (half-word), or 32 bits (word).

### 16.3.10 Channel Reset

The channel reset function refers to modifying the internal status register of the channel through the event request of the peripheral circuit, and reconfiguring the next data transmission mode. Set register DMA\_RCFGCTL.RCFGGEN to 1 to allow channel reset. Select the reset request source through the DMA\_RC\_TRGSEL register or trigger channel reset through software by writing to the DMA\_SWREQ.SWRCFGREQ register. When the selected reset request source input or software writing to the register generates a reset request, the channel selected by DMA\_RCFGCTL.RCFGCHS is updated according to the specified mode. The reset request only updates the internal state and does not start actual data transfer.

There are 3 ways to reset channels: linked list pointer, nonSequence, and repeated.

- When the linked list pointer reset is selected, the descriptor and internal state of the channel are all updated to the new descriptor pointed to by the linked list pointer LLP. Subsequent transfer requests are transferred according to the new descriptor.
- When selecting nonSequence or repeated reset, the channel internal state is updated as described in the following table.

Table 16-1 Channel Reset Description

| Reset Modes ( DMA_RCFGCTL) |           |           | Channel internal state                                                                          |                                                                                     |                                                                                 |
|----------------------------|-----------|-----------|-------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|---------------------------------------------------------------------------------|
| CNTMD                      | SARMD     | DARMD     | Remaining Transfers Counter                                                                     | Source address of next transmission                                                 | Destination address of next transmission                                        |
| 0x0                        | Arbitrary | Arbitrary | Remains unchanged                                                                               | Updated based on SARMD                                                              | Updated based on DARMD                                                          |
| 0x1                        | 0x0       | Arbitrary | Remains unchanged                                                                               | Remains unchanged                                                                   | Updated based on DARMD                                                          |
|                            | 0x1       |           | Updated to the value after the next nonSequence jump in the source address in normal state      | Updated to the start address of the next source address nonSequence transfer region |                                                                                 |
|                            | 0x2, 0x3  |           | Updated to the value after the next repeated jump in the source address in normal state         | Updated to the initial setting in the DMA_SARx register                             |                                                                                 |
| 0x2, 0x3                   | Arbitrary | 0x0       | Remains unchanged                                                                               | Updated based on SARMD                                                              | Remains unchanged                                                               |
|                            |           | 0x1       | Updated to the value after the next nonSequence jump in the destination address in normal state |                                                                                     | Updated to the start address of the next destination address NonSequence region |
|                            |           | 0x2, 0x3  | Updated to the value after the next repeated jump in the destination address in normal state    |                                                                                     | Updated to the initial setting in the DMA_DARx register                         |

**Note**

When the reset function is enabled, the channel uses DMA\_RPTBx and DMA\_SNSEQCTLBx, DMA\_DNSEQCTLBx to control repeated jump and nonSequence jump of transfer addresses. Register DMA\_RPTx and DMA\_SNSEQCTLx, DMA\_DNSEQCTLx are inactive.

In the DMA action shown in the following diagram, each transfer request starts the transfer of one data block. After the first reset request, the controller skips data blocks blk1\_3 and blk1\_4, and the transfer address is updated to the starting address of the next nonSequence zone, i.e., addr2. After the second reset request occurs, the remaining number of transfers is updated to 0, that is, all data transfers are completed, the channel permission bit is automatically cleared to 0, and a transfer completion interrupt and event are generated.

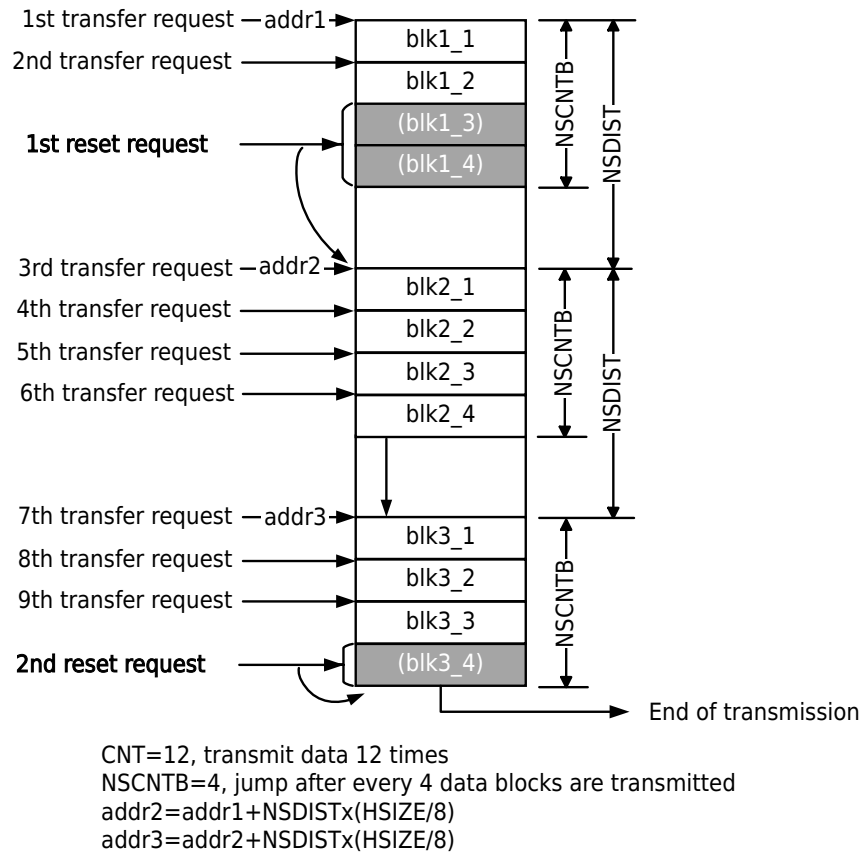


Figure 16-4 Schematic Diagram of NonSequence Reset

### 16.3.11 Early Termination of Transfer

During transfer, the DMA\_CHEN.CHENx bit remains active. For non-chain transfers, the transfer count set in register DMA\_DTCTLx completes automatically and sets the bit to inactive. For chain transfers, the transfer count of the last chain transfer completes automatically and sets the bit to inactive. During transfer, if software writes 1 to DMA\_CHENCLR.CHENCLR<sub>x</sub>, DMA will terminate the transfer after completing the current data read and write, and the DMA\_CHEN.CHENx bit is cleared.

#### Notice

When software writes 1 to the CHENCLR<sub>x</sub> bit to terminate the transfer early, DMA does not save the transfer state at the time of termination. Without resetting the channel configuration register (descriptor) state and setting CHENx to 1 to re-enable the channel, DMA will retransmit the terminated data block instead of resuming from the breakpoint after a transfer request input.

### 16.3.12 Interrupt

The DMA controller can generate the following three types of interrupts:

- **Data Block Complete Interrupt DMA\_BTCx**: generated after completing a data block transfer.
- **Transmission Completion Interrupt DMA\_TCx**: generated after completing the transfer count set by DMA\_DTCTLx. CNT.
- **Transmission Error Interrupt DMA\_ERR**: when the start request overflows (that is, the channel triggers the start request again when the last request of the channel has not been responded), or when a bus error occurs during the transmission (such as accessing an illegal address or a protected address) an interrupt is generated, where in a bus error terminates the transfer immediately.

Except for the transfer request overflow error, all other interrupts can be enabled or disabled through the DMA\_CHCTLx.IE register. Additionally, all interrupts have independent MASK registers to mask the interrupts.

The above-mentioned DMA\_BTCx and DMA\_TCx interrupts can also be used as event signal outputs and can be used as trigger sources for other peripheral circuits. The event output is controlled by the MASK register but not by the DMA\_CHCTLx.IE bit. DMA\_BTCx, DMA\_TCx, and DMA\_ERR events will set the corresponding status bits in the register, regardless of the DMA\_CHCTLx.IE or MASK register settings.

## 16.4 Operation Example

### 16.4.1 Memory-to-Memory Transfer

#### Target

Transfer 12 data items from RAM address 0x20000000 to 0x20001000, with a data width of 32-bit.

#### Register Setting

1. Enable the DMA controller by writing 1 to DMA\_EN.EN.
2. Select a channel, for example channel 0, and configure the channel registers to achieve:
  - a. Write DMA\_SAR0 to configure the source address as SRAM region 0x20000000.
  - b. Write DMA\_DAR0 to configure the destination address as SRAM region 0x20001000.
  - c. Write the DMA\_DTCTL0 configuration data block size is 4, the number of transfers is 3, after each transfer one data block to produce block transfer complete interrupt, after 3 transfers complete one transfer complete interrupt
  - d. Write DMA\_RPT register configures the repeating area of the source address to be 6, that is, the original source address is reloaded after 6 addresses have been transmitted.
  - e. Configure the channel control registers DMA\_CHCTL0 to achieve:
    - 1) The source address and destination address chained transfer is disabled.
    - 2) Source address repeat enabled, destination address increment.
    - 3) The data width is 32-bit.
    - 4) Interrupt enable active.
  - f. Configure the trigger source controller DMA\_TRGSEL0 to select software trigger as the start request for DMA channel 0.
  - g. Write 1 to the channel enable bit DMA\_CHEN.CHEN0 to enable channel 0.
3. Write to the peripheral event software trigger register INTSFTTRG.STRG to 1 to send the first software start request, and DMA starts transferring data.

#### Transfer Process

Since the size of the data block is 4, the first transmission begins when INTSFTTRG.STRG is 1. When a data block transfer is completed, the transfer count DMA\_MONDTCTL0.CNT is decremented by 1, and produces a block transport complete interrupt, the software can continue to write INTSFTTRG.STRG in the interrupt subroutine to start the second transport. During the second transfer, due to the source address repeat zone size of 6, after transferring 2 addresses, the source address will reload the initial address 0x20000000, and continue to transfer the remaining 2 addresses. After the second transfer is completed, the number of transfers DMA\_MONDTCTL0.CNT is reduced by 1, and a block transfer completion interrupt is generated, and the software can continue to write INTSFTTRG.STRG in the interrupt subroutine to start the third transfer. After the third transfer is completed, the number of transfers DMA\_MONDTCTL0.CNT is reduced to 0, that is, this transfer is all completed, DMA generates

a block transfer completion interrupt and a transfer completion interrupt, and the channel enable bit DMA\_CHEN.CHEN0 will be automatically cleared.

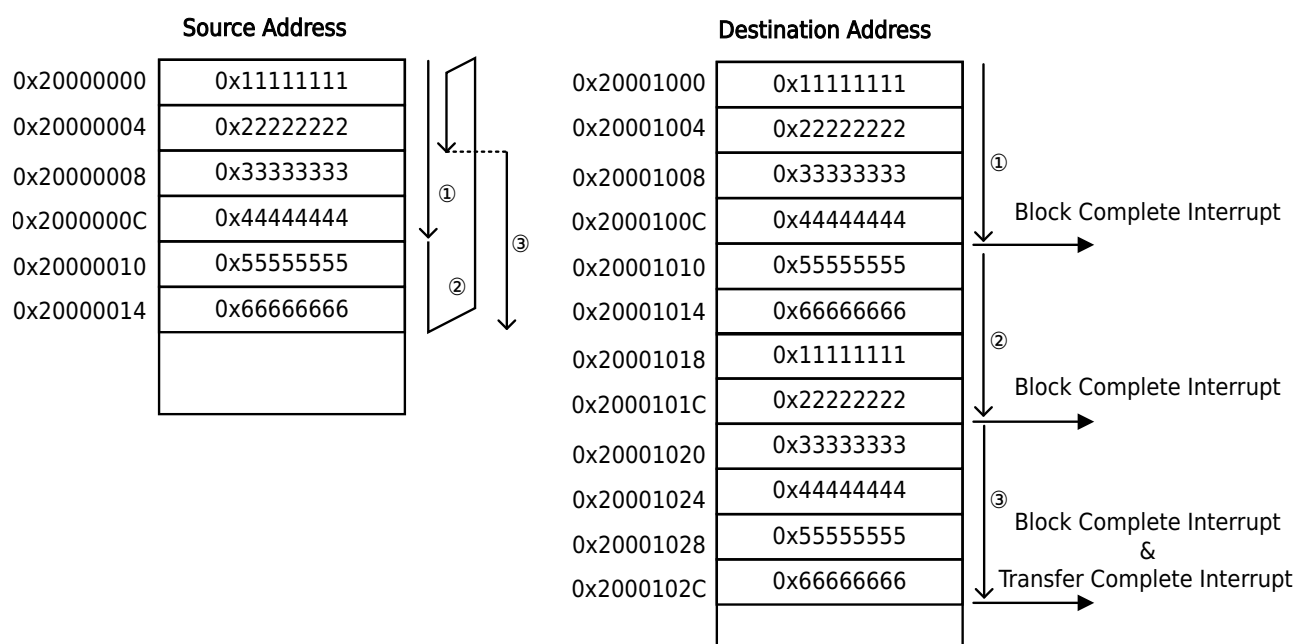


Figure 16-5 Application Example 1: Memory-to-Memory Transfer

## 16.4.2 Memory-to-Peripheral Transfer

### Target

Transfer 10 data items of 16-bit width from RAM address 0x20000000 to the transmit buffer register of a communication module. The communication module generates a transfer request after sending each data item. After the last data item is sent, DMA generates a transfer completed interrupt.

### Register Setting

1. Enable the DMA controller by writing 1 to DMA\_EN.EN.
2. Configure DMA\_INTMSK1 to mask the block transfer completed interrupt and enable the transfer completed interrupt.
3. Select a channel, for example channel 0, and configure the channel registers to achieve:
  - a. Write DMA\_SAR0 to configure the source address as SRAM region 0x20000000.
  - b. Write DMA\_DAR0 configure destination address to register address of peripheral circuit 0x40000000.
  - c. Write DMA\_DTCTL0 configuration data block size is 1, number of transfers is 10, each transfer request is delivered once, one data at a time.
  - d. Configure the channel control registers DMA\_CHCTL0 to achieve:
    - 1) The source address and destination address chained transfer is disabled.
    - 2) Source address increment, destination address fixed.
    - 3) The source address and destination address data width is 16-bit.
    - 4) Interrupt enable active.
  - e. Configure the trigger source controller DMA\_TRGSEL0 to select the transmit buffer empty status of the communication module as the start request for DMA channel 0.
  - f. Write 1 to the channel enable bit DMA\_CHEN.CHEN0 to enable channel 0.

## Transfer Process

After the channel is enabled, DMA waits for the transmission request from the communication module. After the transmission request is generated, the DMA transmits the data in the RAM to the transmission buffer register of the communication module and waits for the second transmission request from the communication module. Because the block transmission is shielded to complete interrupt, the DMA does not generate interrupt request at this time. When all 10 data transfers are completed, DMA generates a transfer completion interrupt, and the channel enable bit DMA\_CHEN.CHEN0 will be automatically cleared.

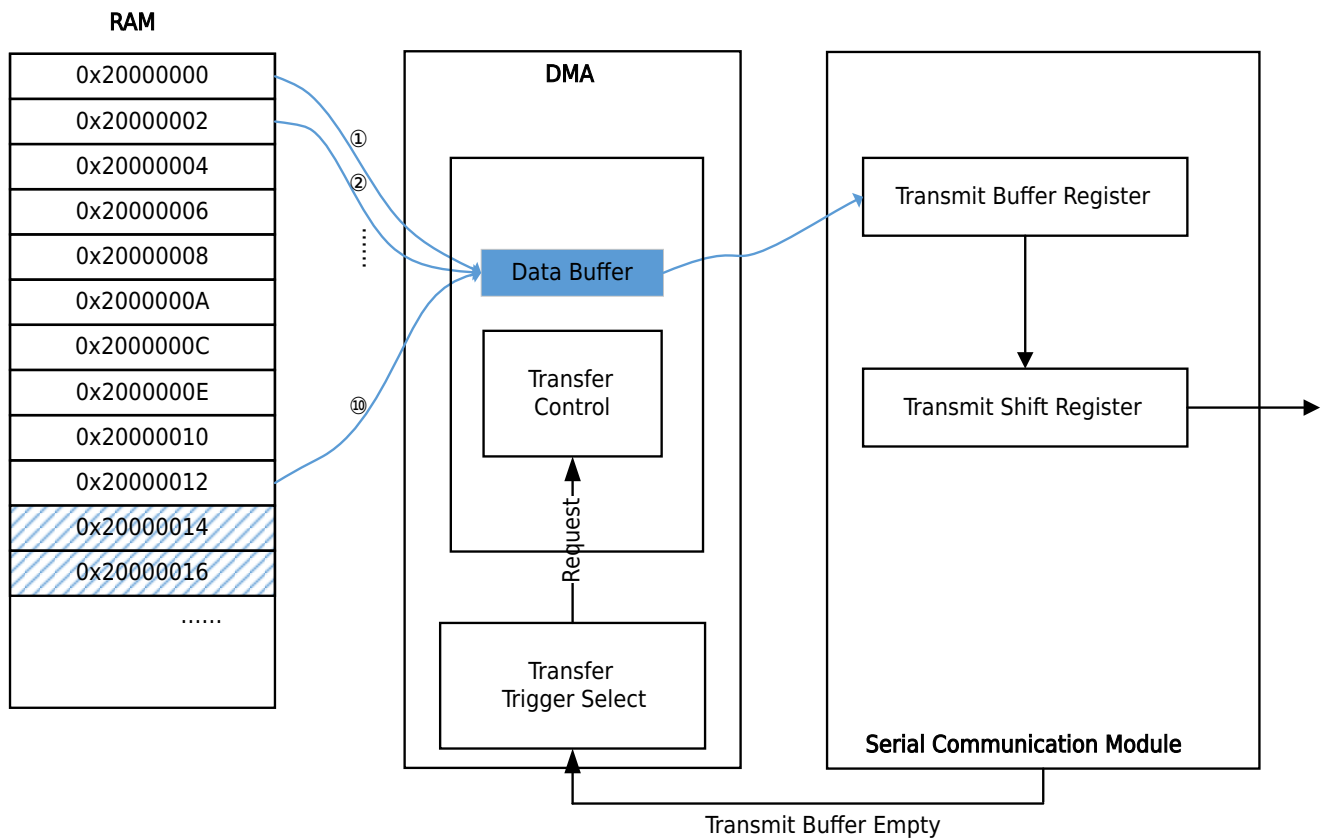


Figure 16-6 Application Example 2: Memory-to-Peripheral Transfer

### 16.4.3 Memory-to-Memory Chain Transfer

#### Target

Perform chain transfer to transfer 3 data blocks from RAM to 3 different addresses. Where, first block: 10 data items of 32-bit width, from 0x20000000 to 0x20001000. Second block: 20 data items of 16-bit width, from 0x2000100 to 0x20001100. Third block: 40 data items of 8-bit width, from 0x20000200 to 0x20001200.

#### Register Setting

1. Enable the DMA controller by writing 1 to DMA\_EN.EN.
2. Select a channel, e.g., channel 0, and configure the first transfer descriptor (descriptor0) as follows:
  - a. Write DMA\_SAR0 to configure the source address as SRAM region 0x20000000.
  - b. Write DMA\_DAR0 to configure the destination address as SRAM region 0x20001000.
  - c. Write DMA\_DTCTL0 to configure the block size as 10.
  - d. Write the address of the second descriptor (descriptor1) 0x20002000 to the linked list pointer register DMA\_LLPO.

- e. Configure the channel control register DMA\_CHCTL0 to configure the transmission parameters for the first data block as follows:
  - 1) Chain transfer enabled.
  - 2) Chain transfer mode is to directly start the next transfer.
  - 3) Source address and destination address increment.
  - 4) The data width is 32-bit.
  - 5) Interrupt enable inactive.
3. Configure the second transfer descriptor (descriptor1) at address 0x20002000 in RAM as follows:
  - a. Write 32-bit data 0x20000100 to 0x20002000, which is the source address for the second transfer.
  - b. Write 32-bit data 0x20001100 to 0x20002004, which is the destination address for the second transfer.
  - c. Write to 0x20002008 to configure the block size as 20.
  - d. Write 32-bit data 0x20002020 to 0x20002018, which is the address of the third transfer descriptor (descriptor2).
  - e. Write to 0x2000201C the control data for the second transfer, which configures:
    - 1) Chain transfer enabled.
    - 2) Chain transfer mode is to directly start the next transfer.
    - 3) Source address and destination address increment.
    - 4) The data width is 16-bit.
    - 5) Interrupt enable inactive.
4. The descriptor for the third transfer (descriptor2) is configured in RAM at address 0x20002020, including:
  - a. Write the 32-bit data 0x20000200 to 0x20002020; this sets the source address for the third transfer.
  - b. Write the 32-bit data 0x20001200 to 0x20002024; this sets the destination address for the third transfer.
  - c. Block size as 40 at 0x20002028.
  - d. Last transfer in chain at 0x20002038 by writing 32-bit data 0x0.
  - e. Write to 0x2000203C the control data for the third transfer, which configures:
    - 1) Chain transfer disabled.
    - 2) Source address and destination address increment.
    - 3) The data width is 8-bit.
    - 4) Interrupt enable active.
5. Configure the trigger source selection register DMA\_TRGSEL0 to select software trigger as the start request for DMA channel 0.
6. Write 1 to the channel enable bit DMA\_CHEN.CHEN0 to enable channel 0.
7. Write to the software trigger register INTSFTTRG.STRG to 1 to send a start request, and DMA starts transferring data.

### Transfer Process

The software starts DMA and starts transmission. After the first transfer is completed, due to the chain transfer mode being set to directly start the next transfer and interrupts being disabled, DMA reads the second transfer descriptor (descriptor1) into the channel register. The second transfer starts directly based on the parameters configured in the descriptor. After the second transfer is completed, the third transfer descriptor (descriptor2) is read into the channel register. The third transfer starts based on the



parameters configured in the descriptor. After the third transmission is complete, the last transmission is linked according to the configuration information, and because the interrupt enable is enabled, DMA will generate a transmission finish interrupt and clear the channel enable bit DMA\_CHEN.CHEN0.

## 16.5 Register Description

### 16.5.1 Register List

- DMA1 base address: 0x40053000
- DMA2 base address: 0x40053400

Table 16-2 DMA Controller (DMA) Register Offset

| Offset      | Register                       | Description                                        | Width |
|-------------|--------------------------------|----------------------------------------------------|-------|
| 0x00        | <a href="#">DMA_EN</a>         | DMA Enable Register                                | 32    |
| 0x04        | <a href="#">DMA_INTSTAT0</a>   | Interrupt Status Register 0                        | 32    |
| 0x08        | <a href="#">DMA_INTSTAT1</a>   | Interrupt Status Register 1                        | 32    |
| 0x0C        | <a href="#">DMA_INTMASK0</a>   | Interrupt Mask Register 0                          | 32    |
| 0x10        | <a href="#">DMA_INTMASK1</a>   | Interrupt Mask Register 1                          | 32    |
| 0x14        | <a href="#">DMA_INTCLR0</a>    | Interrupt Status Clear Register 0                  | 32    |
| 0x18        | <a href="#">DMA_INTCLR1</a>    | Interrupt Status Clear Register 1                  | 32    |
| 0x1C        | <a href="#">DMA_CHEN</a>       | Channel Enable Register                            | 32    |
| 0x20        | <a href="#">DMA_REQSTAT</a>    | Transfer Request Status Register                   | 32    |
| 0x24        | <a href="#">DMA_CHSTAT</a>     | Channel Status Register                            | 32    |
| 0x2C        | <a href="#">DMA_RCFGCTL</a>    | Channel Reset Control Register                     | 32    |
| 0x30        | <a href="#">DMA_SWREQ</a>      | Software Trigger Register                          | 32    |
| 0x34        | <a href="#">DMA_CHENCLR</a>    | Channel Enable Reset Register                      | 32    |
| 0x40+0x40*x | <a href="#">DMA_SARx</a>       | Transmission Source Address Register               | 32    |
| 0x44+0x40*x | <a href="#">DMA_DARx</a>       | Transmission Destination Address Register          | 32    |
| 0x48+0x40*x | <a href="#">DMA_DTCTLx</a>     | Transfer Data Control Register                     | 32    |
| 0x4C+0x40*x | <a href="#">DMA_RPTx</a>       | Address Repeat Control Register                    | 32    |
| 0x4C+0x40*x | <a href="#">DMA_RPTBx</a>      | Address Repeat Control Register B                  | 32    |
| 0x50+0x40*x | <a href="#">DMA_SNSEQCTLx</a>  | Source Address NonSequence Control Register        | 32    |
| 0x50+0x40*x | <a href="#">DMA_SNSEQCTLBx</a> | Source Address NonSequence Control Register B      | 32    |
| 0x54+0x40*x | <a href="#">DMA_DNSEQCTLx</a>  | Destination Address NonSequence Control Register   | 32    |
| 0x54+0x40*x | <a href="#">DMA_DNSEQCTBx</a>  | Destination Address NonSequence Control Register B | 32    |
| 0x58+0x40*x | <a href="#">DMA_LL Px</a>      | Linked List Pointer Register                       | 32    |

| Offset      | Register         | Description                                              | Width |
|-------------|------------------|----------------------------------------------------------|-------|
| 0x5C+0x40*x | DMA_CHCTLx       | Channel Control Register                                 | 32    |
| 0x60+0x40*x | DMA_MONSARx      | Source Address Monitor Register                          | 32    |
| 0x64+0x40*x | DMA_MONDARx      | Destination Address Monitor Register                     | 32    |
| 0x68+0x40*x | DMA_MONDTCTLx    | Transfer Data Control Monitor Register                   | 32    |
| 0x6C+0x40*x | DMA_MONRPTx      | Address Repeat Control Monitor Register                  | 32    |
| 0x70+0x40*x | DMA_MONSNSEQCTLx | Source Address NonSequence Control Monitor Register      | 32    |
| 0x74+0x40*x | DMA_MONDNSEQCTLx | Destination Address NonSequence Control Monitor Register | 32    |

**Note**

The x in the table represents the existence of multiple registers, with values 0-7.

16.5.2 DMA Enable Register (DMA\_EN)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |    |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|----|
| 0x00   | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0  |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |    |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   | EN |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   | RW |

| Bit  | Name     | Function       | Description                                                                      | Access |
|------|----------|----------------|----------------------------------------------------------------------------------|--------|
| 31:1 | Reserved | Reserved       | -                                                                                | -      |
| 0    | EN       | DMA Enable Bit | <ul style="list-style-type: none"><li>0b0: Disable</li><li>0b1: Enable</li></ul> | RW     |

## 16.5.3 Interrupt Status Register 0 (DMA\_INTSTAT0)

| Offset | Bit Position |    |    |    |    |    |    |    |        |    |    |    |    |    |    |    |          |    |    |    |    |    |   |   |        |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|--------|----|----|----|----|----|----|----|----------|----|----|----|----|----|---|---|--------|---|---|---|---|---|---|---|
| 0x04   | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23     | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15       | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7      | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |        |    |    |    |    |    |    |    |          |    |    |    |    |    |   |   |        |   |   |   |   |   |   |   |
| Name   | Reserved     |    |    |    |    |    |    |    | REQERR |    |    |    |    |    |    |    | Reserved |    |    |    |    |    |   |   | TRNERR |   |   |   |   |   |   |   |
| Access |              |    |    |    |    |    |    |    | RO     |    |    |    |    |    |    |    |          |    |    |    |    |    |   |   | RO     |   |   |   |   |   |   |   |

| Bit   | Name     | Function                                      | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | Access |
|-------|----------|-----------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:24 | Reserved | Reserved                                      | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | -      |
| 23:16 | REQERR   | Transfer request overflow error interrupt bit | Corresponding to REQERR7-REQERR0: <ul style="list-style-type: none"> <li>0b0: No transfer request overflow error for this channel.</li> <li>0b1: Transfer request overflow error for this channel, meaning a new transfer request arrived while the previous request is still pending.</li> </ul> <div>  <b>Note</b><br/>           Each bit corresponds to a transmission channel.         </div> | RO     |
| 15:8  | Reserved | Reserved                                      | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | -      |
| 7:0   | TRNERR   | Transfer error interrupt bit                  | Corresponding to TRNERR7-TRNERR0: <ul style="list-style-type: none"> <li>0b0: No transfer error for this channel.</li> <li>0b1: Transfer error for this channel.</li> </ul> <div>  <b>Note</b><br/>           Each bit corresponds to a transmission channel.         </div>                                                                                                                      | RO     |

## 16.5.4 Interrupt Status Register 1 (DMA\_INTSTAT1)

| Offset | Bit Position |    |    |    |    |    |    |    |     |    |    |    |    |    |    |    |          |    |    |    |    |    |   |   |    |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|-----|----|----|----|----|----|----|----|----------|----|----|----|----|----|---|---|----|---|---|---|---|---|---|---|
| 0x08   | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23  | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15       | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7  | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |     |    |    |    |    |    |    |    |          |    |    |    |    |    |   |   |    |   |   |   |   |   |   |   |
| Name   | Reserved     |    |    |    |    |    |    |    | BTC |    |    |    |    |    |    |    | Reserved |    |    |    |    |    |   |   | TC |   |   |   |   |   |   |   |
| Access |              |    |    |    |    |    |    |    | RO  |    |    |    |    |    |    |    |          |    |    |    |    |    |   |   | RO |   |   |   |   |   |   |   |

| Bit   | Name     | Function                              | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                        | Access |
|-------|----------|---------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:24 | Reserved | Reserved                              | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | -      |
| 23:16 | BTC      | Block transfer complete interrupt bit | <p>For BTC7-BTC0, the interrupt occurs after the completion of one block transfer:</p> <ul style="list-style-type: none"> <li>0b0: No block transfer interrupt for this channel.</li> <li>0b1: Block transfer interrupt for this channel.</li> </ul> <div>  <b>Note</b><br/> Each bit corresponds to a transmission channel. </div>                                               | RO     |
| 15:8  | Reserved | Reserved                              | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | -      |
| 7:0   | TC       | Transfer complete interrupt bit       | <p>Corresponding to TC7-TC0, this interrupt occurs after completing the number of transmissions set by DMA_DTCTLx.CNT:</p> <ul style="list-style-type: none"> <li>0b0: No transfer completed interrupt for this channel.</li> <li>0b1: Transfer completed interrupt for this channel.</li> </ul> <div>  <b>Note</b><br/> Each bit corresponds to a transmission channel. </div> | RO     |

## 16.5.5 Interrupt Mask Register 0 (DMA\_INTMASK0)

| Offset | Bit Position |    |    |    |    |    |    |    |           |    |    |    |    |    |    |    |          |    |    |    |    |    |   |   |           |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|-----------|----|----|----|----|----|----|----|----------|----|----|----|----|----|---|---|-----------|---|---|---|---|---|---|---|
| 0x0C   | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23        | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15       | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7         | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |           |    |    |    |    |    |    |    |          |    |    |    |    |    |   |   |           |   |   |   |   |   |   |   |
| Name   | Reserved     |    |    |    |    |    |    |    | MSKREQERR |    |    |    |    |    |    |    | Reserved |    |    |    |    |    |   |   | MSKTRNERR |   |   |   |   |   |   |   |
| Access |              |    |    |    |    |    |    |    | RW        |    |    |    |    |    |    |    |          |    |    |    |    |    |   |   | RW        |   |   |   |   |   |   |   |

| Bit   | Name      | Function                                     | Description                                                                                                                                                                                                                                                                                                                                                                                        | Access |
|-------|-----------|----------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:24 | Reserved  | Reserved                                     | -                                                                                                                                                                                                                                                                                                                                                                                                  | -      |
| 23:16 | MSKREQERR | Transmission request overflow interrupt mask | Corresponding to MSKREQERR7-MSKREQERR0: <ul style="list-style-type: none"> <li>0b0: Do not mask the transfer request overflow interrupt.</li> <li>0b1: Mask the transfer request overflow interrupt.</li> </ul> <div>  <b>Note</b><br/>           Each bit corresponds to a transmission channel.         </div> | RW     |
| 15:8  | Reserved  | Reserved                                     | -                                                                                                                                                                                                                                                                                                                                                                                                  | -      |
| 7:0   | MSKTRNERR | Transfer error interrupt mask                | Corresponding to MSKTRNERR7-MSKTRNERR0: <ul style="list-style-type: none"> <li>0b0: Do not mask the transfer error interrupt.</li> <li>0b1: Mask the transfer error interrupt.</li> </ul> <div>  <b>Note</b><br/>           Each bit corresponds to a transmission channel.         </div>                      | RW     |

## 16.5.6 Interrupt Mask Register 1(DMA\_INTMASK1)

| Offset | Bit Position |    |    |    |    |    |    |    |        |    |    |    |    |    |    |    |          |    |    |    |    |    |   |   |       |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|--------|----|----|----|----|----|----|----|----------|----|----|----|----|----|---|---|-------|---|---|---|---|---|---|---|
| 0x10   | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23     | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15       | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7     | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |        |    |    |    |    |    |    |    |          |    |    |    |    |    |   |   |       |   |   |   |   |   |   |   |
| Name   | Reserved     |    |    |    |    |    |    |    | MSKBTC |    |    |    |    |    |    |    | Reserved |    |    |    |    |    |   |   | MSKTC |   |   |   |   |   |   |   |
| Access |              |    |    |    |    |    |    |    | RW     |    |    |    |    |    |    |    |          |    |    |    |    |    |   |   | RW    |   |   |   |   |   |   |   |

| Bit   | Name     | Function                               | Description                                                                                                                                                                                                                                                                                                                                                                               | Access |
|-------|----------|----------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:24 | Reserved | Reserved                               | -                                                                                                                                                                                                                                                                                                                                                                                         | -      |
| 23:16 | MSKBTC   | Block transfer complete interrupt mask | Corresponding to MSKBTC7-MSKBTC0: <ul style="list-style-type: none"> <li>0b0: Do not mask the block transfer completed interrupt.</li> <li>0b1: Mask the block transfer completed interrupt.</li> </ul> <div>  <b>Note</b><br/>           Each bit corresponds to a transmission channel.         </div> | RW     |
| 15:8  | Reserved | Reserved                               | -                                                                                                                                                                                                                                                                                                                                                                                         | -      |
| 7:0   | MSKTC    | Transfer completed interrupt mask      | Corresponding to MSKTC7-MSKTC0: <ul style="list-style-type: none"> <li>0b0: Do not mask the transfer completed interrupt.</li> <li>0b1: Mask the transfer completed interrupt.</li> </ul> <div>  <b>Note</b><br/>           Each bit corresponds to a transmission channel.         </div>             | RW     |



## 16.5.7 Interrupt Status Clear Register 0 (DMA\_INTCLR0)

| Offset | Bit Position |    |    |    |    |    |    |    |           |    |    |    |    |    |    |    |          |    |    |    |    |    |   |   |           |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|-----------|----|----|----|----|----|----|----|----------|----|----|----|----|----|---|---|-----------|---|---|---|---|---|---|---|
| 0x14   | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23        | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15       | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7         | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |           |    |    |    |    |    |    |    |          |    |    |    |    |    |   |   |           |   |   |   |   |   |   |   |
| Name   | Reserved     |    |    |    |    |    |    |    | CLRREQERR |    |    |    |    |    |    |    | Reserved |    |    |    |    |    |   |   | CLRTRNERR |   |   |   |   |   |   |   |
| Access |              |    |    |    |    |    |    |    | WO        |    |    |    |    |    |    |    |          |    |    |    |    |    |   |   | WO        |   |   |   |   |   |   |   |

| Bit   | Name      | Function                                            | Description                                                                                                                                                                                                                                                                                                                                               | Access |
|-------|-----------|-----------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:24 | Reserved  | Reserved                                            | -                                                                                                                                                                                                                                                                                                                                                         | -      |
| 23:16 | CLRREQERR | Transmission request overflow error interrupt reset | <p>Corresponding to CLRREQERR7-CLRREQERR0: Write 1 to reset the transfer request overflow error interrupt status bit, write 0 has no effect; read always returns 0.</p> <div>  <b>Note</b><br/>           Each bit corresponds to a transmission channel.         </div> | WO     |
| 15:8  | Reserved  | Reserved                                            | -                                                                                                                                                                                                                                                                                                                                                         | -      |
| 7:0   | CLRTRNERR | Transfer error interrupt reset bit:                 | <p>Corresponding to CLRTRNERR7-CLRTRNERR0: Write 1 to reset the transfer error interrupt status bit, write 0 has no effect; read always returns 0.</p> <div>  <b>Note</b><br/>           Each bit corresponds to a transmission channel.         </div>                | WO     |

## 16.5.8 Interrupt Status Clear Register 1 (DMA\_INTCLR1)

| Offset | Bit Position |    |    |    |    |    |    |    |         |    |    |    |    |    |    |    |          |    |    |    |    |    |   |   |        |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|---------|----|----|----|----|----|----|----|----------|----|----|----|----|----|---|---|--------|---|---|---|---|---|---|---|
| 0x18   | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23      | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15       | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7      | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |         |    |    |    |    |    |    |    |          |    |    |    |    |    |   |   |        |   |   |   |   |   |   |   |
| Name   | Reserved     |    |    |    |    |    |    |    | CLR BTC |    |    |    |    |    |    |    | Reserved |    |    |    |    |    |   |   | CLR TC |   |   |   |   |   |   |   |
| Access |              |    |    |    |    |    |    |    | WO      |    |    |    |    |    |    |    |          |    |    |    |    |    |   |   | WO     |   |   |   |   |   |   |   |

| Bit   | Name     | Function                                 | Description                                                                                                                                                                                                                                                                                                                                    | Access |
|-------|----------|------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:24 | Reserved | Reserved                                 | -                                                                                                                                                                                                                                                                                                                                              | -      |
| 23:16 | CLR BTC  | Block transport complete interrupt reset | <p>Corresponding to CLR BTC7-CLR BTC0: Write 1 to reset the block transfer completed interrupt status bit, write 0 has no effect; read always returns 0.</p> <div>  <b>Note</b><br/>           Each bit corresponds to a transmission channel.         </div> | WO     |
| 15:8  | Reserved | Reserved                                 | -                                                                                                                                                                                                                                                                                                                                              | -      |
| 7:0   | CLR TC   | Transfer completed interrupt reset       | <p>Corresponding to CLR TC7-CLR TC0: Write 1 to reset the transfer completed interrupt status bit, write 0 has no effect; read always returns 0.</p> <div>  <b>Note</b><br/>           Each bit corresponds to a transmission channel.         </div>       | WO     |

## 16.5.9 Channel Enable Register (DMA\_CHEN)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |      |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|------|---|---|---|---|---|---|---|
| 0x1C   | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7    | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |      |   |   |   |   |   |   |   |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   | CHEN |   |   |   |   |   |   |   |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   | RW   |   |   |   |   |   |   |   |

| Bit  | Name     | Function           | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | Access |
|------|----------|--------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:8 | Reserved | Reserved           | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | -      |
| 7:0  | CHEN     | Channel enable bit | <p>Corresponding to CHEN7-CHEN0:</p> <ul style="list-style-type: none"> <li>0b0: The channel is disabled</li> <li>0b1: The channel is enabled</li> </ul> <p>Write 1 to CHEN[x] sets CHEN[x] to 1, write 0 has no effect.</p> <p>When writing 1 to the DMA_CHENCLR.CHENCLR[x] bit, CHEN[x] clears 0 and forces to interrupt the ongoing transport of channel x. The enable bit remains at 1 during the transmission process, and will be automatically cleared to 0 after the number of transmissions set by register DMA_DTCTLx.CNT is completed. If DMA_DTCTLx.CNT is set to 0, it will not be automatically cleared after the transfer is completed, which means unlimited transfers.</p> <div>  <b>Note</b> <ul style="list-style-type: none"> <li>Each bit corresponds to a transmission channel.</li> <li>Set CHEN[x] to 1 after configuring all 8 descriptor registers for channel x to enable the channel. When CHEN[x] is 1, write operations to the descriptor registers for channel x has no effect.</li> </ul> </div> | RW     |

## 16.5.10 Transfer Request Status Register (DMA\_REQSTAT)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |         |          |    |    |    |    |   |   |   |       |   |   |   |   |   |   |  |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---------|----------|----|----|----|----|---|---|---|-------|---|---|---|---|---|---|--|
| 0x20   | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15      | 14       | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6     | 5 | 4 | 3 | 2 | 1 | 0 |  |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |         |          |    |    |    |    |   |   |   |       |   |   |   |   |   |   |  |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RCFGREQ | Reserved |    |    |    |    |   |   |   | CHREQ |   |   |   |   |   |   |  |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RO      |          |    |    |    |    |   |   |   | RO    |   |   |   |   |   |   |  |

| Bit   | Name     | Function                          | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | Access |
|-------|----------|-----------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:17 | Reserved | Reserved                          | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | -      |
| 16    | RCFGREQ  | Channel reset request flag        | <ul style="list-style-type: none"> <li>0b0: No channel reset request.</li> <li>0b1: Channel reset request present.</li> </ul> <div>  <b>Note</b><br/> Set to 1 after external reset request input; cleared to 0 after channel reset initiation or when channel reset is disabled. </div>                                                                                                                                                                                                                                                                                                                                                                                                           | RO     |
| 15:8  | Reserved | Reserved                          | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | -      |
| 7:0   | CHREQ    | Channel transfer request flag bit | Corresponding to CHREQ7-CHREQ0: <ul style="list-style-type: none"> <li>0b0: No transfer request for this channel</li> <li>0b1: Transfer request for this channel present</li> </ul> <div>  <b>Note</b><br/> Each bit corresponds to a transmission channel.<br/> Set to 1 after external transfer request input; cleared to 0 after transfer initiation for the channel, transmission error, or when the transfer permission bit (DMAEN or CHEN[x]) is written to 0.<br/> If this bit is 1, and another transfer request is input for the channel, a transfer request overflow error occurs, the second request is ignored, but this flag bit remains 1, continuing to wait for transfer. </div> | RO     |

## 16.5.11 Channel Status Register (DMA\_CHSTAT)

| Offset | Bit Position |    |    |    |    |    |    |    |       |    |    |    |    |    |    |    |          |    |    |    |    |    |   |   |         |   |        |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|-------|----|----|----|----|----|----|----|----------|----|----|----|----|----|---|---|---------|---|--------|---|---|---|---|---|
| 0x24   | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23    | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15       | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7       | 6 | 5      | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |       |    |    |    |    |    |    |    |          |    |    |    |    |    |   |   |         |   |        |   |   |   |   |   |
| Name   | Reserved     |    |    |    |    |    |    |    | CHACT |    |    |    |    |    |    |    | Reserved |    |    |    |    |    |   |   | RCFGACT |   | DMAACT |   |   |   |   |   |
| Access |              |    |    |    |    |    |    |    | RO    |    |    |    |    |    |    |    |          |    |    |    |    |    |   |   | RO      |   | RO     |   |   |   |   |   |

| Bit   | Name     | Function                             | Description                                                                                                                                                                                                                                                                                                                         | Access |
|-------|----------|--------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:24 | Reserved | Reserved                             | -                                                                                                                                                                                                                                                                                                                                   | -      |
| 23:16 | CHACT    | Transfer action channel monitor bit  | Corresponding to CHACT7-CHACT0: <ul style="list-style-type: none"> <li>0b0: The channel is idle</li> <li>0b1: The channel is in action</li> </ul> <div>  <b>Note</b><br/>           Each bit corresponds to a transmission channel.         </div> | RO     |
| 15:2  | Reserved | Reserved                             | -                                                                                                                                                                                                                                                                                                                                   | -      |
| 1     | RCFGACT  | DMA channel reset action monitor bit | <ul style="list-style-type: none"> <li>0b0: DMA is not in channel reset action</li> <li>0b1: DMA is in channel reset action</li> </ul>                                                                                                                                                                                              | RO     |
| 0     | DMAACT   | DMA action monitor bit               | <ul style="list-style-type: none"> <li>0b0: DMA is not in transfer action</li> <li>0b1: DMA is in transfer action</li> </ul>                                                                                                                                                                                                        | RO     |

## 16.5.12 Channel Reset Control Register (DMA\_RCFGCTL)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |       |       |       |          |    |    |    |    |         |    |    |    |   |          |   |   |   |   |         |        |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|-------|-------|-------|----------|----|----|----|----|---------|----|----|----|---|----------|---|---|---|---|---------|--------|---|---|
| 0x2C   | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21    | 20    | 19    | 18       | 17 | 16 | 15 | 14 | 13      | 12 | 11 | 10 | 9 | 8        | 7 | 6 | 5 | 4 | 3       | 2      | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |       |       |       |          |    |    |    |    |         |    |    |    |   |          |   |   |   |   |         |        |   |   |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    | CNTMD | DARMD | SARMD | Reserved |    |    |    |    | RCFGCHS |    |    |    |   | Reserved |   |   |   |   | RCFGLLP | RCFGEN |   |   |
| Access |              |    |    |    |    |    |    |    |    |    | RW    | RW    | RW    |          |    |    |    |    | RW      |    |    |    |   |          |   |   |   |   | RW      | RW     |   |   |

| Bit   | Name     | Function                                      | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | Access |
|-------|----------|-----------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:22 | Reserved | Reserved                                      | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | -      |
| 21:20 | CNTMD    | Remaining transfer count counter reset method | <ul style="list-style-type: none"> <li>0b00: Remain unchanged</li> <li>0b01: Source address mode</li> <li>0b10/0b11: Destination address mode</li> </ul> <p>When source address mode is selected, and source address nonSequence reset is selected, the remaining transfer count counter updates to the state after the transfer count specified by DMA_SNSEQCTLBx.SNSCNTB; when source address repeat is selected, the counter updates to the state after the transfer count specified by DMA_RPTBx.SRPTB. When the source address selection is held, the remaining transfer times counter also remains unchanged.</p> <p>When the destination address mode is selected, it is similar to the source address mode.</p>                                                                                                                                                                                                                                                                 | RW     |
| 19:18 | DARMD    | Destination address reset mode                | <ul style="list-style-type: none"> <li>0b00: Remain unchanged</li> <li>0b01: NonSequence reset, the next transfer's destination address updates to <math>\text{addr\_base} + (\text{DMA\_DNSEQCTLBx.DNSDIST} \times \text{DMA\_CHCTLx.HSIZE}(\text{bit}) / 8)</math>, where <math>\text{addr\_base}</math> is the starting address of the current nonSequence transmission region.</li> <li>0b10/0b11: Repeated reset, the next transfer's destination address updates to the initial setting value of the DMA_DARx register.</li> </ul> <div style="border: 1px solid black; padding: 5px; margin-top: 10px;"> <p> <b>Notice</b></p> <p>Only set DARMD[1:0] to 0b01 when destination address nonSequence is enabled (DMA_CHCTLx.DNSEQEN=1) for this channel.</p> <p>Only set DARMD[1] to 1 when destination address repeated transfer is enabled (DMA_CHCTLx.DRPTEN=1) for this channel.</p> </div> | RW     |

| Bit   | Name     | Function                          | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | Access |
|-------|----------|-----------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 17:16 | SARMD    | Source address reset method       | <ul style="list-style-type: none"> <li>0b00: Remain unchanged</li> <li>0b01: NonSequence reset, the next transfer's source address updates to <math>\text{addr\_base} + (\text{DMA\_SNSEQCTLBx.SNSDIST} \times \text{DMA\_CHCTLx.HSIZE}(\text{bit}) / 8)</math>, where <math>\text{addr\_base}</math> is the starting address of the current nonSequence transmission region.</li> <li>0b10/0b11: Repeated reset, the next transfer's source address updates to the initial setting value of the DMA_SARx register.</li> </ul> <div>  <b>Notice</b><br/> Only set SARMD[1:0] to 0b01 when source address NonSequence is enabled (DMA_CHCTLx.SNSEQEN=1) for this channel.<br/><br/> Only set SARMD[1] to 1 when source address repeated transfer is enabled (DMA_CHCTLx.SRPTEN=1) for this channel. </div> | RW     |
| 15:12 | Reserved | Reserved                          | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | -      |
| 11:8  | RCFGCHS  | Reset Channel Selection           | <ul style="list-style-type: none"> <li>0x0: Channel 0</li> <li>0x1: Channel 1</li> <li>...</li> <li>0x7: Channel 7</li> <li>Others: Setting prohibited</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | RW     |
| 7:2   | Reserved | Reserved                          | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | -      |
| 1     | RCFGLLP  | Linked list pointer channel reset | <ul style="list-style-type: none"> <li>0b0: Linked list pointer reset inactive</li> <li>0b1: Linked list pointer reset active</li> </ul> <div>  <b>Notice</b><br/> When RCFGPLL is set to 1, the channel will reload a new descriptor from memory, so Bits 21-16 of this register are inactive. </div>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | RW     |
| 0     | RCFGEN   | Channel Reset Permission          | <ul style="list-style-type: none"> <li>0b0: Disable event-triggered channel configuration register forced update</li> <li>0b1: Enable event-triggered channel configuration register forced update</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | RW     |

 **Notice**

Set this register when DMA\_EN.EN is 0, and it must be set before the first transfer of the reset channel.

## 16.5.13 Software Trigger Register (DMA\_SWREQ)

| Offset | Bit Position |    |    |    |    |    |    |    |         |    |    |    |    |    |    |    |           |          |    |    |    |    |   |   |   |       |   |   |   |   |   |   |  |
|--------|--------------|----|----|----|----|----|----|----|---------|----|----|----|----|----|----|----|-----------|----------|----|----|----|----|---|---|---|-------|---|---|---|---|---|---|--|
| 0x30   | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23      | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15        | 14       | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6     | 5 | 4 | 3 | 2 | 1 | 0 |  |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |         |    |    |    |    |    |    |    |           |          |    |    |    |    |   |   |   |       |   |   |   |   |   |   |  |
| Name   | SWRCFGWP     |    |    |    |    |    |    |    | SWREQWP |    |    |    |    |    |    |    | SWRCFGREQ | Reserved |    |    |    |    |   |   |   | SWREQ |   |   |   |   |   |   |  |
| Access |              |    |    |    |    |    |    |    |         |    |    |    |    |    |    |    | WO        |          |    |    |    |    |   |   |   |       |   |   |   |   |   |   |  |

| Bit   | Name      | Function                                          | Description                                                                                                                                                                                                                                                                                                                                                                                                                       | Access |
|-------|-----------|---------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:24 | SWRCFGWP  | Software Channel Reset Request Write Protection   | Software reset request write protection, write 0xA2 to SWRCFGWP to release write protection.                                                                                                                                                                                                                                                                                                                                      | WO     |
| 23:16 | SWREQWP   | Software Channel Trigger Request Write Protection | Software trigger request write protection, write 0xA1 to SWREQWP to release write protection.                                                                                                                                                                                                                                                                                                                                     | WO     |
| 15    | SWRCFGREQ | Software Channel Reset Request                    | Writing 1 (with SWRCFGWP written to 0xA2) generates a software channel reset request, write 0 has no effect.<br><br><div>  <b>Note</b><br/>           Software-triggered channel reset function, equivalent to AOS trigger effect.         </div>                                                                                              | WO     |
| 14:8  | Reserved  | Reserved                                          | -                                                                                                                                                                                                                                                                                                                                                                                                                                 | -      |
| 7:0   | SWREQ     | Software Channel Trigger Request                  | Corresponding to SWREQ7-SWREQ0:<br>Writing 1 (with SWREQWP written to 0xA1) generates a software channel reset request, write 0 has no effect.<br><br><div>  <b>Note</b><br/>           Each bit corresponds to a transmission channel.<br/>           Software-triggered start and transfer, equivalent to AOS trigger effect.         </div> | WO     |



### 16.5.14 Channel Enable Reset Register (DMA\_CHENCLR)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |         |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---------|---|---|---|---|---|---|---|
| 0x34   | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7       | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |         |   |   |   |   |   |   |   |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   | CHENCLR |   |   |   |   |   |   |   |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   | WO      |   |   |   |   |   |   |   |

| Bit  | Name     | Function                 | Description                                                                                                                                                                                                                                                                                                                                                                                                                           | Access |
|------|----------|--------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:8 | Reserved | Reserved                 | -                                                                                                                                                                                                                                                                                                                                                                                                                                     | -      |
| 7:0  | CHENCLR  | Channel enable bit reset | <p>Write 1 to reset the channel enable bit CHEN, write 0 has no effect; read always returns 0. When writing 1, if the channel is currently transferring, DMA will forcibly terminate the transfer after completing the current data read/write.</p> <div>  <b>Note</b><br/>           Each bit corresponds to a transmission channel.         </div> | WO     |

### 16.5.15 Transmission Source Address Register (DMA\_SARx) (x=0-7)

Offset: 0x40+0x40\*x

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| -      | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | SAR          |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access | RW           |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit  | Name | Function                | Description                                                                                                                                                                                                                                                                                                                                                                              | Access |
|------|------|-------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:0 | SAR  | Transfer source address | <p>Set source address</p> <div>  <b>Note</b> <ul style="list-style-type: none"> <li>When the transfer data width is 16-bit, i.e. DMA_CHCTLx.HSIZE=0b01, SAR[0] is disabled.</li> <li>When the transfer data width is 32-bit, i.e. DMA_CHCTLx.HSIZE=0b10/0b11, SAR[1:0] is disabled.</li> </ul> </div> | RW     |

16.5.16 Transmission Destination Address Register (DMA\_DARx) (x=0-7)

Offset: 0x44+0x40\*x

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| '      | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | DAR          |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access | RW           |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit  | Name | Function                         | Description                                                                                                                                                                                                                                                                                                                                                                                                                             | Access |
|------|------|----------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:0 | DAR  | Transmission destination address | <div>Set transmission destination address</div> <div><div><div> <b>Note</b></div><div><ul style="list-style-type: none"><li>When the transfer data width is 16-bit, i.e. DMA_CHCTLx.HSIZE=0b01, DAR[0] is disabled.</li><li>When the transfer data width is 32-bit, i.e. DMA_CHCTLx.HSIZE=0b10/0b11, DAR[1:0] is disabled.</li></ul></div></div></div> | RW     |

## 16.5.17 Transfer Data Control Register (DMA\_DTCTLx) (x=0-7)

Offset: 0x48+0x40\*x

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |          |    |    |    |         |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----------|----|----|----|---------|----|---|---|---|---|---|---|---|---|---|---|
| -      | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15       | 14 | 13 | 12 | 11      | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000001   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |          |    |    |    |         |    |   |   |   |   |   |   |   |   |   |   |
| Name   | CNT          |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | Reserved |    |    |    | BLKSIZE |    |   |   |   |   |   |   |   |   |   |   |
|        |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |          |    |    |    |         |    |   |   |   |   |   |   |   |   |   |   |
| Access | RW           |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |          |    |    |    | RW      |    |   |   |   |   |   |   |   |   |   |   |

| Bit   | Name     | Function        | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                         | Access |
|-------|----------|-----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:16 | CNT      | Transfer count  | The total number of transfers, each request starts the transfer of a data block, the number of transfers is decremented by 1 when it is completed, and a transfer completion interrupt occurs when it is reduced to 0. If it is set to 0, it means unlimited transmission. Each time a data block is requested to be transmitted, the transmission times counter will remain at 0 when it is completed, and no transmission completion interrupt will be generated. | RW     |
| 15:10 | Reserved | Reserved        | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | -      |
| 9:0   | BLKSIZE  | Data block size | Set the size of the data block, up to 1024 pieces of data can be configured. The width of each data item is determined by the DMA_CHCTLx.HSIZE bit. Register value set to 1 transfers 1 data item, set to 0 transfers 1024 data items.                                                                                                                                                                                                                              | RW     |

### 16.5.18 Address Repeat Control Register (DMA\_RPTx) (x=0-7)

Offset: 0x4C+0x40\*x

| Offset | Bit Position |    |    |    |    |    |    |    |           |    |    |    |    |    |    |    |          |    |    |    |    |    |   |   |           |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|-----------|----|----|----|----|----|----|----|----------|----|----|----|----|----|---|---|-----------|---|---|---|---|---|---|---|
| -      | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23        | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15       | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7         | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |           |    |    |    |    |    |    |    |          |    |    |    |    |    |   |   |           |   |   |   |   |   |   |   |
| Name   | Reserved     |    |    |    |    |    |    |    | DRPT[9:0] |    |    |    |    |    |    |    | Reserved |    |    |    |    |    |   |   | SRPT[9:0] |   |   |   |   |   |   |   |
| Access |              |    |    |    |    |    |    |    | RW        |    |    |    |    |    |    |    |          |    |    |    |    |    |   |   | RW        |   |   |   |   |   |   |   |

| Bit   | Name      | Function                               | Description                                                                                                                                                                                                                                                                                                                                                                  | Access |
|-------|-----------|----------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:26 | Reserved  | Reserved                               | -                                                                                                                                                                                                                                                                                                                                                                            | -      |
| 25:16 | DRPT[9:0] | Destination address repeat region size | Set destination address repeat area size<br>The destination device reloads the destination address to the initial setting value of the DMA_DARx register after transferring DRPT data items. Register set to 10 reloads the address after transferring 10 data items, set to 0 reloads after 1024 data items. The width of each data item is determined by DMA_CHCTLx.HSIZE. | RW     |
| 15:10 | Reserved  | Reserved                               | -                                                                                                                                                                                                                                                                                                                                                                            | -      |
| 9:0   | SRPT[9:0] | Source address repeat zone size        | Set the source address repeat area size<br>The source device reloads the source address to the initial setting value of the DMA_SARx register after transferring SRPT data items. Register set to 10 reloads the address after transferring 10 data items, set to 0 reloads after 1024 data items. The width of each data item is determined by DMA_CHCTLx.HSIZE.            | RW     |



#### Note

1. This register configures the size of the source and destination address repeat regions. Repeat transfer requires configuring the DMA\_CHCTLx.SRPTEN/DRPTEN bits to be active, and configuring the DMA\_CHCTLx.SINC/DINC bits to enable address update as increment or decrement. When configuring the DMA\_CHCTLx.SINC/DINC bits to enable address update as fixed, address repeated jump is invalid.
2. This register is disabled when the channel's reset function is enabled.

### 16.5.19 Address Repeat Control Register B (DMA\_RPTBx) (x=0-7)

Offset: 0x4C+0x40\*x

| Offset | Bit Position |    |    |    |    |    |            |    |    |    |    |    |    |    |          |    |    |    |    |    |            |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|------------|----|----|----|----|----|----|----|----------|----|----|----|----|----|------------|----|---|---|---|---|---|---|---|---|---|---|
| -      | 31           | 30 | 29 | 28 | 27 | 26 | 25         | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17       | 16 | 15 | 14 | 13 | 12 | 11         | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |            |    |    |    |    |    |    |    |          |    |    |    |    |    |            |    |   |   |   |   |   |   |   |   |   |   |
| Name   | Reserved     |    |    |    |    |    | DRPTB[9:0] |    |    |    |    |    |    |    | Reserved |    |    |    |    |    | SRPTB[9:0] |    |   |   |   |   |   |   |   |   |   |   |
| Access |              |    |    |    |    |    | RW         |    |    |    |    |    |    |    |          |    |    |    |    |    | RW         |    |   |   |   |   |   |   |   |   |   |   |

| Bit   | Name       | Function                               | Description                                                                                                                                                                                                                                                         | Access |
|-------|------------|----------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:26 | Reserved   | Reserved                               | -                                                                                                                                                                                                                                                                   | -      |
| 25:16 | DRPTB[9:0] | Destination address repeat region size | Set destination address repeat area size<br>The destination device reloads the destination address to the initial setting value of the DMA_DARx register after transferring DRPTB blocks. The block size is determined by DMA_DTCTLx.BLKSIZE and DMA_CHCTLx.HSIZE.  | RW     |
| 15:10 | Reserved   | Reserved                               | -                                                                                                                                                                                                                                                                   | -      |
| 9:0   | SRPTB[9:0] | Source address repeat zone size        | Set the source address repeat area size<br>The source device reloads the initial value set in the DMA_SARx register into the Source Address after transferring every SRPTB blocks of data. The block size is determined by DMA_DTCTLx.BLKSIZE and DMA_CHCTLx.HSIZE. | RW     |



#### Note

1. This register configures the size of the source and destination address repeat regions. Repeat transfer requires configuring the DMA\_CHCTLx.SRPTEN/DRPTEN bits to be active, and configuring the DMA\_CHCTLx.SINC/DINC bits to enable address update as increment or decrement. When configuring the DMA\_CHCTLx.SINC/DINC bits to enable address update as fixed, address repeated jump is invalid.
2. This register is only active when the channel's reset function is enabled, replacing the DMA\_RPTx register. It is disabled when the channel's reset function is disabled.

### 16.5.20 Source Address NonSequence Control Register (DMA\_SNSEQCTLx) (x=0-7)

Offset: 0x50+0x40\*x

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |               |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|---------------|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| -      | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19            | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |               |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | SNSCNT[11:0] |    |    |    |    |    |    |    |    |    |    |    | SOFFSET[19:0] |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access | RW           |    |    |    |    |    |    |    |    |    |    |    | RW            |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit   | Name          | Function                      | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | Access |
|-------|---------------|-------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:20 | SNSCNT[11:0]  | Source address jump data size | Set the amount of data transferred before the source address jumps.<br>After the source device transmits SNSCNT data, the source address jumps according to the offset specified by SOFFSET. If the register is set to 10, the address will jump after every 10 data transmissions, if it is set to 0, the address will jump after every 4096 data transmissions.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | RW     |
| 19:0  | SOFFSET[19:0] | Source address jump offset    | When nonSequence address transmission, set the offset of the source address jump. The offset is relative to the current transfer address, that is, the last transfer address before the jump. The jump direction is forward or backward according to the DMA_CHCTLx.SINC bit. For details, refer to <a href="#">Schematic Diagram of NonSequence Address Transfer</a> . NonSequence address transfer is disabled when DMA_CHCTLx.SINC is set to fixed address. The jump address will be calculated according to the number of bits set by the data width (DMA_CHCTLx.HSIZE) and the value of SOFFSET.<br>Address Offset=SOFFSET×(DMA_CHCTLx.HSIZE(bit)/8)<br>For example, when SOFFSET is set to 10, and the data width is 32-bit, the address offset is 10×4=40; if the data width is 16-bit, the offset is 10×2=20; if 8-bit, the offset is 10×1=10. The next transmission's source address=current source address ± address offset | RW     |



#### Note

1. NonSequence for the source device requires configuring the DMA\_CHCTLx.SNSEQEN bit to be active, and configuring the DMA\_CHCTLx.SINC bit to enable address update as increment or decrement.
2. This register is disabled when the channel's reset function is enabled.

### 16.5.21 Source Address NonSequence Control Register B (DMA\_SNSEQCTLBx) (x=0-7)

Offset: 0x50+0x40\*x

| Offset | Bit Position  |    |    |    |    |    |    |    |    |    |    |    |               |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|---------------|----|----|----|----|----|----|----|----|----|----|----|---------------|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| '      | 31            | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19            | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000    |    |    |    |    |    |    |    |    |    |    |    |               |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | SNSCNTB[11:0] |    |    |    |    |    |    |    |    |    |    |    | SNSDIST[19:0] |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access | RW            |    |    |    |    |    |    |    |    |    |    |    | RW            |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit   | Name          | Function                                 | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | Access |
|-------|---------------|------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:20 | SNSCNTB[11:0] | Source address jump data size            | Set the amount of data transferred before the source address jumps.<br>After the source device transmits SNSCNTB data blocks, the source address jumps according to the address interval specified by SNSDIST. The block size is determined by DMA_DTCTLx.BLKSIZE and DMA_CHCTLx.HSIZE.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | RW     |
| 19:0  | SNSDIST[19:0] | Source nonSequence area address distance | When nonSequence address transmission, set the distance between two nonSequence areas of the source device. The jump direction is forward or backward according to the DMA_CHCTLx.SINC bit. For details, refer to <a href="#">Figure 16-4 : Schematic Diagram of NonSequence Reset</a> . NonSequence address transfer is disabled when DMA_CHCTLx.SINC is set to fixed address.<br>The address spacing is calculated based on the data width defined by the DMA_CHCTLx.HSIZE bit field and the SNSDIST value as follows:<br>Address Distance=SNSDIST×(DMA_CHCTLx.HSIZE(bit)/8)<br>For example, when SNSDIST is set to 10, and the data width is 32-bit, the address distance is 10×4=40; if 16-bit, the distance is 10×2=20; if 8-bit, the distance is 10×1=10.<br>The next transmission's source address=starting address of the current source nonSequence region ± address distance. | RW     |



#### Note

1. NonSequence for the source device requires configuring the DMA\_CHCTLx.SNSEQEN bit to be active, and configuring the DMA\_CHCTLx.SINC bit to enable address update as increment or decrement.
2. This register is only active when the channel's reset function is enabled, replacing the DMA\_SNSEQCTLx register. It is disabled when the channel's reset function is disabled.

### 16.5.22 Destination Address NonSequence Control Register (DMA\_DNSEQCTLx) (x=0-7)

Offset: 0x54+0x40\*x

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |               |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|---------------|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| -      | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19            | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |               |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | DNSCNT[11:0] |    |    |    |    |    |    |    |    |    |    |    | DOFFSET[19:0] |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access | RW           |    |    |    |    |    |    |    |    |    |    |    | RW            |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit   | Name          | Function                                | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | Access |
|-------|---------------|-----------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:20 | DNSCNT[11:0]  | Destination address jump data size      | Set the amount of data transferred before the destination address jumps.<br>After the destination device transmits DNSCNT data, the destination address jumps according to the offset specified by DOFFSET. If the register is set to 10, the address will jump after every 10 data transmissions, if it is set to 0, the address will jump after every 4096 data transmissions.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | RW     |
| 19:0  | DOFFSET[19:0] | Destination address jump address offset | When nonSequence address transfer, set the offset of the destination address jump. The offset is relative to the current transfer address, that is, the last transfer address before the jump. The jump direction is forward or backward according to the DMA_CHCTLx.DINC bit. For details, refer to the <a href="#">Schematic Diagram of NonSequence Address Transfer</a> . Non-Sequence address transfer is disabled when DMA_CHCTLx.DINC is set to fixed address. The jump address is calculated based on the bit count set by the data width (DMA_CHCTLx.HSIZE) and the DOFFSET value.<br>Address Offset=DOFFSET×(DMA_CHCTLx.HSIZE(bit)/8)<br>For example, when DOFFSET is set to 10, and the data width is 32-bit, the address offset is 10×4=40; if 16-bit, the offset is 10×2=20; if 8-bit, the offset is 10×1=10.<br>The next transmission's destination address=current destination address ± address offset. | RW     |



**Note**

1. NonSequence for the destination device requires configuring the DMA\_CHCTLx.DNSEQEN bit to be active, and configuring the DMA\_CHCTLx.DINC bit to enable address update as increment or decrement.
2. This register is disabled when the channel's reset function is enabled.

### 16.5.23 Destination Address NonSequence Control Register B (DMA\_DNSEQCTLBx) (x=0-7)

Offset: 0x54+0x40\*x

| Offset | Bit Position  |    |    |    |    |    |    |    |    |    |    |    |               |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|---------------|----|----|----|----|----|----|----|----|----|----|----|---------------|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| '      | 31            | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19            | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000    |    |    |    |    |    |    |    |    |    |    |    |               |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | DNSCNTB[11:0] |    |    |    |    |    |    |    |    |    |    |    | DNSDIST[19:0] |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|        |               |    |    |    |    |    |    |    |    |    |    |    |               |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access | RW            |    |    |    |    |    |    |    |    |    |    |    | RW            |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit   | Name          | Function                                        | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | Access |
|-------|---------------|-------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:20 | DNSCNTB[11:0] | Destination address jump data size              | Set the amount of data transferred before the destination address jumps.<br>The destination device jumps the destination address by the DNSDIST-specified distance after transferring DNSCNTB blocks. The block size is determined by DMA_DTCTLx.BLKSIZE and DMA_CHCTLx.HSIZE.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | RW     |
| 19:0  | DNSDIST[19:0] | Destination nonSequence region address distance | When nonSequence address transmission, set the distance between two nonSequence areas of the destination device. The jump direction is forward or backward according to the DMA_CHCTLx.DINC bit. For details, refer to <a href="#">Figure 16-4 : Schematic Diagram of NonSequence Reset</a> . NonSequence address transfer is disabled when DMA_CHCTLx.DINC is set to fixed address.<br>The address distance is calculated based on the bit count set by the data width (DMA_CHCTLx.HSIZE) and the DNSDIST value.<br>Address Distance=DNSDIST×(DMA_CHCTLx.HSIZE(bit) /8)<br>For example, when DNSDIST is set to 10, and the data width is 32-bit, the address distance is 10×4=40; if 16-bit, the distance is 10×2=20; if 8-bit, the distance is 10×1=10.<br>The next transmission's destination address=starting address of the current destination nonSequence region ± address distance | RW     |



#### Note

1. NonSequence for the destination device requires configuring the DMA\_CHCTLx.DNSEQEN bit to be active, and configuring the DMA\_CHCTLx.DINC bit to enable address update as increment or decrement.
2. This register is only active when the channel's reset function is enabled, replacing the DMA\_DNSEQCTLx register. It is disabled when the channel's reset function is disabled.

### 16.5.24 Linked List Pointer Register (DMA\_LL Px) (x=0-7)

Offset: 0x58+0x40\*x

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |          |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|----------|
| '      | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0        |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |          |
| Name   | LLP[31:2]    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   | Reserved |
| Access | RW           |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |          |

| Bit  | Name      | Function            | Description                                                                                                                                    | Access |
|------|-----------|---------------------|------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:2 | LLP[31:2] | Linked list pointer | When the chain transmission is enabled, set the address of the next transmission descriptor to word alignment, that is, LLP[1:0] is fixed to 0 | RW     |
| 1:0  | Reserved  | Reserved            | -                                                                                                                                              | -      |

### 16.5.25 Channel Control Register (DMA\_CHCTLx) (x=0-7)

Offset: 0x5C+0x40\*x

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |            |    |          |    |        |    |       |    |       |    |         |         |        |    |        |    |      |    |      |    |    |    |    |    |    |    |    |    |    |    |  |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|------------|----|----------|----|--------|----|-------|----|-------|----|---------|---------|--------|----|--------|----|------|----|------|----|----|----|----|----|----|----|----|----|----|----|--|
| -      | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15         | 14 | 13       | 12 | 11     | 10 | 9     | 8  | 7     | 6  | 5       | 4       | 3      | 2  | 1      | 0  |      |    |      |    |    |    |    |    |    |    |    |    |    |    |  |
| Reset  | 0x00001000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |            |    |          |    |        |    |       |    |       |    |         |         |        |    |        |    |      |    |      |    |    |    |    |    |    |    |    |    |    |    |  |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | HPROT[3:2] |    | Reserved | IE | LLPRUN |    | LLPEN |    | HSIZE |    | DNSEQEN | SNSEQEN | DRPTEN |    | SRPTEN |    | DINC |    | SINC |    |    |    |    |    |    |    |    |    |    |    |  |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RW         |    |          | RW | RW     | RW | RW    | RW |       | RW | RW      | RW      | RW     | RW | RW     | RW | RW   | RW | RW   | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW |  |
|        |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |            |    |          |    |        |    |       |    |       |    |         |         |        |    |        |    |      |    |      |    |    |    |    |    |    |    |    |    |    |    |  |
|        |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |            |    |          |    |        |    |       |    |       |    |         |         |        |    |        |    |      |    |      |    |    |    |    |    |    |    |    |    |    |    |  |

| Bit   | Name       | Function                                | Description                                                                                                                                                                                                                                                                                                                                                                                                                                     | Access |
|-------|------------|-----------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:16 | Reserved   | Reserved                                | -                                                                                                                                                                                                                                                                                                                                                                                                                                               | -      |
| 15:14 | HPROT[3:2] | AHB Bus HPROT Value                     | HPROT[3:2] bits control AHB bus HPROT[3:2]. When using DMA to access DMC or SMC, please set HPROT[3:2] to 0b11; otherwise, please set HPROT[3:2] to 0b00.                                                                                                                                                                                                                                                                                       | RW     |
| 13    | Reserved   | Reserved                                | -                                                                                                                                                                                                                                                                                                                                                                                                                                               | -      |
| 12    | IE         | Interrupt enable bit                    | Configure whether the channel generates an interrupt <ul style="list-style-type: none"> <li>0b0: The channel does not generate an interrupt</li> <li>0b1: The channel generates an interrupt</li> </ul>                                                                                                                                                                                                                                         | RW     |
| 11    | LLPRUN     | Chain transfer mode selection           | When the chain transfer is active, set whether to start the transfer corresponding to the new descriptor immediately after loading the new descriptor specified by the linked list pointer when the current transfer is complete <ul style="list-style-type: none"> <li>0b0: Do not initiate transfer immediately, wait for the next transfer request</li> <li>0b1: Initiate transfer immediately after the new descriptor is loaded</li> </ul> | RW     |
| 10    | LLPEN      | Chain transfer enable                   | <ul style="list-style-type: none"> <li>0b0: Chain transfer is disabled</li> <li>0b1: Chain transfer is enabled</li> </ul>                                                                                                                                                                                                                                                                                                                       | RW     |
| 9:8   | HSIZE      | Transfer data width                     | <ul style="list-style-type: none"> <li>0b00: 8-bit</li> <li>0b01: 16-bit</li> <li>0b10/0b11: 32-bit</li> </ul>                                                                                                                                                                                                                                                                                                                                  | RW     |
| 7     | DNSEQEN    | Destination address NonSequence enabled | <ul style="list-style-type: none"> <li>0b0: NonSequence address transfer disabled</li> <li>0b1: NonSequence address transfer enabled</li> </ul>                                                                                                                                                                                                                                                                                                 | RW     |
| 6     | SNSEQEN    | Source address NonSequence enabled      | <ul style="list-style-type: none"> <li>0b0: NonSequence address transfer disabled</li> <li>0b1: NonSequence address transfer enabled</li> </ul>                                                                                                                                                                                                                                                                                                 | RW     |

| Bit | Name   | Function                              | Description                                                                                                                                                                                                                            | Access |
|-----|--------|---------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 5   | DRPTEN | Destination address repeat enable bit | Set whether to allow the destination address to reload the initial value <ul style="list-style-type: none"> <li>● 0b0: Do not reload the initial value</li> <li>● 0b1: Reload the initial value for address repeat transfer</li> </ul> | RW     |
| 4   | SRPTEN | Source address repeat enable bit      | Set whether to allow the source address to reload the initial value <ul style="list-style-type: none"> <li>● 0b0: Do not reload the initial value</li> <li>● 0b1: Reload the initial value for address repeat transfer</li> </ul>      | RW     |
| 3:2 | DINC   | Destination address update method     | <ul style="list-style-type: none"> <li>● 0b00: Fixed</li> <li>● 0b01: Incrementing</li> <li>● 0b10/0b11: Decrementing</li> </ul>                                                                                                       | RW     |
| 1:0 | SINC   | Source address update method          | <ul style="list-style-type: none"> <li>● 0b00: Fixed</li> <li>● 0b01: Incrementing</li> <li>● 0b10/0b11: Decrementing</li> </ul>                                                                                                       | RW     |

### 16.5.26 Source Address Monitor Register (DMA\_MONSARx) (x=0-7)

Offset: 0x60+0x40\*x

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| -      | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | SAR          |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access | RO           |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit  | Name | Function                | Description                                                                                                                                                                                                                                                                                                                                                                        | Access |
|------|------|-------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:0 | SAR  | Transfer source address | Set source address<br><div>  <b>Note</b> <ul style="list-style-type: none"> <li>When the transfer data width is 16-bit, i.e., DMA_CHCTLx.HSIZE=0b01, SAR[0] is invalid.</li> <li>When the transfer data width is 32-bit, i.e., DMA_CHCTLx.HSIZE=0b10/0b11, SAR[1:0] is invalid.</li> </ul> </div> | RO     |



#### Note

The DMA\_SARx register remains unchanged before and after DMA transfer, while the DMA\_MONSARx register updates after each data block transfer by DMA. DMA\_MONSARx.SAR updates to the next transmission address based on the fixed/incrementing/decrementing/repeated jump/nonSequence jump settings.

### 16.5.27 Destination Address Monitor Register (DMA\_MONDARx) (x=0-7)

Offset: 0x64+0x40\*x

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| -      | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | DAR          |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access | RO           |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit  | Name | Function                         | Description                                                                                                                                                                                                                                                                                                                                                                             | Access |
|------|------|----------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:0 | DAR  | Transmission destination address | Set transmission destination address<br> <b>Note</b> <ul style="list-style-type: none"> <li>When the transfer data width is 16-bit, i.e., DMA_CHCTLx.HSIZE=0b01, DAR[0] is invalid.</li> <li>When the transfer data width is 32-bit, i.e., DMA_CHCTLx.HSIZE=0b10/0b11, DAR[1:0] is invalid.</li> </ul> | RO     |

#### Note

The DMA\_DARx register remains unchanged before and after DMA transfer, while the DMA\_MONDARx register updates after each block transfer. DMA\_MONDARx.DAR updates to the next transmission address based on the fixed/incrementing/decrementing/repeated jump/nonSequence jump settings.

## 16.5.28 Transfer Data Control Monitor Register (DMA\_MONDTCTLx) (x=0-7)

Offset: 0x68+0x40\*x

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |          |    |    |    |              |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----------|----|----|----|--------------|----|---|---|---|---|---|---|---|---|---|---|
| '      | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15       | 14 | 13 | 12 | 11           | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000001   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |          |    |    |    |              |    |   |   |   |   |   |   |   |   |   |   |
| Name   | CNT[15:0]    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | Reserved |    |    |    | BLKSIZE[9:0] |    |   |   |   |   |   |   |   |   |   |   |
|        |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |          |    |    |    |              |    |   |   |   |   |   |   |   |   |   |   |
| Access | RO           |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |          |    |    |    | RO           |    |   |   |   |   |   |   |   |   |   |   |

| Bit   | Name         | Function        | Description                                                                                                                                                                                                           | Access |
|-------|--------------|-----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:16 | CNT[15:0]    | Transfer count  | The number of transfers remaining<br> <b>Note</b><br>When the transfer count is 0 (DMA_DTCTLx.CNT=0x0000), this bit field remains 0. | RO     |
| 15:10 | Reserved     | Reserved        | -                                                                                                                                                                                                                     | -      |
| 9:0   | BLKSIZE[9:0] | Data block size | Data block size<br> <b>Note</b><br>When this bit field is 0, it indicates that 1024 data items are transferred per transaction.    | RO     |



### 16.5.29 Address Repeat Control Monitor Register (DMA\_MONRPTx) (x=0-7)

Offset: 0x6C+0x40\*x

| Offset | Bit Position |    |    |    |    |    |           |    |    |    |    |    |    |    |          |    |    |    |    |    |           |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|-----------|----|----|----|----|----|----|----|----------|----|----|----|----|----|-----------|----|---|---|---|---|---|---|---|---|---|---|
| '      | 31           | 30 | 29 | 28 | 27 | 26 | 25        | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17       | 16 | 15 | 14 | 13 | 12 | 11        | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |           |    |    |    |    |    |    |    |          |    |    |    |    |    |           |    |   |   |   |   |   |   |   |   |   |   |
| Name   | Reserved     |    |    |    |    |    | DRPT[9:0] |    |    |    |    |    |    |    | Reserved |    |    |    |    |    | SRPT[9:0] |    |   |   |   |   |   |   |   |   |   |   |
| Access |              |    |    |    |    |    | RO        |    |    |    |    |    |    |    |          |    |    |    |    |    | RO        |    |   |   |   |   |   |   |   |   |   |   |

| Bit   | Name      | Function                               | Description | Access |
|-------|-----------|----------------------------------------|-------------|--------|
| 31:26 | Reserved  | Reserved                               | -           | -      |
| 25:16 | DRPT[9:0] | Destination address repeat region size | -           | RO     |
| 15:10 | Reserved  | Reserved                               | -           | -      |
| 9:0   | SRPT[9:0] | Source address repeat zone size        | -           | RO     |



#### Note

The (DMA\_RPTx) register remains unchanged before and after DMA transfer, while the (DMA\_MONRPTx) register updates after each block transfer.

DMA\_MONRPTx.SRPT[9:0] and DMA\_MONRPTx.DRPT[9:0] update as follows:

- When channel reset is disabled: subtract the block size, and reload the DMA\_RPTx setting value when it is reduced to 0.
- When channel reset is enabled: Decrement by 1, reload the initial value when it reaches 0.

### 16.5.30 Source Address NonSequence Control Monitor Register (DMA\_MONSNSEQCTLx) (x=0-7)

Offset: 0x70+0x40\*x

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |               |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |  |  |  |  |  |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---------------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|--|--|--|--|--|--|
| -      | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15            | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |  |  |  |  |  |  |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |               |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |  |  |  |  |  |
| Name   | SNSCNT[11:0] |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | SOFFSET[19:0] |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |  |  |  |  |  |
| Access | RO           |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RO            |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |  |  |  |  |  |

| Bit   | Name          | Function                      | Description                                                       | Access |
|-------|---------------|-------------------------------|-------------------------------------------------------------------|--------|
| 31:20 | SNSCNT[11:0]  | Source address jump data size | Set the amount of data transferred before the source address jump | RO     |
| 19:0  | SOFFSET[19:0] | Source address jump offset    | Source address offset                                             | RO     |



#### Note

The DMA\_SNSEQCTLx register remains unchanged before and after DMA transfer, while the DMA\_MONSNSEQCTLx register updates after each data block transfer by DMA.

Update method for DMA\_MONSNSEQCTLx.SNSCNT[11:0]:

- When channel reset is disabled: Subtract the block size, reload initial value of DMA\_SNSEQCTLx when it is reduced to 0.
- When channel reset is enabled: Decrement by 1, reload the initial value when it reaches 0.

### 16.5.31 Destination Address NonSequence Control Monitor Register (DMA\_MONDNSEQCTLx) (x=0-7)

Offset: 0x74+0x40\*x

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |               |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |  |  |  |  |  |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---------------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|--|--|--|--|--|--|
| -      | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15            | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |  |  |  |  |  |  |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |               |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |  |  |  |  |  |
| Name   | DNSCNT[11:0] |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | DOFFSET[19:0] |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |  |  |  |  |  |
| Access | RO           |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RO            |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |  |  |  |  |  |

| Bit   | Name          | Function                                | Description                                                            | Access |
|-------|---------------|-----------------------------------------|------------------------------------------------------------------------|--------|
| 31:20 | DNSCNT[11:0]  | Destination address jump data size      | Set the amount of data transferred before the destination address jump | RO     |
| 19:0  | DOFFSET[19:0] | Destination address jump address offset | Destination address offset                                             | RO     |



#### Note

The DMA\_DNSEQCTLx register remains unchanged before and after DMA transfer, while the DMA\_MONDNSEQCTLx register updates after each data block transfer by DMA.

Update method for DMA\_MONDNSEQCTLx.DNSCNT[11:0]:

- When channel reset is disabled: Subtract block size, reload initial value of DMA\_DNSEQCTLx when it is reduced to 0.
- When channel reset is enabled: Decrement by 1, reload the initial value when it reaches 0.

## 16.6 Precautions

- DMA registers only support 32-bit read and write operations, 8/16-bit read and write operations are inactive.

## 17 Voltage Comparator (CMP)

### 17.1 Introduction

Voltage Comparator (hereinafter referred to as CMP) is a peripheral module that compares two analog voltages and outputs the comparison result. This product is equipped with two groups of 4 comparison channels: CMP1 and CMP2, CMP3 and CMP4.

### 17.2 Features

- 4 comparison channels can independently perform normal comparison.
- Two comparison channels in the same group can be combined to achieve up to 2 window comparisons.
- Each comparison channel's positive and negative voltage inputs have multiple input sources (IO, PGA, DAC) to choose from.
- Hysteresis voltage is configurable.
- A noise filter can filter the comparator output, with 3 sampling clocks to choose from.
- Timer PWM can be used for comparator blanking window control.
- Interrupts can be generated at the edges of comparison result changes, other peripherals can be triggered, and STOP mode can be woken up.
- Compare results can be monitored through registers and output to the external pin VCOUT.
- Compare results can be used for EMB control events.

## 17.3 Functional Description

### 17.3.1 Function Block Diagram

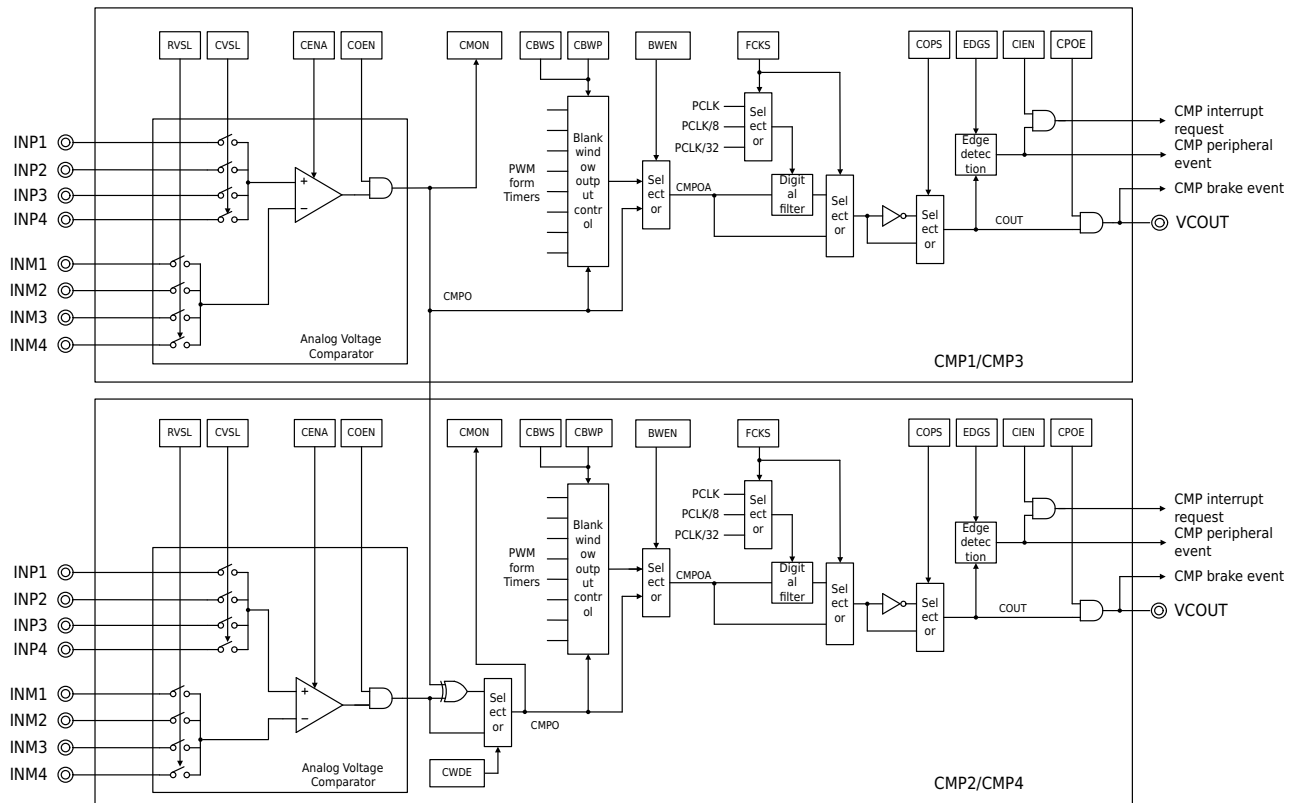


Figure 17-1 CMP Function Block Diagram

Table 17-1 CMP Pin List

| Pin Name    | Input/Output | Function                     |
|-------------|--------------|------------------------------|
| AVCC        | Input        | Analog Power                 |
| AVSS        | Input        | Analog Ground                |
| CMP1_INP2   | Input        | CMP1 Positive Analog Input   |
| CMP1_INP3   | Input        | CMP1 Positive Analog Input   |
| CMP1_INP4   | Input        | CMP1 Positive Analog Input   |
| CMP123_INM3 | Input        | CMP1-3 Negative Analog Input |
| CMP1_INM4   | Input        | CMP1 Negative Analog Input   |
| CMP2_INP3   | Input        | CMP2 Positive Analog Input   |
| CMP2_INP4   | Input        | CMP2 Positive Analog Input   |
| CMP2_INM4   | Input        | CMP2 Negative Analog Input   |
| CMP3_INP2   | Input        | CMP3 Positive Analog Input   |
| CMP3_INP3   | Input        | CMP3 Positive Analog Input   |
| CMP3_INP4   | Input        | CMP3 Positive Analog Input   |
| CMP3_INM4   | Input        | CMP3 Negative Analog Input   |

| Pin Name   | Input/Output | Function                    |
|------------|--------------|-----------------------------|
| CMP4_INP3  | Input        | CMP4 Positive Analog Input  |
| CMP4_INP4  | Input        | CMP4 Positive Analog Input  |
| CMP4_INM3  | Input        | CMP4 Negative Analog Input  |
| CMP4_INM4  | Input        | CMP4 Negative Analog Input  |
| CMP1_VCOUT | Output       | CMP1 External Port Output   |
| CMP2_VCOUT | Output       | CMP2 External Port Output   |
| CMP3_VCOUT | Output       | CMP3 External Port Output   |
| CMP4_VCOUT | Output       | CMP4 External Port Output   |
| CMP_VCOUT  | Output       | CMP1-4 External Port Output |

### 17.3.2 Input Selection

The analog inputs of CMP1-4 are selected according to the following table.

Table 17-2 CMP Positive Input List

| Compare channels | CVSL=0b0001 (INP1) | CVSL=0b0010 (INP2) |           | CVSL=0b0100 (INP3) |           | CVSL=0b1000 (INP4) |
|------------------|--------------------|--------------------|-----------|--------------------|-----------|--------------------|
| CMP1             | PGA1_BP            | P2SL=0b001         | PGA1      | P3SL=0b01          | CMP1_INP3 | CMP1_INP4          |
|                  |                    | P2SL=0b010         | PGA2      | P3SL=0b10          | CMP2_INP3 |                    |
|                  |                    | P2SL=0b100         | CMP1_INP2 | -                  | -         |                    |
| CMP2             | PGA2_BP            | PGA2               |           | CMP2_INP3          |           | CMP2_INP4          |
| CMP3             | PGA3_BP            | P2SL=0b001         | PGA3      | P3SL=0b01          | CMP3_INP3 | CMP3_INP4          |
|                  |                    | P2SL=0b010         | PGA4      | P3SL=0b10          | CMP4_INP3 |                    |
|                  |                    | P2SL=0b100         | CMP3_INP2 | -                  | -         |                    |
| CMP4             | PGA4_BP            | PGA4               |           | CMP4_INP3          |           | CMP4_INP4          |



#### Note

In the table, PGA1\_BP/PGA2\_BP/PGA3\_BP/PGA4\_BP are the unamplified voltage signals output by PGA1-4, respectively. When using them, PGA should also be set to active.

Table 17-3 CMP Negative Input List

| Compare channels | RVSL=0b0001 (INM1) | RVSL=0b0010 (INM2) | RVSL=0b0100 (INM3) | RVSL=0b1000 (INM4) |
|------------------|--------------------|--------------------|--------------------|--------------------|
| CMP1             | DA1O1              | DA1O2              | CMP123_INM3        | CMP1_INM4          |
| CMP2             | DA1O1              | DA1O2              | CMP123_INM3        | CMP2_INM4          |
| CMP3             | DA2O1              | DA2O2              | CMP123_INM3        | CMP3_INM4          |
| CMP4             | DA2O1              | DA2O2              | CMP4_INM3          | CMP4_INM4          |

### 17.3.3 Normal Compare

CMP 1-4 can independently perform normal comparison. Taking CMP1 as an example, the setting steps are as follows:

**Step1.** Write 0 to PWC\_FCG3.Bit3 and wait for at least 2μs.

**Step2.** Enable CMP1 module function (PWC\_FCG3.CMP12 write 0).

- Step3.** Set CMP\_PMSR.CVSL to select the positive voltage.  
Set CMP\_PMSR.RVSL to select the negative voltage.
- Step4.** Set CMP\_MDR.CENA to 1 to enable the comparator.
- Step5.** Wait for the comparator to stabilize  $t_{CMP}$  (for specific values, see the "Electrical Characteristics" in the "Datasheet").
- Step6.** Set CMP\_FIR.FCKS to set the sampling period for digital filtering.  
Set CMP\_FIR.EDGS and CMP\_FIR.CIEN to set the edge where the interrupt occurs.
- Step7.** Set CMP\_OCR.CPOE and CMP\_OCR.COPS to set the output of VCOUT.
- Step8.** Set CMP\_OCR.COEN to 1 to allow comparator output.

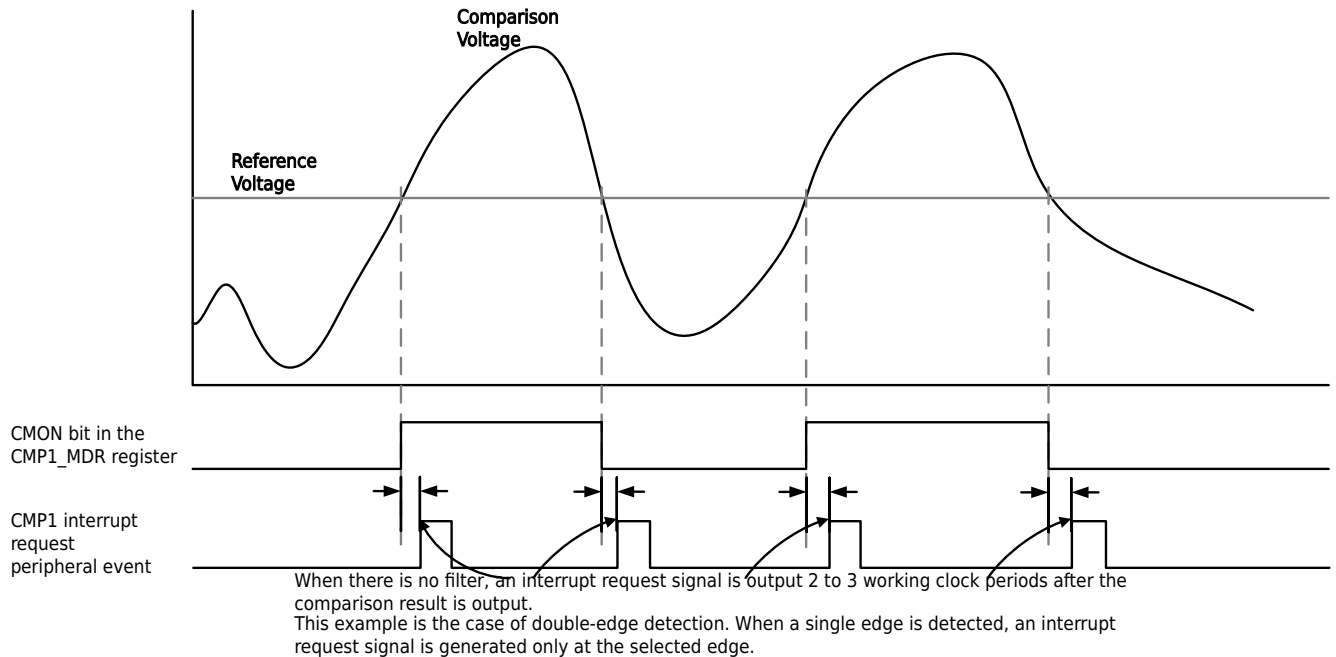


**Figure 17-2 Voltage Comparison Diagram**

As shown in the figure above, during non-inverting output, if the positive voltage is higher than the negative voltage after CMP output is enabled, the CMON bit in the CMP\_MDR register is 1; if the positive voltage is lower than the negative voltage, the CMON bit is 0. During inverting output, the result of CMON is opposite to that during non-inverting output.

For CMP1 and CMP3, if INP2 or INP3 is selected as the positive voltage, then the voltage input source also needs to be selected by setting CMP\_VISR. Please see for [Table 17-2 : CMP Positive Input List](#) details.

### 17.3.4 Window Compare

Window comparison can be achieved when CMP1 and CMP2 or CMP3 and CMP4 operate simultaneously; Taking CMP1 and CMP2 as an example, the setting steps are as follows:

- Step1.** Write 0 to PWC\_FCG3.Bit3 and wait for at least 2 $\mu$ s.
- Step2.** Enable CMP1 and CMP2 module functions (PWC\_FCG3.CMP12 write 0).
- Step3.** Set CMP1\_PMSR.CVSL and select the positive terminal voltage.  
Set CMP1\_PMSR.RVSL to select the negative voltage as the lower limit of the window.
- Step4.** Set CMP2\_PMSR.CVSL to select the same positive terminal voltage as CMP1.  
Set CMP2\_PMSR.RVSL to select the negative terminal voltage as the upper limit of the window.
- Step5.** Set CMP2\_MDR.WDEN to 1 to start window comparison;
- Step6.** Set CMP1\_MDR.CENA and CMP2\_MDR.CENA to 1 to start the comparators.

- Step7.** Wait for the stabilization time  $t_{CMP}$  (refer to the "Electrical Specifications" section of the Data-sheet for specific values).
- Step8.** Set CMP2\_FIR.FCKS to set the digital filter.  
Set CMP2\_FIR.EDGS, CMP2\_FIR.CIEN to set interrupt and edge detection conditions.
- Step9.** Set CMP2\_OCR.CPOE, CMP2\_OCR.COPS to set the output of VCOOUT;
- Step10.** Set CMP1\_OCR.COEN and CMP2\_OCR.COEN to 1 to allow comparator output.

During window comparison, CMP2 completes the monitoring, filtering, interrupting, and outputting of comparison results.

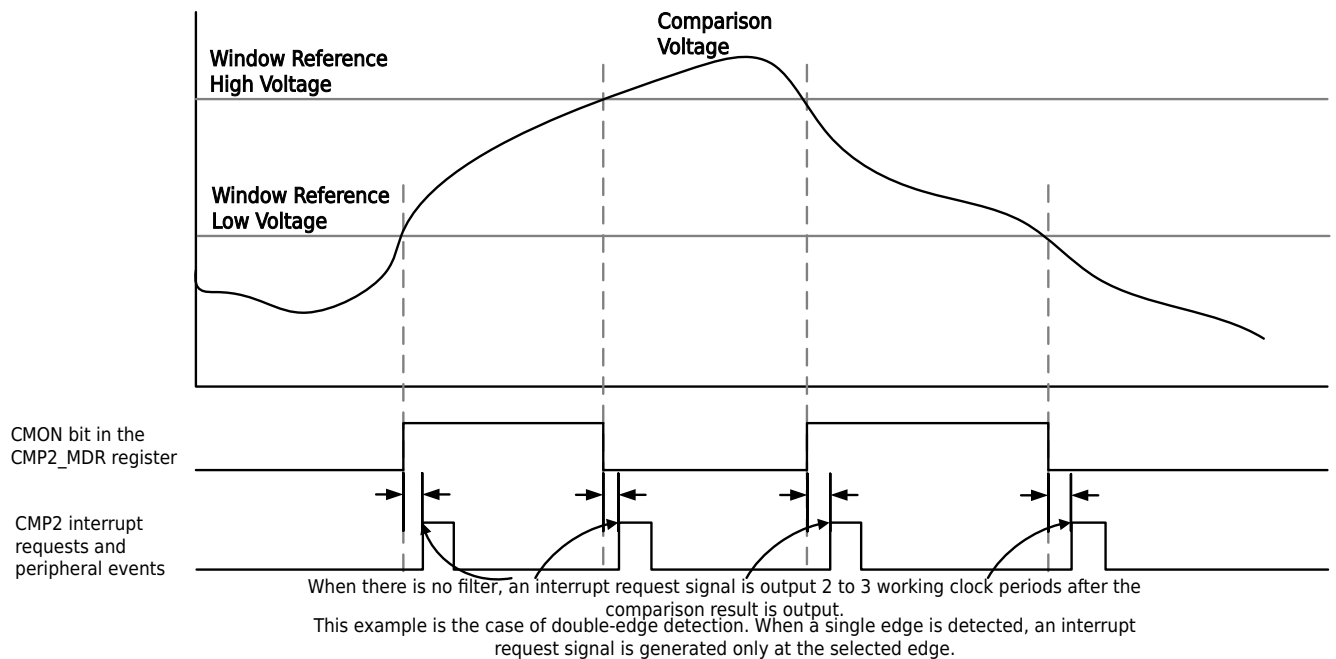


**Figure 17-3 Window Compare Diagram**

As shown in the figure above, during non-inverting output, if the positive voltage is within the window composed of the two negative voltages after CMP output is enabled, the CMON bit in the CMP2\_MDR register is 1; if the positive voltage is outside the window composed of the two negative voltages, the CMON bit is 0. During inverting output, the result of CMON is opposite to that during non-inverting output.

### 17.3.5 Blank Window

The CMP can use the outputs of Timer4, Timer6, TimerA as Blank-Window signals to control the windowed output of the comparison result CMPOA, providing convenience for applications such as timer PWM zero-crossing detection.

To enable blank window output (set the CMP\_OCR.BWEN bit to 1), select one or more timer PWMs as Blank-Windows through the CMP\_BWSR.CBWS bit, and select the active level for each Blank-Window through the CMP\_BWPR.CBWP bit. In this way, when a Blank-Window is active, the window output CMPOA is fixed to the state set by the CMP\_OCR.BWOL bit.



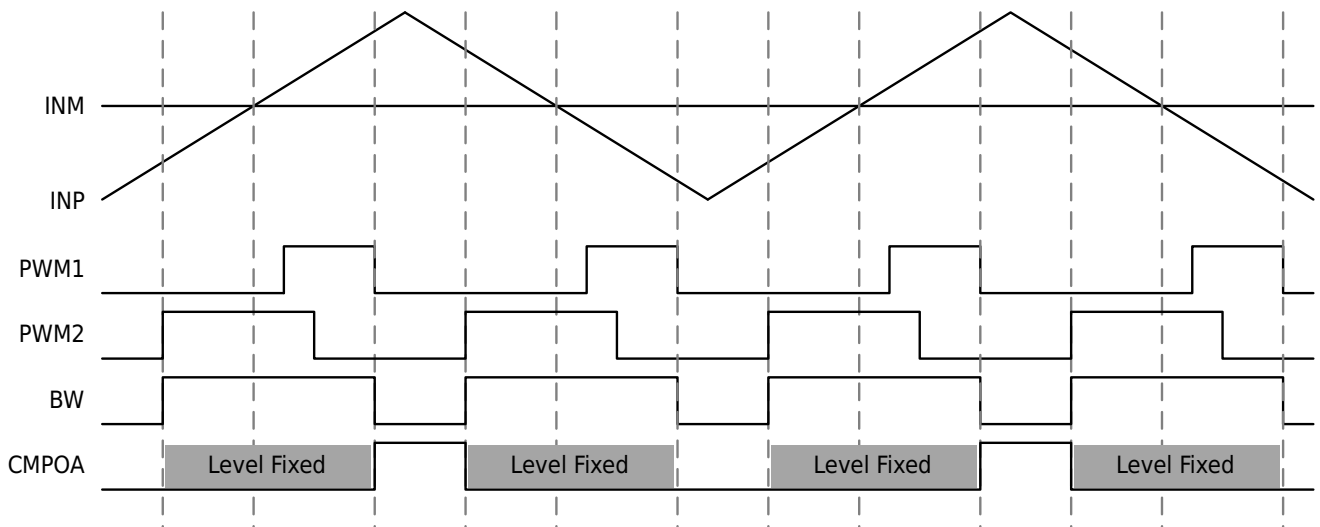


Figure 17-4 Blank Window Output Diagram

As shown in the figure above, two PWM signals are selected as the Blank-Window, both configured as active-high. When the Blank-Window is enabled, the output state of CMPOA is forced to low level. Thus, when the Blank-Window signal is at a high level, CMPOA is forced to a low level; when Blank-Window is at a low level, CMPOA outputs the normal comparator result.

### 17.3.6 Hysteresis Function

To avoid false outputs caused by noise signals, each comparison channel has a configurable hysteresis function, and the hysteresis voltage can be selected through the CMP\_MDR.HYST bit. The hysteresis function is disabled when the HYST bit is set to 0b000.

### 17.3.7 Noise Filter

Each comparison channel contains a noise filter. The noise filter samples the comparator output CMPOA during each sampling clock period. If the same value is sampled three consecutive times, the value is output to COUT in the next sampling clock period; otherwise, COUT remains unchanged. The sampling clock can be selected through the CMP\_FIR.FCKS bit. The noise filter is disabled when the FCKS bit is set to 0b00.

### 17.3.8 External Pin Output

The comparison result COUT output by the noise filter can be output to the external port VCOU. Whether VCOU output is enabled and the output polarity (non-inverting output or inverting output) can be set through the CMP\_OCR.CPOE and COPS bits. After reset, the port corresponding to VCOU is a general-purpose port in input status. Therefore, it must also be set to VCOU output function through the port register.

### 17.3.9 Interrupt and Event

Interrupt requests can be generated for the results of all comparison channels. When using, the CMP\_FIR.CIEN bit must be set to 1, and the edge of the comparison result that can generate an interrupt is set through the CMP\_FIR.EDGS bit (value other than 0b00). If noise filtering and inverting output are used, edge detection is performed on the signal after noise filtering and inverting output.

When INTC\_WKEN.CMP\*WKEN=1, CMPO can wake up the stop low power mode. The system will be woken up and an interrupt will be generated when CMPO changes from low to high.

The generation conditions for peripheral trigger events and interrupt requests are the same, but the event trigger function must be enabled or disabled through the hardware trigger source register of the target peripheral, independent of the CIEN bit.

## 17.4 Register Description

### 17.4.1 Register List

- CMP1 base address: 0x4004A000
- CMP2 base address: 0x4004A010
- CMP3 base address: 0x4004A400
- CMP4 base address: 0x4004A410

Table 17-4 Voltage Comparator (CMP) Register Offset

| Offset | Register                 | Description                                                                | Width |
|--------|--------------------------|----------------------------------------------------------------------------|-------|
| 0x00   | <a href="#">CMP_MDR</a>  | Comparator Working Mode Register                                           | 8     |
| 0x01   | <a href="#">CMP_FIR</a>  | Comparator Filter and Interrupt Register                                   | 8     |
| 0x02   | <a href="#">CMP_OCR</a>  | Comparator Output Control Register                                         | 8     |
| 0x03   | <a href="#">CMP_PMSR</a> | Comparator Positive and Negative Input Selection Register                  | 8     |
| 0x04   | <a href="#">CMP_BWSR</a> | Comparator Blank Window Output Source Register                             | 16    |
| 0x06   | <a href="#">CMP_BWPR</a> | Comparator Blank Window Output Polarity Register                           | 16    |
| 0x08   | <a href="#">CMP_VISR</a> | Comparator Voltage Input Selection Register (only exists in CMP1 and CMP3) | 16    |

## 17.4.2 Comparator Working Mode Register (CMP\_MDR)

| Offset | Bit Position |      |   |   |          |   |      |      |
|--------|--------------|------|---|---|----------|---|------|------|
| 0x00   | 7            | 6    | 5 | 4 | 3        | 2 | 1    | 0    |
| Reset  | 0x00         |      |   |   |          |   |      |      |
| Name   | CMON         | HYST |   |   | Reserved |   | WDEN | CENA |
| Access | RO           | RW   |   |   |          |   | RW   | RW   |

| Bit | Name     | Function                     | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | Access |
|-----|----------|------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 7   | CMON     | Compare Result Monitoring    | <p>Normal Comparative Mode:</p> <ul style="list-style-type: none"> <li>0b0: Positive Side Voltage &lt; Negative Side Voltage</li> <li>0b1: Positive Side Voltage &gt; Negative Side Voltage</li> </ul> <p>Window Compare Mode (This bit is only active for CMP2 and CMP4) :</p> <ul style="list-style-type: none"> <li>0b0: Positive voltage &lt; lower limit negative voltage, or positive voltage &gt; upper limit negative voltage</li> <li>0b1: Lower limit negative voltage &lt; positive voltage &lt; upper limit negative voltage</li> </ul> | RO     |
| 6:4 | HYST     | Hysteresis Voltage Selection | <ul style="list-style-type: none"> <li>0b000: 0mV (no hysteresis)</li> <li>0b001: 10mV</li> <li>0b010: 20mV</li> <li>0b011: 30mV</li> <li>0b100: 40mV</li> <li>0b101: 50mV</li> <li>0b110: 60mV</li> <li>0b111: 70mV</li> </ul> <div>  <b>Note</b><br/>           The above are typical values, and the error range can be found in the "Electrical Specifications" section of the Datasheet.         </div>                                                     | RW     |
| 3:2 | Reserved | Reserved                     | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | -      |
| 1   | WDEN     | Window Compare Selection     | <ul style="list-style-type: none"> <li>0b0: Normal compare mode</li> <li>0b1: Window compare mode</li> </ul> <div>  <b>Note</b><br/>           This bit exists only in CMP2 and CMP4. After setting to window compare mode, the filtering, edge detection, and output control for the window comparison results are all performed by CMP2 and CMP4.         </div>                                                                                               | RW     |

| Bit | Name | Function                        | Description                                                                                                                                                                                                                                                                                                                                                   | Access |
|-----|------|---------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 0   | CENA | Comparator Operation Permission | <ul style="list-style-type: none"><li>● 0b0: Comparator stopped</li><li>● 0b1: Comparator working</li></ul> <div> <b>Note</b><br/>After setting CENA from 0 to 1 each time, please wait for about 300ns of stabilization time before performing subsequent operations.</div> | RW     |

## 17.4.3 Comparator Filter and Interrupt Register (CMP\_FIR)

| Offset | Bit Position |      |      |   |          |   |      |   |
|--------|--------------|------|------|---|----------|---|------|---|
| 0x01   | 7            | 6    | 5    | 4 | 3        | 2 | 1    | 0 |
| Reset  | 0x00         |      |      |   |          |   |      |   |
| Name   | Reserved     | CIEN | EDGS |   | Reserved |   | FCKS |   |
| Access |              | RW   | RW   |   |          |   | RW   |   |

| Bit | Name            | Function                                     | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | Access |
|-----|-----------------|----------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 7   | <i>Reserved</i> | <i>Reserved</i>                              | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | -      |
| 6   | CIEN            | Comparator Interrupt Permission              | <ul style="list-style-type: none"> <li>0b0: Comparator interrupt disabled</li> <li>0b1: Comparator interrupt enabled</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | RW     |
| 5:4 | EDGS            | Comparator Output Edge Detection Selection   | <ul style="list-style-type: none"> <li>0b00: Not detect edges of comparator output COUT</li> <li>0b01: Detect rising edge of comparator output COUT</li> <li>0b10: Detect falling edge of comparator output COUT</li> <li>0b11: Detect rising and falling edges of comparator output COUT</li> </ul> <div>  <b>Note</b><br/>           Changing EDGS[1:0] may cause a interrupt or peripheral trigger event, so set the register if the interrupt disable or peripheral trigger is disabled. After register settings, please clear the corresponding interrupt flag.         </div>                                                                                                                                                                                                                            | RW     |
| 3:2 | <i>Reserved</i> | <i>Reserved</i>                              | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | -      |
| 1:0 | FCKS            | Comparator Result Filtering Sample Selection | <ul style="list-style-type: none"> <li>0b00: Noise filter not used</li> <li>0b01: Noise filter used, sampling through PCLK3</li> <li>0b10: Noise filter used, sampling through PCLK3/8</li> <li>0b11: Noise filter used, sampling through PCLK3/32</li> </ul> <div>  <b>Note</b><br/>           Modify FCKS[1:0] only when the comparator output is disabled (i.e., CMP_OCR.COEN=0). When switching FCKS[1:0] from 0b00 to other values, please use the filtered output after 4 samples as the interrupt request or peripheral trigger event.<br/><br/>           Changing FCKS[1:0] may cause a interrupt or peripheral trigger event, so set the register if the interrupt disable or peripheral trigger is disabled. After register settings, please clear the corresponding interrupt flag.         </div> | RW     |

## 17.4.4 Comparator Output Control Register (CMP\_OCR)

| Offset | Bit Position |   |   |      |      |      |      |      |
|--------|--------------|---|---|------|------|------|------|------|
| 0x02   | 7            | 6 | 5 | 4    | 3    | 2    | 1    | 0    |
| Reset  | 0x00         |   |   |      |      |      |      |      |
| Name   | Reserved     |   |   | BWOL | BWEN | CPOE | COPS | COEN |
| Access |              |   |   | RW   | RW   | RW   | RW   | RW   |

| Bit | Name            | Function                             | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | Access |
|-----|-----------------|--------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 7:5 | <i>Reserved</i> | <i>Reserved</i>                      | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | -      |
| 4   | BWOL            | Blank-Window Output Level            | <ul style="list-style-type: none"> <li>0b0: CMPOA output low during active Blank-Window</li> <li>0b1: CMPOA output high during active Blank-Window</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                   | RW     |
| 3   | BWEN            | Blank-Window Enable                  | <ul style="list-style-type: none"> <li>0b0: Disable</li> <li>0b1: Enable</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | RW     |
| 2   | CPOE            | VCOUT Output Enable                  | <ul style="list-style-type: none"> <li>0b0: VCOUT output disabled</li> <li>0b1: VCOUT output enabled</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | RW     |
| 1   | COPS            | Comparator Output Polarity Selection | <ul style="list-style-type: none"> <li>0b0: Comparator non-inverting output</li> <li>0b1: Comparator inverting output</li> </ul> <div>  <b>Note</b><br/> Please rewrite COPS when the comparator output is disabled (i.e., COEN bit is 0). Changing the COPS bit may cause an interrupt or peripheral trigger event, so please set this register in the state where the interrupt is disabled or the peripheral trigger function is disabled. After register settings, please clear the corresponding interrupt flag. </div> | RW     |
| 0   | COEN            | Comparator Output Permissible        | <ul style="list-style-type: none"> <li>0b0: Comparator output disabled (comparator output fixed at low level)</li> <li>0b1: Comparator output enabled</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                | RW     |

## 17.4.5 Comparator Positive and Negative Input Selection Register (CMP\_PMSR)

| Offset | Bit Position |   |   |   |      |   |   |   |
|--------|--------------|---|---|---|------|---|---|---|
| 0x03   | 7            | 6 | 5 | 4 | 3    | 2 | 1 | 0 |
| Reset  | 0x00         |   |   |   |      |   |   |   |
| Name   | CVSL         |   |   |   | RVSL |   |   |   |
| Access | RW           |   |   |   | RW   |   |   |   |

| Bit | Name | Function                 | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | Access |
|-----|------|--------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 7:4 | CVSL | Positive Input Selection | <p>Select the positive input voltage of the voltage comparator. For details of the selected input, please refer to the <a href="#">CMP Positive Input</a>:</p> <ul style="list-style-type: none"> <li>● 0b0000: No positive voltage input</li> <li>● 0b0001: Select INP1 as positive voltage</li> <li>● 0b0010: Select INP2 as positive voltage</li> <li>● 0b0100: Select INP3 as positive voltage</li> <li>● 0b1000: Select INP4 as positive voltage</li> <li>● Others: Setting prohibited</li> </ul> <div>  <b>Note</b><br/> Please rewrite CVSL when CMP_OCR.COEN is 0. After rewriting CVSL each time, please wait for 300ns before setting CMP_OCR.COEN to 1. Changing the CVSL may cause a interrupt or peripheral trigger event, so set the register if the interrupt disable or peripheral trigger is disabled. After register settings, please clear the corresponding interrupt flag. </div>                                             | RW     |
| 3:0 | RVSL | Negative Input Selection | <p>Select the negative input voltage of the voltage comparator. For details of the selected input, please refer to the <a href="#">CMP Negative Input List</a>:</p> <ul style="list-style-type: none"> <li>● 0b0000: No negative voltage input</li> <li>● 0b0001: Select INM1 as negative voltage</li> <li>● 0b0010: Select INM2 as negative voltage</li> <li>● 0b0100: Select INM3 as negative voltage</li> <li>● 0b1000: Select INM4 as negative voltage</li> <li>● Others: Setting prohibited</li> </ul> <div>  <b>Note</b><br/> Please rewrite RVSL when CMP_OCR.COEN is 0. After rewriting RVSL each time, please wait for 300ns before setting CMP_OCR.COEN to 1. Changing RVSL may cause an interrupt or peripheral trigger event, so please set this register in the state where the interrupt is disabled or the peripheral trigger function is disabled. After register settings, please clear the corresponding interrupt flag. </div> | RW     |



### 17.4.6 Comparator Voltage Input Selection Register (CMP\_VISR) (Only exists in CMP1 and CMP3)

| Offset | Bit Position |    |    |    |    |    |   |   |   |   |      |          |      |   |   |   |
|--------|--------------|----|----|----|----|----|---|---|---|---|------|----------|------|---|---|---|
| 0x08   | 15           | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5    | 4        | 3    | 2 | 1 | 0 |
| Reset  | 0x0000       |    |    |    |    |    |   |   |   |   |      |          |      |   |   |   |
| Name   | Reserved     |    |    |    |    |    |   |   |   |   | P3SL | Reserved | P2SL |   |   |   |
| Access |              |    |    |    |    |    |   |   |   |   | RW   |          | RW   |   |   |   |

| Bit  | Name     | Function             | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | Access |
|------|----------|----------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 15:6 | Reserved | Reserved             | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | -      |
| 5:4  | P3SL     | INP3 Input Selection | <p><b>CMP1:</b> Select the input voltage when CMP1_PMSR.CVSL[3:0] is 0b0100</p> <ul style="list-style-type: none"> <li>0b00: No voltage input</li> <li>0b01/0b10: The selected input is detailed in the <a href="#">Table 17-2 : CMP Positive Input List</a></li> <li>0b11: Setting prohibited</li> </ul> <p><b>CMP3:</b> Select the input voltage when CMP3_PMSR.CVSL[3:0] is 0b0100</p> <ul style="list-style-type: none"> <li>0b00: No voltage input</li> <li>0b01/0b10: The selected input is detailed in the <a href="#">Table 17-2 : CMP Positive Input List</a></li> <li>0b11: Setting prohibited</li> </ul> <div>  <b>Notice</b><br/>Please rewrite P3SL when CMP_PMSR.CVSL[2] is 0.         </div>                      | RW     |
| 3    | Reserved | Reserved             | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | -      |
| 2:0  | P2SL     | INP2 Input Selection | <p><b>CMP1:</b> Select the input voltage when CMP1_PMSR.CVSL[3:0] is 0b0010</p> <ul style="list-style-type: none"> <li>0b000: No voltage input</li> <li>0b001/0b010/0b100: The selected input is detailed in the <a href="#">Table 17-2 : CMP Positive Input List</a></li> <li>Others: Setting prohibited</li> </ul> <p><b>CMP3:</b> Select the input voltage when CMP3_PMSR.CVSL[3:0] is 0b0010</p> <ul style="list-style-type: none"> <li>0b00: No voltage input</li> <li>0b001/0b010/0b100: The selected input is detailed in the <a href="#">Table 17-2 : CMP Positive Input List</a></li> <li>Others: Setting prohibited</li> </ul> <div>  <b>Notice</b><br/>Please rewrite P2SL when CMP_PMSR.CVSL[1] is 0.         </div> | RW     |

## 17.4.7 Comparator Blank Window Output Source Register(CMP\_BWSR)

| Offset | Bit Position |        |        |        |        |        |       |       |       |       |       |       |       |       |       |       |
|--------|--------------|--------|--------|--------|--------|--------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| 0x04   | 15           | 14     | 13     | 12     | 11     | 10     | 9     | 8     | 7     | 6     | 5     | 4     | 3     | 2     | 1     | 0     |
| Reset  | 0x0000       |        |        |        |        |        |       |       |       |       |       |       |       |       |       |       |
| Name   | CBWS15       | CBWS14 | CBWS13 | CBWS12 | CBWS11 | CBWS10 | CBWS9 | CBWS8 | CBWS7 | CBWS6 | CBWS5 | CBWS4 | CBWS3 | CBWS2 | CBWS1 | CBWS0 |
| Access | RW           | RW     | RW     | RW     | RW     | RW     | RW    | RW    | RW    | RW    | RW    | RW    | RW    | RW    | RW    | RW    |

| Bit  | Name              | Function             | Description                                                                                                                                                                                                                                                                                                                                                      | Access |
|------|-------------------|----------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 15:0 | CBWSn<br>(n=15-0) | Window signal enable | <p>When Blank-Window is enable (CMP_OCR.BWEN=1), select the window signal used to MASK the comparator output. The timer PWM signals corresponding to each bit are detailed in the table below:</p> <ul style="list-style-type: none"> <li>0b0: Do not select timer PWM output as window signal</li> <li>0b1: Select timer PWM output as window signal</li> </ul> | RW     |

Table 17-17 Blank Window PWM List

| CBWS Bit | CMP1        | CMP2        | CMP3        | CMP4        |
|----------|-------------|-------------|-------------|-------------|
| 15       | TMR4_1_OWL  | TMR4_2_OWL  | TMR4_3_OWL  | TMR4_3_OWL  |
| 14       | TMR4_1_OWH  | TMR4_2_OWH  | TMR4_3_OWH  | TMR4_3_OWH  |
| 13       | TMR4_1_OVL  | TMR4_2_OVL  | TMR4_3_OVL  | TMR4_3_OVL  |
| 12       | TMR4_1_OVH  | TMR4_2_OVH  | TMR4_3_OVH  | TMR4_3_OVH  |
| 11       | TMR4_1_OUL  | TMR4_2_OUL  | TMR4_3_OUL  | TMR4_3_OUL  |
| 10       | TMR4_1_OUH  | TMR4_2_OUH  | TMR4_3_OUH  | TMR4_3_OUH  |
| 9        | TMR6_4_PWMA | TMR6_8_PWMA | TMR6_4_PWMB | TMR6_8_PWMB |
| 8        | TMR6_3_PWMA | TMR6_7_PWMA | TMR6_3_PWMB | TMR6_7_PWMB |
| 7        | TMR6_2_PWMA | TMR6_6_PWMA | TMR6_2_PWMB | TMR6_6_PWMB |
| 6        | TMR6_1_PWMA | TMR6_5_PWMA | TMR6_1_PWMB | TMR6_5_PWMB |
| 5        | TMRA_2_PWM3 | TMRA_4_PWM3 | TMRA_3_PWM3 | TMRA_4_PWM3 |
| 4        | TMRA_2_PWM2 | TMRA_4_PWM2 | TMRA_3_PWM2 | TMRA_4_PWM2 |
| 3        | TMRA_2_PWM1 | TMRA_4_PWM1 | TMRA_3_PWM1 | TMRA_4_PWM1 |
| 2        | TMRA_1_PWM3 | TMRA_3_PWM3 | TMRA_1_PWM3 | TMRA_2_PWM3 |
| 1        | TMRA_1_PWM2 | TMRA_3_PWM2 | TMRA_1_PWM2 | TMRA_2_PWM2 |
| 0        | TMRA_1_PWM1 | TMRA_3_PWM1 | TMRA_1_PWM1 | TMRA_2_PWM1 |

## 17.4.8 Comparator Blank Window Output Polarity Register (CMP\_BWPR)

| Offset | Bit Position |        |        |        |        |        |       |       |       |       |       |       |       |       |       |       |
|--------|--------------|--------|--------|--------|--------|--------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| 0x06   | 15           | 14     | 13     | 12     | 11     | 10     | 9     | 8     | 7     | 6     | 5     | 4     | 3     | 2     | 1     | 0     |
| Reset  | 0x0000       |        |        |        |        |        |       |       |       |       |       |       |       |       |       |       |
| Name   | CBWP15       | CBWP14 | CBWP13 | CBWP12 | CBWP11 | CBWP10 | CBWP9 | CBWP8 | CBWP7 | CBWP6 | CBWP5 | CBWP4 | CBWP3 | CBWP2 | CBWP1 | CBWP0 |
| Access | RW           | RW     | RW     | RW     | RW     | RW     | RW    | RW    | RW    | RW    | RW    | RW    | RW    | RW    | RW    | RW    |

| Bit  | Name              | Function                        | Description                                                                                                                                                                                                                                    | Access |
|------|-------------------|---------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 15:0 | CBWPn<br>(n=15-0) | Timer Window Polarity Selection | Set the active level of the window signal selected by CMP_BWSR:<br><ul style="list-style-type: none"> <li>0b0: Comparator output closed when window signal is high</li> <li>0b1: Comparator output closed when window signal is low</li> </ul> | RW     |

## 17.5 Precautions

### 17.5.1 Module Stop Function

CMP has a module stop function, which can turn off the digital part of the module by setting the module stop register.

CMP1 and CMP2 share a module stop control bit (PWC\_FCG3.Bit8).

CMP3 and CMP4 share a module stop control bit (PWC\_FCG3.Bit9).

All comparator channels are initially in the stopped state, and their respective registers can only be accessed when the module is set to work. Please refer to the "Action Mode and Low Power Mode" section of the "Power Control (PWC) " chapter for register settings.

### 17.5.2 Power Consumption During Module Stop

When CMP transitions from the working state to the module stop state, the comparator will continue to operate, and the power consumption is equivalent to the working state. To further reduce power consumption, please clear the CMP\_MDR.CENA bit.

### 17.5.3 Power Consumption During Stop Low Power Mode

When the chip enters stop low power mode, if CMP is in the working state, it will continue to operate after entering stop low power mode, and the power consumption is equivalent to the level before entering stop low power mode. To further reduce power consumption, please clear the CMP\_MDR.CENA bit before entering stop low power mode.

## 18 Analog-to-Digital Converter (ADC)

### 18.1 Introduction

The 12-bit ADC is an analog-to-digital converter that adopts the successive approximation method. This MCU is equipped with 3 ADC units, units 1 and 2 support 16 channels, unit 3 supports 20 channels, which can convert analog signals from external pins and inside the chip. Analog input channels can be arbitrarily combined into a sequence for single scan or continuous scan conversion. It supports continuous multiple conversions on any specified channel and averaging of the conversion results. The ADC block also includes an analog watchdog function that monitors the conversion results of any specified channel to detect if they exceed user-set ranges.

### 18.2 Features

- High Performance
  - ▶ Configurable 12-bit, 10-bit and 8-bit resolution
  - ▶ The frequency ratio of the ADC digital interface clock (PCLK4) and conversion clock (PCLK2, also known as ADCLK) can be set to 1:1, 2:1, 4:1, 8:1, 1:2, or 1:4  
PCLK2 can optionally use a PLL clock source asynchronous to the system clock (HCLK), with a frequency ratio of PCLK4:PCLK2=1:1  
PCLK2 frequency supports up to 60MHz
  - ▶ Sample Rate: 2.5Msps (PCLK2=60MHz, 12-bit, 11 sample periods, 13 conversion periods)
  - ▶ Independent programming of channel sampling time is supported
  - ▶ Each channel has an independent data register
  - ▶ Data registers can be configured for left/right alignment
  - ▶ Continuous multiple conversion averaging function
  - ▶ Oversampling function
  - ▶ Analog watchdog to monitor conversion results
  - ▶ The ADC module can be set to stop when not in use
- Analog Input Channel
  - ▶ Maximum of 28 external analog inputs, with a maximum of 20 channels supported by a single ADC unit
  - ▶ 2 internal analog signals: internal reference voltage, and VBAT divider
- Conversion Start Conditions
  - ▶ Software setup conversion started
  - ▶ Peripheral synchronously trigger the conversion to start
  - ▶ External pin trigger the conversion to start
- Conversion Mode
  - ▶ 2 scan sequences A, B, optionally specifying a single or multiple channels
  - ▶ Sequence A single scan
  - ▶ Sequence A continuous scan
  - ▶ Sequence A data buffer
  - ▶ Dual sequence scan, with sequences A and B independently selecting trigger sources, and sequence B having higher priority than A.

- ▶ Cooperative operating mode (applicable to devices with two or three ADCs)
- Interrupt and Event Signal Output
  - ▶ Sequence A scan end interrupt and event ADC\_EOCA
  - ▶ Sequence B scan end interrupt and event ADC\_EOCB
  - ▶ Analog watchdog 0 compare interrupt and event ADC\_CMP0
  - ▶ Analog watchdog 1 compare interrupt and event ADC\_CMP1
  - ▶ All the event outputs mentioned above can start DMA

## 18.3 Functional Description

### 18.3.1 Function Block Diagram

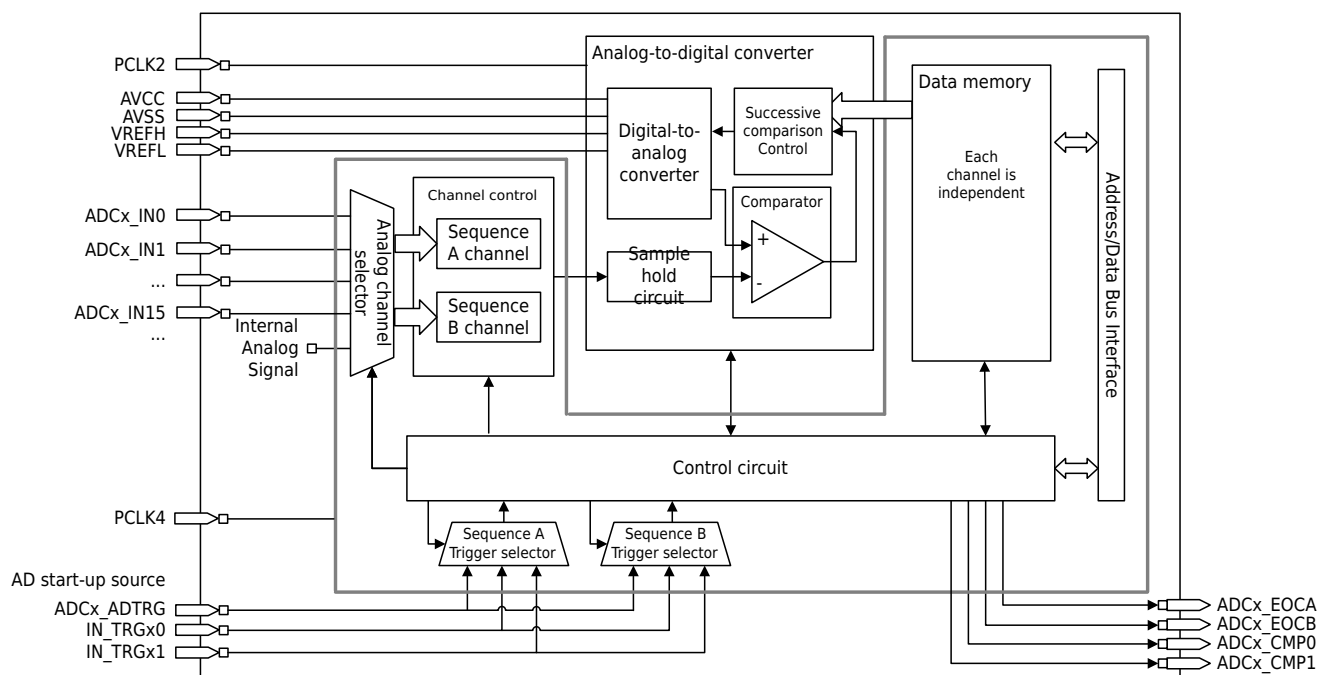


Figure 18-1 ADC Block Diagram

This chip is equipped with 3 ADC units, each with different configurations. Please refer to the table below for details:

Table 18-1 Specifications of Each ADC Unit

| Item              |     | Unit 1 (ADC1) | Unit 2 (ADC2) | Unit 2 (ADC3) |
|-------------------|-----|---------------|---------------|---------------|
| Power Supply      |     | AVCC          |               |               |
|                   |     | AVSS/VREFL    |               |               |
| Reference voltage |     | VREFH         |               |               |
| Analog Channel*1  | CH0 | ADC123_IN0    | ADC123_IN0    | ADC123_IN0    |
|                   | CH1 | ADC123_IN1    | ADC123_IN1    | ADC123_IN1    |
|                   | CH2 | ADC123_IN2    | ADC123_IN2    | ADC123_IN2    |
|                   | CH3 | ADC123_IN3    | ADC123_IN3    | ADC123_IN3    |
|                   | CH4 | ADC12_IN4     | ADC12_IN4     | ADC3_IN4      |
|                   | CH5 | ADC12_IN5     | ADC12_IN5     | ADC3_IN5      |

| Item                               |                     | Unit 1 (ADC1)                      | Unit 2 (ADC2)                      | Unit 2 (ADC3)                     |
|------------------------------------|---------------------|------------------------------------|------------------------------------|-----------------------------------|
| Analog Channel*1                   | CH6                 | ADC12_IN6                          | ADC12_IN6                          | ADC3_IN6                          |
|                                    | CH7                 | ADC12_IN7                          | ADC12_IN7                          | ADC3_IN7                          |
|                                    | CH8                 | ADC12_IN8                          | ADC12_IN8                          | ADC3_IN8                          |
|                                    | CH9                 | ADC12_IN9                          | ADC12_IN9                          | ADC3_IN9                          |
|                                    | CH10                | ADC123_IN10                        | ADC123_IN10                        | ADC123_IN10                       |
|                                    | CH11                | ADC123_IN11                        | ADC123_IN11                        | ADC123_IN11                       |
|                                    | CH12                | ADC123_IN12                        | ADC123_IN12                        | ADC123_IN12                       |
|                                    | CH13                | ADC123_IN13                        | ADC123_IN13                        | ADC123_IN13                       |
|                                    | CH14                | ADC12_IN14                         | ADC12_IN14                         | ADC3_IN14                         |
|                                    | CH15                | ADC12_IN15/ Internal analog signal | ADC12_IN15/ Internal analog signal | ADC3_IN15/ Internal analog signal |
|                                    | CH16                | -                                  | -                                  | ADC3_IN16                         |
|                                    | CH17                | -                                  | -                                  | ADC3_IN17                         |
|                                    | CH18                | -                                  | -                                  | ADC3_IN18                         |
|                                    | CH19                | -                                  | -                                  | ADC3_IN19                         |
| Program-mable Gain Amplifier PGA   | PGA1/4              | ADC123_IN0                         | ADC12_IN6                          | -                                 |
|                                    | PGA2                | ADC123_IN1                         | -                                  | -                                 |
|                                    | PGA3                | ADC123_IN2                         | -                                  | -                                 |
|                                    | COM                 | PGA123_VSS                         | PGA4_VSS                           | -                                 |
| Dedicated Sample Hold Circuit (SH) | SH1                 | ADC123_IN0                         | -                                  | -                                 |
|                                    | SH2                 | ADC123_IN1                         | -                                  | -                                 |
|                                    | SH3                 | ADC123_IN2                         | -                                  | -                                 |
| Hardware Trigger Source            | External pin        | ADC1_ADTRG                         | ADC2_ADTRG                         | ADC3_ADTRG                        |
|                                    | On-chip peripherals | IN_TRG10                           | IN_TRG20                           | IN_TRG30                          |
|                                    |                     | IN_TRG11                           | IN_TRG21                           | IN_TRG31                          |

**Note**

In the ADC, virtual channels CH0-CH15 can be freely mapped to physical channels ADCx\_INy (actual analog input sources) through register configuration. The table shows the default mapping after reset.

### 18.3.2 ADC Clock

The ADC module requires two clocks: the digital interface clock PCLK4 and the analog circuit clock PCLK2. PCLK4 and PCLK2 are synchronized, with frequency ratios settable to 1:1, 2:1, 4:1, 8:1, 1:2, or 1:4.

PCLK2 can optionally use a PLL clock source asynchronous to HCLK, with PCLK4 and PCLK2 being the same in a synchronous and same-frequency relationship in this case.

**Notice**

PCLK2 frequency should be set within 1MHz-60MHz.

### 18.3.3 Channel Selection

The ADC block supports channel mapping, allowing virtual channels to be mapped to physical channels. Virtual channels are hypothetical channels within the ADC block. For example, setting ADC\_CHSELRA to 0x1 indicates that sequence A selects CH0 for conversion. CH0 is a virtual channel, and ADC\_DR0 is the conversion result register for virtual channel CH0. Physical channels refer to actual analog channels, including external pin analog inputs ADCx\_INy and internal analog signals such as the internal reference voltage. The mapping between virtual and physical channels can be configured through the ADC\_CHMUXR register. Refer to the register description for specifics.

**Note**

Unless otherwise specified, channel n or CHn refers to a virtual channel.

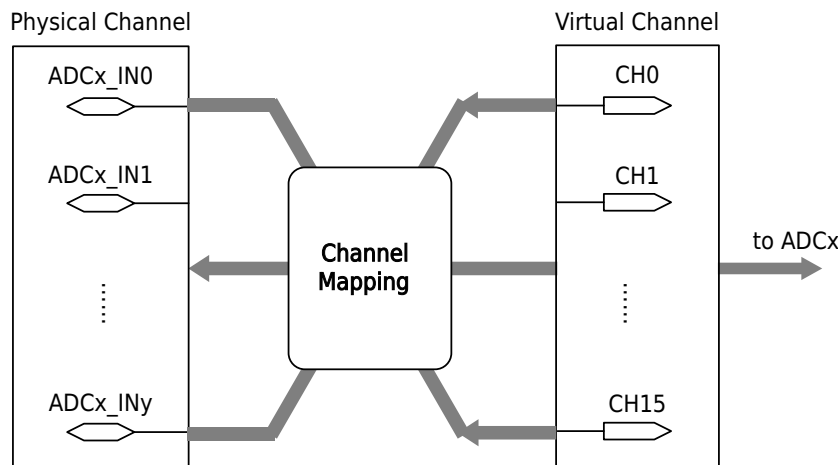


Figure 18-2 Channel Mapping Diagram

The ADC block has multiple channels configurable into two sequences: A and B. Sequences A and B have independent channel selection registers ADC\_CHSELRA and ADC\_CHSELRB. Each bit in these registers represents a channel. Writing 1 to Bit0 indicates conversion of CH0, while writing 0 indicates no conversion. Sequences A and B can independently select any one or more channels for conversion. For example, setting ADC\_CHSELRA to 0x0055 and ADC\_CHSELRB to 0x0002 means that when the trigger condition for sequence A occurs, CH0, CH2, CH4, and CH6 will be converted sequentially. When the trigger condition for sequence B occurs, CH1 will be converted.

Including, internal analog signals share CH15 with external input ADCx\_IN15. Setting ADC\_CHMUX simultaneously changes the mapping between internal analog signals/ADCx\_IN15 and CHn. To convert internal analog signals or ADCx\_IN15, first set the extended channel register ADC\_EXCHSELR, then write 1 to the corresponding virtual channel bit in ADC\_CHSELRA/B (default mapping is Bit15). When ADC\_EXCHSELR is set to 0x0, external analog input is selected. When set to 0x1, internal analog signals are selected. Selection of the 2 internal analog signals is detailed in the PWC\_PWRC4 register description in the "Power Control (PWC)" chapter.



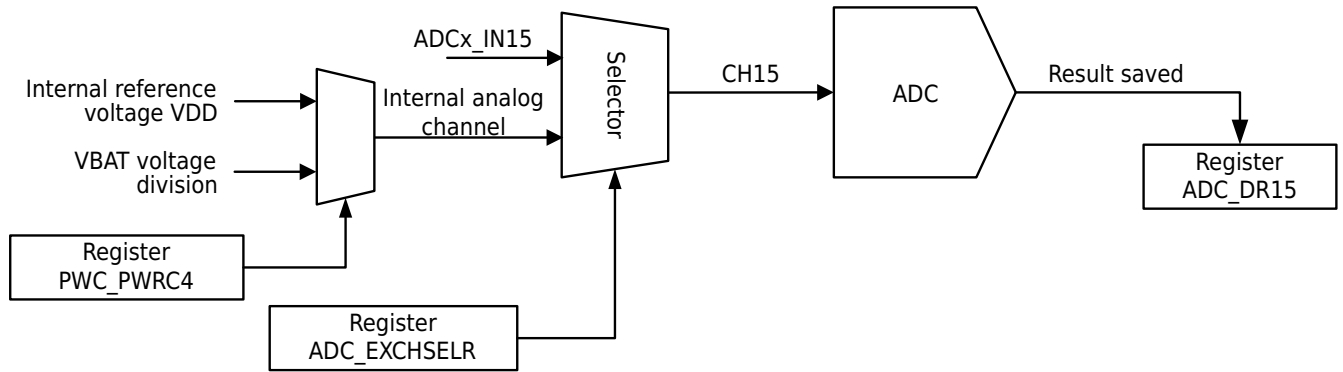


Figure 18-3 Internal Analog Signal Selection

Additionally, most physical channels can input to multiple ADC units. For example, ADC123\_IN0 can input to ADC1, ADC2, and ADC3, while ADC12\_IN4 can input to ADC1 and ADC2. Combining the cooperative operating mode of multiple ADCs and channel mapping allows high sample rate conversion of specified channels.

**Notice**

Do not select the same channel for sequences A, B. For non-existent channels, do not set the corresponding registers and leave them in their reset state.

### 18.3.4 Trigger Source Selection

Sequences A, B independently select trigger sources. The selectable trigger sources include external ports ADCx\_ADTRG and internal events IN\_TRGx0/1. Among them, port ADCx\_ADTRG triggers are active on falling edges. IN\_TRGx0/1 are set by registers ADC\_TRGSEL0/1, which can select rich event sources inside the chip. Additionally, writing to the ADC\_STR register can generate a software trigger signal for sequence A, but can only be used when the ADC is in standby mode. Software triggers are independent of the ADC\_TRGSR register.

### 18.3.5 Sequence A Single Scan Mode

Set ADC\_CR0.MS to 0b000 to select sequence A single scan mode.

In this mode, when the trigger condition for sequence A selected by ADC\_TRGSR register occurs or ADC\_STR.STRT is written to 1 for software trigger, and then the ADC starts, sequentially sampling and converting all channels selected in ADC\_CHSELRA. Conversion results are stored in the corresponding data registers ADC\_DR. ADC\_STR.STRT remains 1 during the ADC conversion process, and is automatically cleared to 0 when all channel conversions are completed, and the ADC enters the conversion standby state, waiting for the next trigger condition to occur.

When all channel conversion ends, the sequence A conversion end flag bit ADC\_ISR.EOCAP is set to 1 and the sequence A conversion end event ADC\_EOCA is generated, which can be used to start the DMA. If the ADC\_ICR.EOCAIEN is 1, the sequence A conversion end interrupt request is generated in the state of interrupt enable.

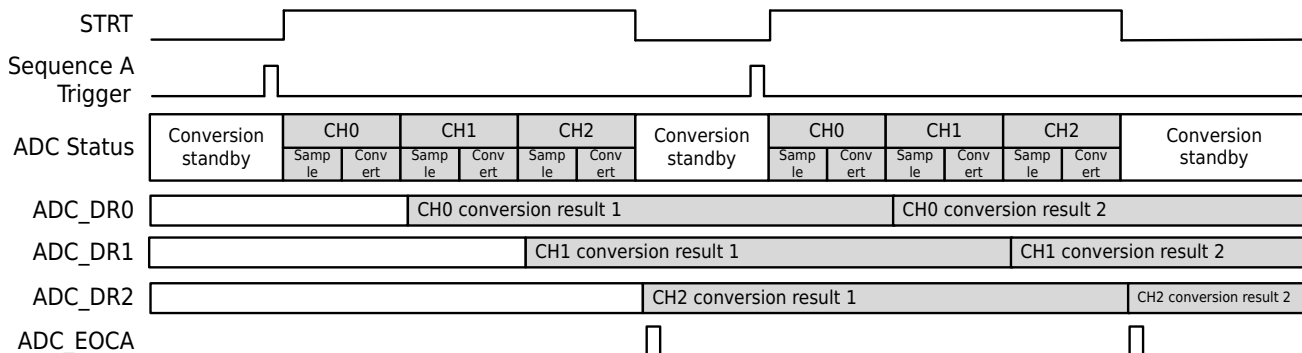


Figure 18-4 Sequence A Single Scan Mode

Software flow for sequence A single scan mode:

- Step1.** Confirm ADC\_STR.STRT is 0, placing the ADC in conversion standby mode;
- Step2.** Set ADC\_CR0.MS to 0b000 to select sequence A one-shot scan mode;
- Step3.** Set sequence A channel selection register ADC\_CHSELRA;
- Step4.** Set the sampling time register ADC\_SSTR;
- Step5.** Write 1 to ADC\_STR.STRT for software trigger of sequence A or set ADC\_TRGSR to select the trigger condition for sequence A;
- Step6.** Poll the sequence A conversion end flag bit EOCAF;
- Step7.** Read the channel data registers ADC\_DR;
- Step8.** Write 0 to clear the EOCAF flag bit, preparing for the next conversion.

The CPU query modes in steps 6-8 above can also be replaced by interrupt mode, use the ADC\_EOCA interrupt to handle the converted data. Or start DMA read data using the ADC\_EOCA event.

### 18.3.6 Sequence A Continuous Scan Mode

Set ADC\_CR0.MS to 0b001 to select sequence A continuous scan mode.

Sequence A continuous scan mode is similar to single scan mode, but after all channels are converted, conversion sequence A is restarted instead of entering standby mode. STRT bits also do not automatically clear 0.

Write 0 to the STRT bit and read STRT to confirm 0 to determine that the ADC enters the conversion standby state when a continuous scan needs to be stopped.

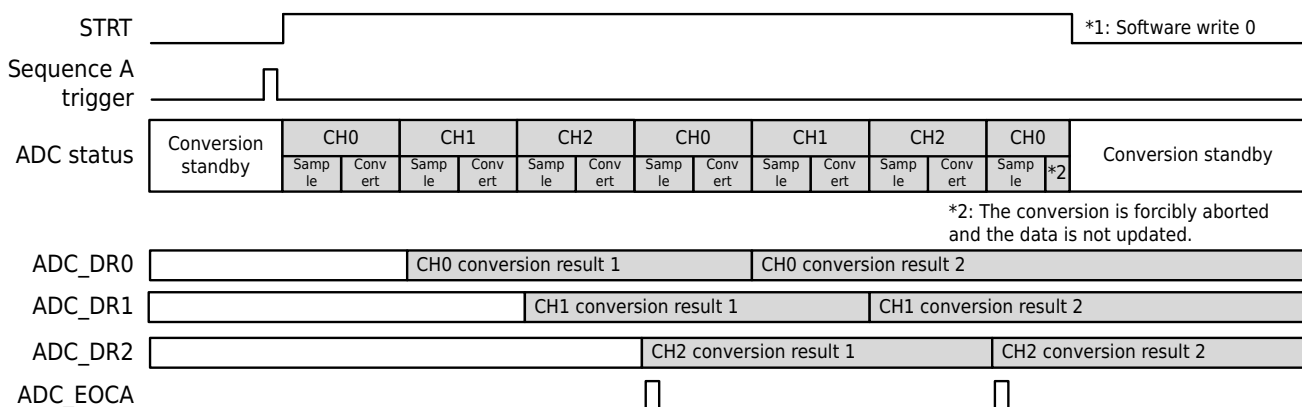


Figure 18-5 Continuous Scan

Software flow for sequence A continuous scan mode:

- Step1.** Confirm ADC\_STR.STRT is 0, placing the ADC in conversion standby mode;

- Step2.** Set ADC\_CR0.MS to 0b001 to select sequence A continuous scan mode;
- Step3.** Set sequence A channel selection register ADC\_CHSELRA;
- Step4.** Set the sampling time register ADC\_SSTR;
- Step5.** Write 1 to ADC\_STR.STRT for software trigger of sequence A or set ADC\_TRGSR to select the trigger condition for sequence A;
- Step6.** Poll the sequence A conversion end flag bit EOCAF;
- Step7.** Read the channel data registers ADC\_DR;
- Step8.** Write 0 to clear the EOCAF flag for preparation of the next conversion.
- Step9.** Write 0 to the STRT bit when you do not need to continue the conversion, and read STRT to confirm 0 to determine the ADC is in conversion standby.

The query modes in steps 6-8 can be replaced with interrupt handling, use the ADC\_EOCA interrupt to handle the converted data. Or start DMA read data using the ADC\_EOCA event.



#### Notice

Because it is a continuous conversion, the interval between each scan is relatively short, especially when only one channel is selected for conversion. It is recommended to use the ADC\_EOCA event to start DMA read data in order to avoid data loss due to intinelly processing in query mode.

### 18.3.7 Sequence A Data Buffer Mode

Set ADC\_CR0.MS to 0b100 to select data buffer mode for sequence A.

The sequence A data buffer mode is similar in action to the sequence A single scan mode. The difference lies in that after the conversion of one channel in the data buffer mode is completed, it does not directly start the conversion of the next channel, but enters the conversion standby state and does not start the conversion of the next channel until the startup condition occurs again.

When all channel conversion ends, the sequence A conversion end flag bit ADC\_ISR.EOCAF is set to 1 and the sequence A conversion end event ADC\_EOCA is generated, which can be used to start the DMA. If the ADC\_ICR.EOCAIEN is 1, the sequence A conversion end interrupt request is generated in the state of interrupt enable.

This mode allows multiple virtual channels selected in ADC\_CHSELRA to be mapped to the same physical channel using channel mapping. After all channels are converted, conversion data can be read sequentially from the corresponding ADC\_DR registers.

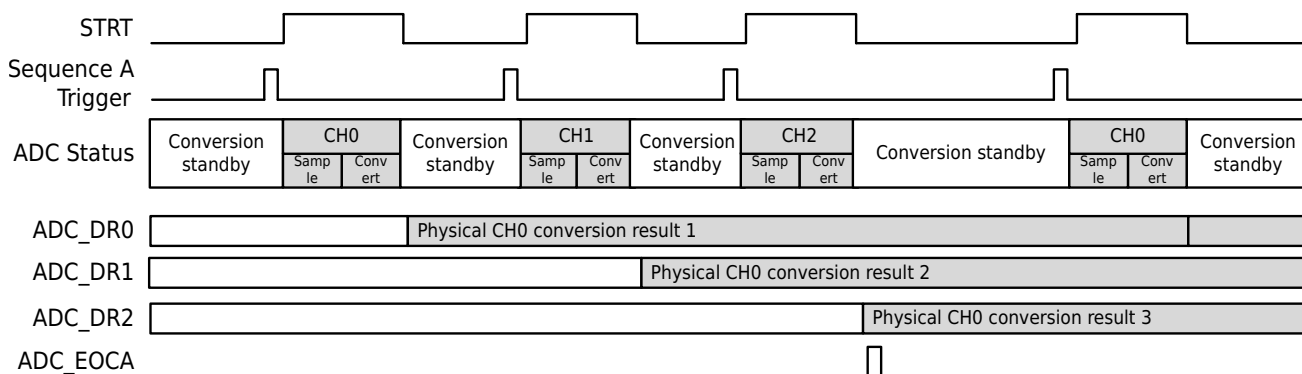


Figure 18-6 Data Buffer Mode

Software flow for sequence A data buffer mode:

- Step1.** Confirm ADC\_STR.STRT is 0, placing the ADC in standby mode;

- Step2.** Set ADC\_CR0.MS to 0b100 to select sequence A data buffer mode;
- Step3.** Set sequence A channel selection register ADC\_CHSELRA;
- Step4.** Set the channel mapping control register ADC\_CHMUXR;
- Step5.** Set the sampling time register ADC\_SSTR;
- Step6.** Write 1 to ADC\_STR.STRT for software trigger of sequence A or set ADC\_TRGSR to select the trigger condition for sequence A;
- Step7.** Poll for single conversion end ADC\_STR.STRT=0;
- Step8.** Repeat steps 6-7 until all channels are converted;
- Step9.** Poll the sequence A conversion end flag bit EOCAF;
- Step10.** Read the channel data registers ADC\_DR;
- Step11.** Write 0 to clear the EOCAF flag bit, preparing for the next conversion.
- Steps 9-11 can be replaced with interrupt handling, use the ADC\_EOCA interrupt to handle the converted data. Or start DMA read data using the ADC\_EOCA event.

### 18.3.8 Dual Sequence Scan Mode

Set ADC\_CR0.MS to 0b010 or 0b011 to select dual sequence scan mode, allowing both sequences A and B to be started by their respective trigger conditions.

When MS=0b010, sequences A and B are equivalent to two independent single scan sequences. MS=0b011, sequence A is continuous scan mode, sequence B is single scan mode. MS=0b110, sequence A is data buffer mode, B is single scan mode.

Sequence A is triggered by ADC\_TRGSR.TRGSELA and selects channels with ADC\_CHSELRA. Sequence B is triggered by ADC\_TRGSR.TRGSELB and selects channels with ADC\_CHSELRB.

When the conversion of all channels in sequence A is completed, the sequence A conversion end flag bit ADC\_ISR.EOCAF is set to 1 and the sequence A conversion end event ADC\_EOCA is generated. If the sequence A conversion end flag bit ADC\_ISR.EOCAIEN is set to 1, the sequence A conversion end interrupt request is generated in the state of interrupted permission. When the conversion of all channels of sequence B ends, the sequence B conversion end flag ADC\_ISR.EOCBF is set to 1, and the sequence B conversion end event ADC\_EOCB is generated. If ADC\_ISR.EOCBIEN is 1, the interrupt is enabled and the sequence B conversion end interrupt is generated at the same time.

In dual sequence scanning mode, sequence B will be prioritized when sequence A is competing with sequence B, that is, sequence B has higher priority than sequence A. Please refer to the table below for details.

Table 18-2 Various Conflicts Between Sequences A and B

| ADC Conversion               | Trigger Signal Occurrence | Mode of Treatment                                                                                                                                              |                                                                                                                                                         |
|------------------------------|---------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------|
|                              |                           | ADC_CR1.RSCHSEL=0                                                                                                                                              | ADC_CR1.RSCHSEL=1                                                                                                                                       |
| During Sequence A Conversion | Sequence A Triggering     | Inactive trigger signal                                                                                                                                        |                                                                                                                                                         |
|                              | Sequence B Triggering     | 1. Sequence A conversion interrupted, sequence B conversion started<br>2. After Sequence B conversion completes, sequence A continues from interrupted channel | 1. Sequence A conversion interrupted, sequence B conversion started<br>2. After Sequence B conversion completes, sequence A restarts from first channel |

| ADC Conversion                                            | Trigger Signal Occurrence | Mode of Treatment                                                                                       |                   |
|-----------------------------------------------------------|---------------------------|---------------------------------------------------------------------------------------------------------|-------------------|
|                                                           |                           | ADC_CR1.RSCHSEL=0                                                                                       | ADC_CR1.RSCHSEL=1 |
| During Sequence B Conversion                              | Sequence B Triggering     | Sequence B all channel conversion is complete and sequence A conversion is started                      |                   |
|                                                           | Sequence B Triggering     | Inactive trigger signal                                                                                 |                   |
| In AD idle, sequence A and B are triggered simultaneously |                           | Sequence B starts first, after all channel conversions are complete, sequence A conversions are started |                   |

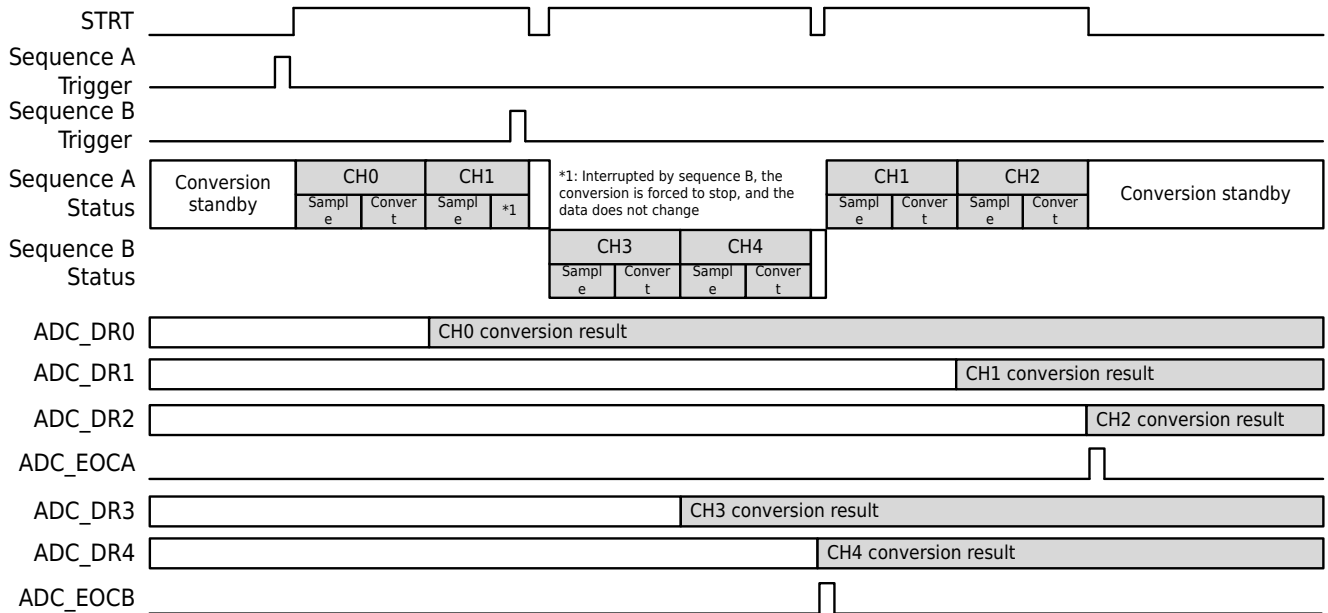


Figure 18-7 Dual Sequence Scan Mode (Sequence A Restarts from Interrupted Channel)

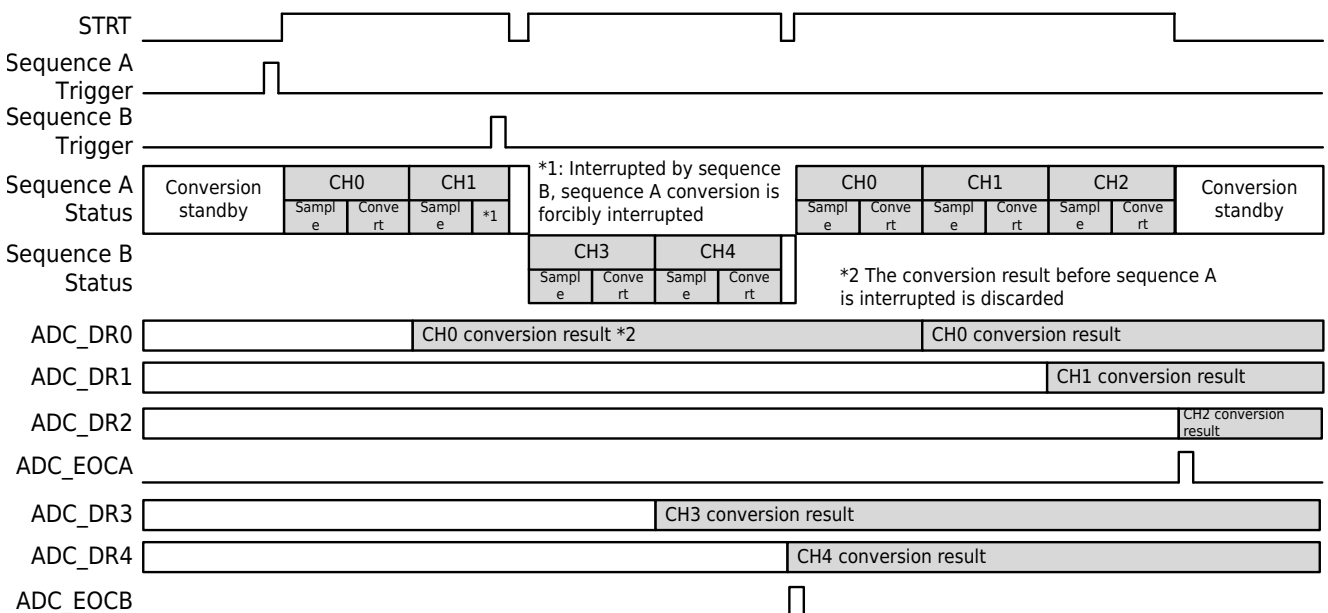


Figure 18-8 Dual Sequence Scan Mode (Sequence A Restarts from First Channel)

Software flow for dual sequence scan mode:

**Step1.** Confirm ADC\_STR.STRT is 0, placing the ADC in standby mode;

- Step2.** Set ADC\_CR0.MS to 0b010 or 0b011 to select dual sequence scan mode;
- Step3.** Set ADC\_CR1.RSCHSEL register to select the startup method after Sequence A is interrupted;
- Step4.** Set sequence A channel selection register ADC\_CHSELRA;
- Step5.** Set sequence B channel selection register ADC\_CHSELRB;
- Step6.** Set the sampling time register ADC\_SSTR;
- Step7.** Set ADC\_TRGSR register to select trigger conditions for Sequence A and B;
- Step8.** Handle conversion data after Sequence A or B conversion ends by polling EOCAF, EOCBF, using ADC\_EOCA, ADC\_EOCB interrupts, or initiating DMA.

**Notice**

Do not select the same channel for sequences A and B. Do not select the same trigger source for sequences A and B.

### 18.3.9 Analog Watchdog Function

The analog watchdog function refers to comparing the conversion result at the end of an ADC conversion for a channel. This ADC supports two comparison windows: window 0 and window 1. Taking window 0 as an example, as shown in the figure below, if the conversion result is within the protected area, a watchdog comparison interrupt and event ADC\_CMP0 are generated.

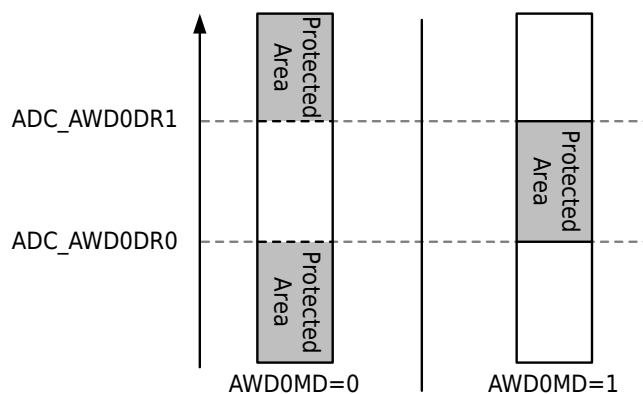


Figure 18-9 Analog Watchdog Protected Area (Comparative Mode)

Software flow for using the analog watchdog function (Taking window 0 as an example) :

- Step1.** Set the threshold registers AD\_AWD0DR0 and ADC\_AWD0DR1;
- Step2.** Set the comparison channel register (ADC\_AWD0CHSR) to select the channels to be compared;
- Step3.** Set ADC\_AWDCR.AWD0MD to select the comparative mode;
- Step4.** Set ADC\_AWDCR.AWD0IEN interrupt permission bit;
- Step5.** Set ADC\_AWDCR.AWD0EN to enable the watchdog window 0 comparison function;
- Step6.** Set the scan mode according to the previous steps and start the ADC conversion;
- Step7.** In the ADC\_CMP0 interrupt or after ADC conversion ends, query the comparison status register (ADC\_AWDSR.AWD0F) and process the comparison result accordingly.

The usage of comparison window 1 is the same as window 0.

Two comparison windows can be used in combination. When the window combination function is enabled, at the end of the conversion for the channel selected in window 1, the comparison interrupt ADC\_CMP1 outputs not just the comparison result from window 1 alone, but a combined result of logical OR, AND, or XOR of the comparison results from windows 0 and 1, as set.

The software flow for using the window combination comparison function is similar to using the windows individually. After setting up windows 0 and 1, additionally set the ADC\_AWDCR.AWDCM register to select the combination method.

### 18.3.10 Sampling Time and Conversion Time for Analog Inputs

In single scan mode, ADC conversion can be started by software settings, internal triggers IN\_TRGx0/1, and external pin triggers ADCx\_ADTRG startup mode. After the scan conversion delay time  $t_D$ , the ADC module starts to sample and convert the analog channel. After all the conversions are completed, the conversion end delay time  $t_{ED}$  enters the standby state, and a scan is finally completed. The continuous scanning mode is similar to a single scanning mode, except that there is no  $t_D$  time at the second time of the sequence and at the start of the sequence.

The conversion time of a single channel is  $t_{CONV} = t_{SPL} + t_{CMP}$ . Among them,  $t_{SPL}$  represents the sampling time of the analog input, and the number of sampling periods can be adjusted according to the input impedance setting register ADC\_SSTR.  $t_{CMP}$  represents the successive comparison time, with 13 PCLK2 for 12-bit precision, 11 PCLK2 for 10-bit precision, and 9 PCLK2 for 8-bit precision.

The time of one scan conversion  $t_{SCAN} = t_D + \sum t_{CONV} + t_{ED}$ . Among them,  $\sum t_{CONV}$  represents the sum of the conversion time of all scanning channels. Since the sampling time  $t_{SPL}$  can be set independently, the conversion time  $t_{CONV}$  of each channel can be different.

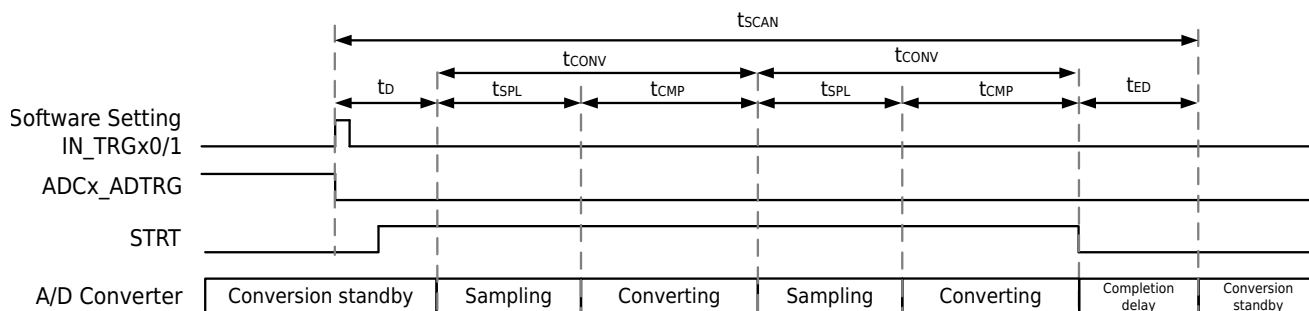


Figure 18-10 ADC Conversion Time Diagram

Table 18-3 ADC Conversion Time

| Name              |                   | Description                         |                                                                | Conditions                     |                                                |                      |                  |
|-------------------|-------------------|-------------------------------------|----------------------------------------------------------------|--------------------------------|------------------------------------------------|----------------------|------------------|
|                   |                   |                                     |                                                                | Synchronous Peripheral Trigger | Asynchronous Peripheral Trigger <sup>(1)</sup> | External Pin Trigger | Software Trigger |
| t <sub>D</sub>    |                   | Scan start processing time          | ADC is idle, start conversion                                  | 4 PCLK2                        | 3 PCLK4 + 4 PCLK2 + 1 PCLK4_SYNC               | 2 PCLK4 + 4 PCLK2    | 3 PCLK2          |
|                   |                   |                                     | Sequence A conversion interrupted, start sequence B conversion | 5 PCLK2                        | 4 PCLK4 + 5 PCLK2 + 1 PCLK4_SYNC               | 2 PCLK4 + 5 PCLK2    | -                |
| t <sub>CONV</sub> | t <sub>SPL</sub>  | Sample time                         |                                                                | ADC_SSTRx.SST x PCLK2          |                                                |                      |                  |
|                   |                   | Successive conversion time          | 12-bit resolution                                              | 13 PCLK2                       |                                                |                      |                  |
|                   | 10-bit resolution |                                     | 11 PCLK2                                                       |                                |                                                |                      |                  |
|                   | 8-bit resolution  |                                     | 9 PCLK2                                                        |                                |                                                |                      |                  |
| t <sub>ED</sub>   |                   | Scan completion processing time     |                                                                | 3 PCLK2                        |                                                |                      |                  |
| t <sub>TD</sub>   |                   | Minimum continuous trigger interval |                                                                | Σt <sub>CONV</sub> + 6 PCLK2   |                                                |                      |                  |



**Note**

1. Asynchronous peripheral trigger refers to the situation where the ADC block selects a PLL clock that is asynchronous to the system clock for operation. At this time, the peripheral module clock and the ADC module clock are in an asynchronous relationship. PCLK4\_SYNC represents the original synchronization clock of the ADC block (i.e., the clock set by CMU\_SCFGR), where PCLK4 and PCLK2 are the same and are both asynchronous PLL clocks.

### 18.3.11 ADC Data Register Automatic Clear Function

When ADC\_CR0.CLREN is 1, the ADC\_DR register will be automatically cleared to 0x0000 after being read by the CPU or DMA.

Using this function allows detection of whether the ADC\_DR register has been updated. Examples are provided below.

- When ADC\_CR0.CLREN is 0, automatic clear function disabled, if the analog quantity to be measured (0x0222) is not converted for some reason or the result is not updated to the ADC\_DR register, the ADC\_DR register will continue to hold the previous conversion value (0x0111). During ADC conversion completion interrupt handling, the not updated value (0x0111) will be read. To detect whether the ADC conversion value is enabled, it is necessary to additionally store the previous conversion value in RAM and judge by comparing the conversion results.
- When ADC\_CR0.CLREN is 1, automatic clear function enabled, after the previous conversion result (0x0111) is read by the CPU or DMA, the ADC\_DR register will be automatically cleared to 0x0000. After subsequent ADC conversions, if the conversion result is not correctly transferred to the ADC\_DR register, the ADC\_DR register will remain at 0x0000. In this case, if 0x0000 is read during interrupt handling, it is easy to determine whether the ADC conversion data has been stored correctly.

### 18.3.12 Conversion Data Averaging Function

The ADC conversion averaging function refers to performing 2, 4, 8, 16, 32, 64, 128, or 256 consecutive conversions on the same channel and averaging the conversion results before storing them in the data register. Using the averaging function can remove certain noise components and make the conversion results more precise.

The register ADC\_CR0.AVCNT sets the number of consecutive conversions, and the register ADC\_AVCHSELR selects any one or more channels to be averaged.

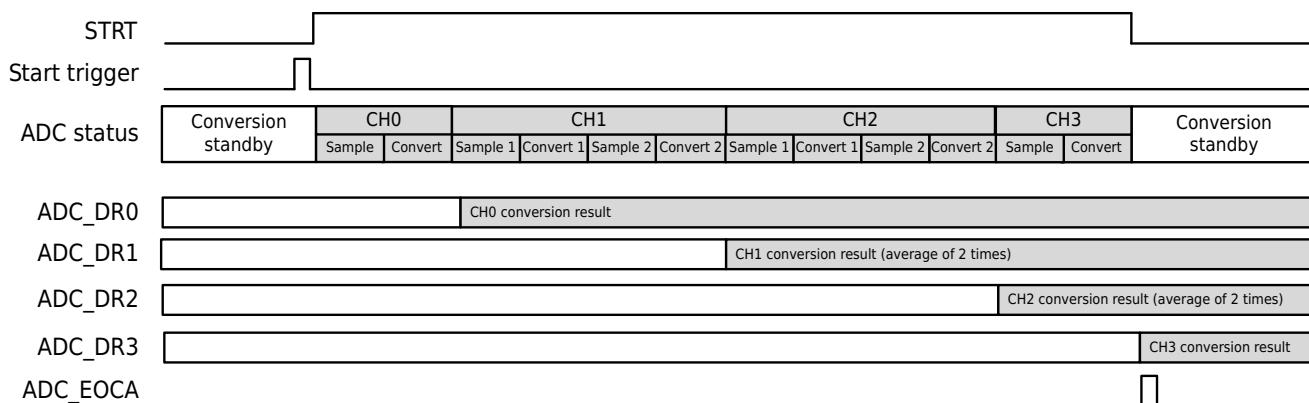


Figure 18-11 Conversion Action When Averaging Function is Enabled

In the diagram above, in sequence A single scan mode, 4 channels CH0-3 are selected for conversion, with CH1 and CH2 set to twice averaging mode. During the scanning process, CH1 and 2 will perform



two consecutive conversions, and save the averaged results to the data registers ADC\_DR1 and 2 of the corresponding channel.

### 18.3.13 Oversampling

By setting ADC\_CR2.OVSMOD to 1 and selecting any one or more channels for oversampling through setting ADC\_AVCHSELR, the AD enters oversampling mode. In this mode, multiple consecutive sampling conversions and result shifting are performed to obtain a 16-bit precision conversion result. The final conversion result is calculated using the following formula:

$$\text{AD oversampling result} = \frac{1}{2^M} \times \sum_{n=1}^{n=N} \text{AD conversion result of the } n\text{th time}$$

The sample rate N is set through ADC\_CR0.AVCNT and can be set to 2 to 256 samples.

The divisor series M is set through ADC\_CR2.OVSS and can be set to shift right by up to 8 bits.

### 18.3.14 Programmable Gain Amplifier PGA

This product is equipped with a Programmable Gain Amplifier PGA. The register ADC\_PGACR can be set to enable the PGA circuit and select the gain multiplier, with a gain range of x2-x32. In this case, the analog input is first amplified by the PGA circuit before being input to the ADC module for conversion.

Before using the PGA, please set bit3 of the register PWC\_FCG3 to enable the PGA and wait for a startup stabilization time of 2μs.

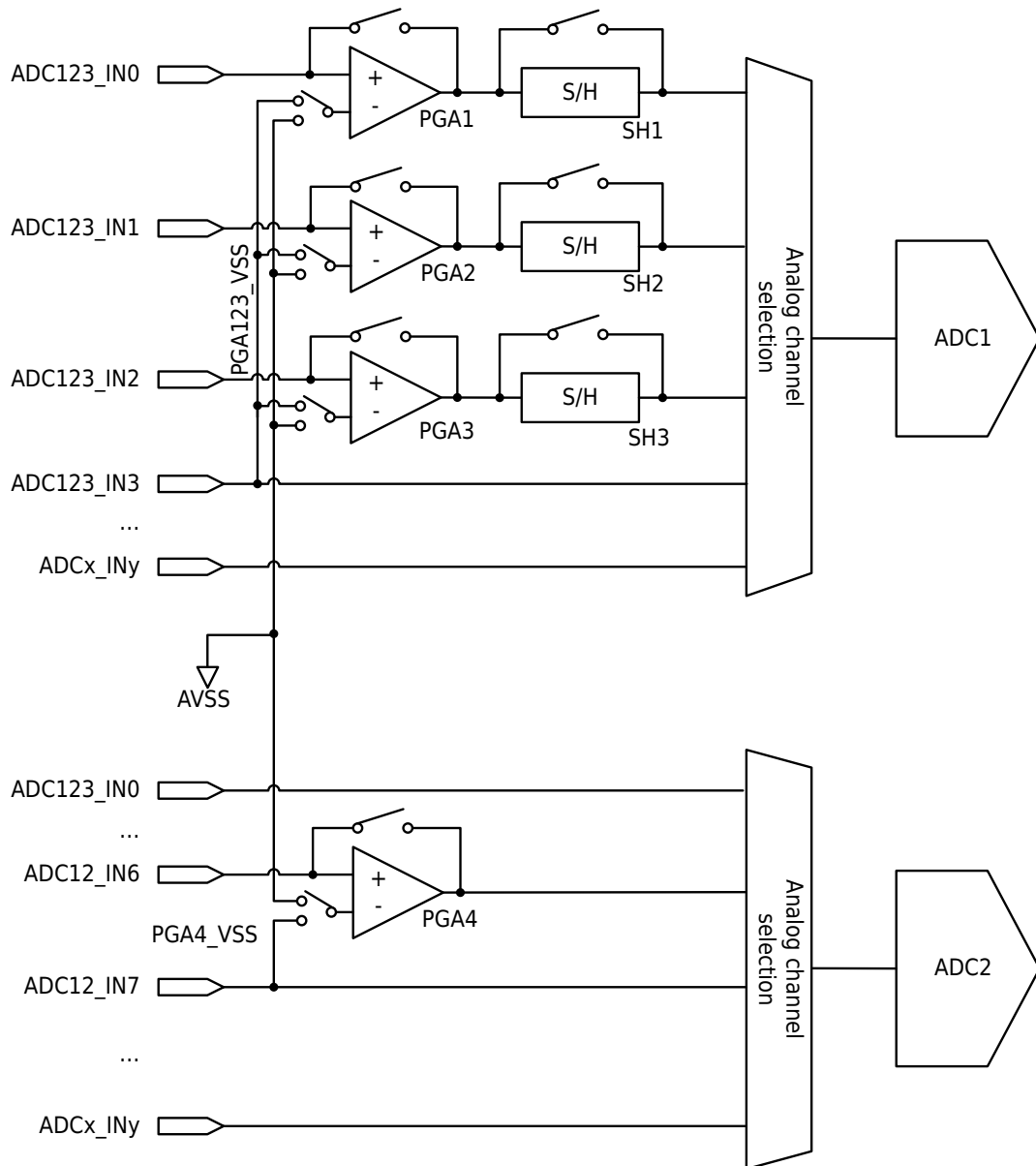


Figure 18-12 PGA and SH Channel Diagram for ADC1 and ADC2

As shown in the diagram above, ADC1 contains 3 PGA channels PGA1-3, which correspond to physical channels ADC123\_IN0-2, respectively. ADC2 contains 1 PGA channel PGA4, which corresponds to physical channel ADC12\_IN6.

The reference ground for the PGA can be selected as either internal AVSS or external PGAx\_VSS. When selecting external PGAx\_VSS, please connect it to the same voltage as AVSS.

### 18.3.15 Dedicated Sample Hold Circuit (SH) for Channels

This product is equipped with 3 dedicated sample hold circuits (SH) for use by channels CH0-2 of ADC1. When the dedicated sample hold circuit is enabled, at the start of each sequence, all channels with SH enabled are sampled simultaneously, then the ADC is triggered to perform conversion on each channel in the sequence one by one. In continuous scanning mode, the sampling time of SH will be inserted into the sequence at the start of the second and subsequent scanning.

The ADC\_SHCR.SHSEL bit can be set to enable the SH circuit, and the ADC\_SHCR.SHSST bit can be set to modify the sample period count of the SH circuit.

Before using SH, please set Bit3 of the PWC\_FCG3 register to enable SH and wait for a startup stabilization time of 2μs.

As shown in the diagram below, ADC1 contains 3 SH circuits, which can be used in combination with PGA. That is, it is first amplified by PGA, then sampled and held by SH, and finally input to the ADC for conversion. Alternatively, the PGA can be disabled, and the external pin input can be directly sampled by SH and then input to the ADC for conversion.

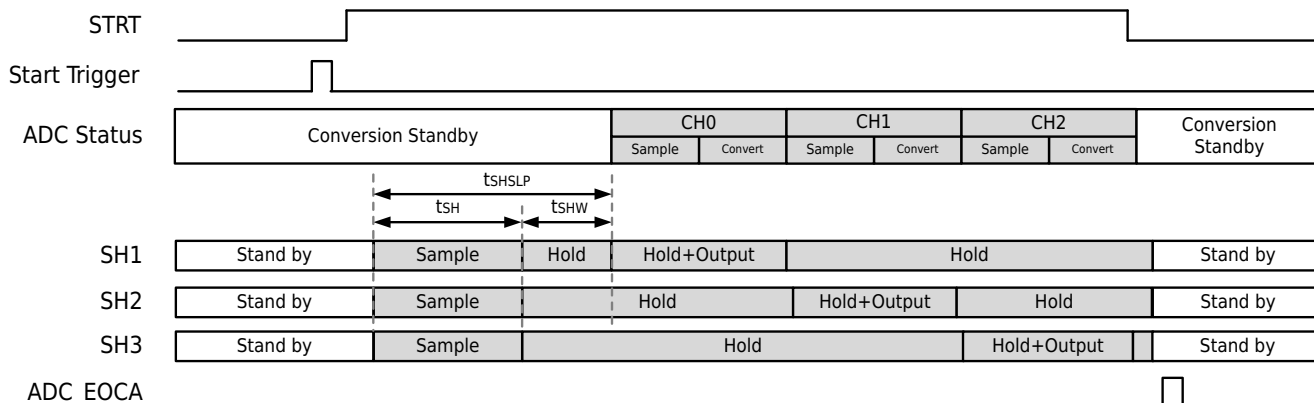


Figure 18-13 Dedicated Sample Hold Circuit for Channels Enabled

As shown in the diagram above, the dedicated sample hold circuits SH1-3 for CH0-2 are all enabled. After the sequence start trigger, the SH1-3 circuits pre-sample the analog inputs of channels CH0-2 simultaneously, which is represented by the time  $t_{SHSLP}$  in the diagram. Then, the ADC converts the analog voltages held by SH1-3. Where,  $t_{SHSLP} = t_{SH} + t_{SHW}$ .  $t_{SH}$  represents the sampling time of the dedicated sample hold circuit, which is set by ADC\_SHCR.SHSST and should be set to above  $0.4\mu s$ .  $t_{SHW}$  represents the waiting time after the sampling hold circuit completes sampling, which is 12 PCLK2 periods.



#### Notice

- In dual sequence scan mode, do not use SH in sequence A. This is because sequence A may be interrupted by sequence B, resulting in inconsistent sampling times for each dedicated sample hold circuit and unexpected conversion results.
- After writing 1 to ADC\_SHCR.SHSEL, the SH circuit is enabled, and the corresponding virtual channels CH0-2 are fixedly mapped to physical channels ADC123\_IN0-2. That is, when the SH circuit is enabled, the corresponding virtual channels do not support channel mapping.

### 18.3.16 Multi-ADC Cooperative Working Mode

On chips equipped with 2 or 3 ADC units, the ADC cooperative working mode can be used.

In the ADC cooperative working mode, ADC1 acts as the master unit, and the conversion of ADC2 and ADC3 is synchronized through the trigger signal of ADC1. That is, the setting of ADC\_TRGSR.TRGSELA for sequence A of ADC2 and ADC3 is invalid. All ADC units are triggered by the trigger source selected by the trigger source selection register for sequence A of ADC1. In this mode, writing 1 to ADC\_STR.STRT does not start the conversion, i.e., software start is disabled.

When using the cooperative working mode, please prohibit the sequence B action, so as not to disturb the synchronization. Two ADC units, ADC1 and ADC2, can be set to work cooperatively, or ADC1, ADC2, and ADC3 three ADC units can be set to work cooperatively.

The ADC can be configured into the following four cooperative working modes:

- [Single Parallel Trigger Mode](#)
- [Single Delay Trigger Mode](#)
- [Loop Parallel Trigger Mode](#)

## ● Loop Delay Trigger Mode

### Single Parallel Trigger Mode

The trigger condition of sequence A of ADC1 simultaneously triggers all ADC units in cooperative working mode, and only triggers once.

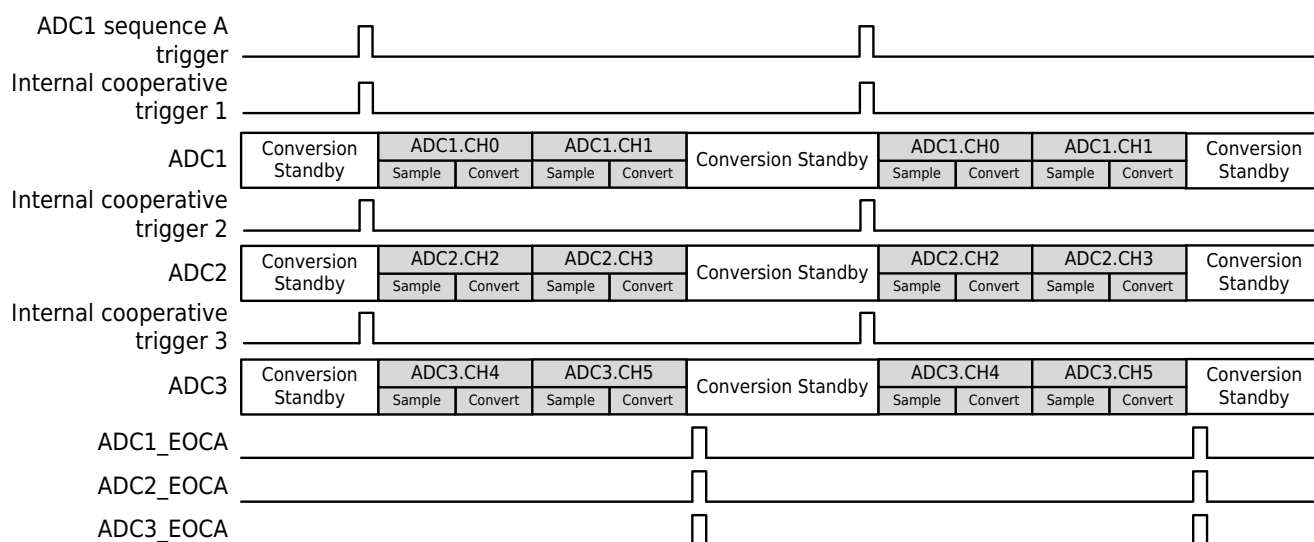


Figure 18-14 Single Parallel Trigger Mode (Three ADCs)



#### Notice

Disable multiple ADCs from simultaneously converting the same analog input. One analog channel can only be sampled by one ADC unit at a time; otherwise, the accuracy is not guaranteed. The same applies below.

Software setup flow for this mode:

- Step1.** Write 0 to the collaborative work permission register `ADC_SYNCRR.SYNCEN` to confirm that the collaborative work is disabled.
- Step2.** Set ADC1 unit:
- Confirm that `ADC1_STR.STRT` is 0, and ADC1 is in conversion standby mode.
  - Set the control register `ADC1_CR0.MS` to 0b000 (Sequence A single scan mode) or 0b001 (Sequence A continuous scan mode).
  - Set sequence A channel selection register `ADC1_CHSELRA`.
  - Set the sampling time register `ADC1_SSTR`.
  - Set the sequence A trigger source selection register `ADC1_TRGSR`.
- Step3.** Set ADC2 unit:
- Confirm that `ADC2_STR.STRT` is 0, and ADC2 is in conversion standby mode.
  - Set the control register `ADC2_CR0.MS`, channel selection register `ADC2_CHSELRA`, and channel sampling time register `ADC2_SSTR`.



#### Note

To ensure synchronous operation of ADC2 and ADC1, the above registers should be set to the same values as those of ADC1 as much as possible. The specific channels do not need to be the same, as long as the number of channels and the sampling time of the corresponding channels are consistent.

- Step4.** Set the ADC3 unit (For three ADCs working cooperatively).
- Confirm that `ADC3_STR.STRT` is 0, and ADC3 is in conversion standby mode.

- b. Set the control register ADC3\_CR0.MS, channel selection register ADC3\_CHSELRA, and channel sampling time register ADC3\_SSTR.

**Note**

Same as ADC2, to ensure synchronous operation between ADC3 and ADC1, the above registers should preferably be configured with the same values as ADC1's registers.

- Step5.** Set the cooperative mode control register ADC\_SYNCCR.SYNCMD to 0b010 (Cooperative working of ADC1 and ADC2) or 0b011 (Cooperative working of ADC1, ADC2, and ADC3).
- Step6.** Write 1 to the cooperative work enable register ADC\_SYNCCR.SYNCEN to enable cooperative working.
- Step7.** Wait for the input of the trigger source for sequence A of ADC1, and process the results after ADC1, ADC2, and ADC3 complete the conversion.

### Single Delay Trigger Mode

The trigger condition of sequence A of ADC1 triggers ADC1, and after a set delay, triggers ADC2 to start conversion, and then after another set delay, triggers ADC3 to start conversion. Each ADC unit is triggered only once.

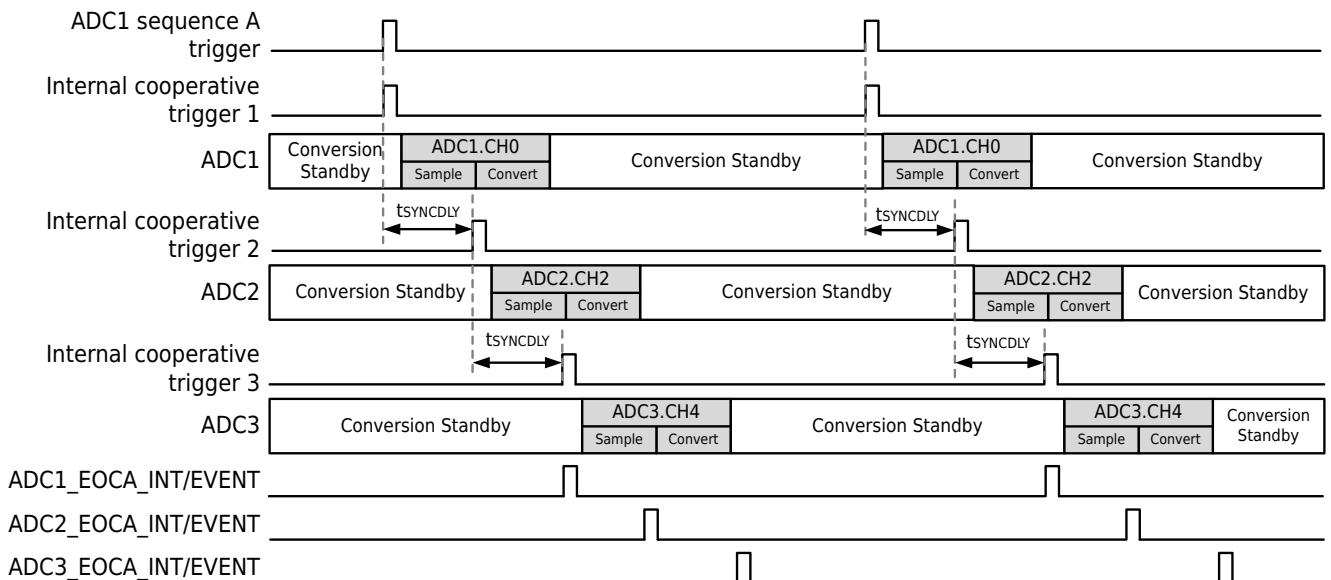


Figure 18-15 Single Delay Trigger Mode (Three ADCs)

**Notice**

- After the first input of the sequence A trigger of ADC1, before the cooperative trigger of ADC3 occurs, any subsequent input of the sequence A trigger of ADC1 will be ignored.
- If each ADC unit converts the same analog channel, the sampling time needs to be staggered, that is, the delay time  $t_{SYNCDLY}$  and the channel opening sampling time  $t_{SPL}$  need to satisfy:  $t_{SYNCDLY} > t_{SPL}$ .

Software setup flow for this mode:

- Step1.** Write 0 to the cooperative work enable register ADC\_SYNCCR.SYNCEN to disable cooperative working.
- Step2.** Set ADC1, ADC2, and ADC3 units (refer to single parallel mode).
- Step3.** Set the cooperative mode control register ADC\_SYNCCR.SYNCDLY to set the start delay for the two ADCs.

- Step4.** Set the cooperative mode control register ADC\_SYNCRR.SYNCMD to 0b000 (Cooperative working of ADC1 and ADC2) or 0b001 (Cooperative working of ADC1, ADC2, and ADC3).
- Step5.** Write 1 to the cooperative work enable register ADC\_SYNCRR.SYNCEN to enable cooperative working.
- Step6.** Wait for the input of the trigger source for sequence A of ADC1, and process the results after ADC1, ADC2, and ADC3 complete the conversion.

### Loop Parallel Trigger Mode

The trigger condition of sequence A of ADC1 simultaneously triggers all ADC units in cooperative working mode, and then after a specified delay, triggers all ADC units again simultaneously. This continues until the ADC1 unit is manually stopped by software or the cooperative working mode is disabled.

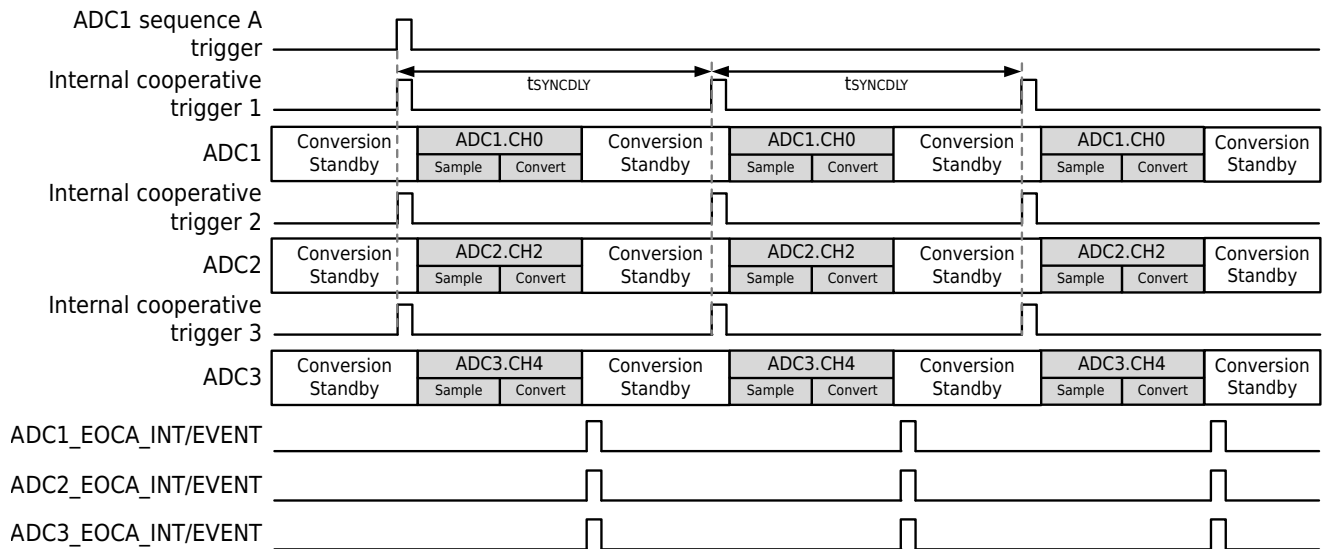


Figure 18-16 Loop Parallel Trigger Mode (Three ADCs)



#### Notice

The delay time  $t_{SYNCDLY}$  and the time  $t_{SCAN}$  of a scan conversion need to satisfy:  $t_{SYNCDLY} > t_{SCAN}$ .

The software setting process of this mode is as follows:

- Step1.** Write 0 to the cooperative work enable register ADC\_SYNCRR.SYNCEN to disable cooperative working.
- Step2.** Set ADC1, ADC2, and ADC3 units (refer to single parallel mode). Set ADC\_CR0.MS to 0b000 (refer to sequence A single scan mode).
- Step3.** Set the cooperative mode control register ADC\_SYNCRR.SYNCDLY to set the delay for each parallel trigger.
- Step4.** Set the cooperative mode control register ADC\_SYNCRR.SYNCMD, write 0b110 (Cooperative working of ADC1 and ADC2) or write 0b111 (Cooperative working of ADC1, ADC2, and ADC3).
- Step5.** Write 1 to the cooperative operation enable register ADC\_SYNCRR.SYNCEN to enable cooperative working.
- Step6.** Wait for the input of the trigger source for sequence A of ADC1, and process the results after ADC1, ADC2, and ADC3 complete the conversion.

## Loop Delay Trigger Mode

After the trigger condition of sequence A of ADC1 triggers ADC1, after a set delay, it sequentially and cyclically triggers ADC2, ADC3, ADC1, ADC2, etc., until the ADC1 unit is manually stopped by software or the cooperative working mode is disabled.

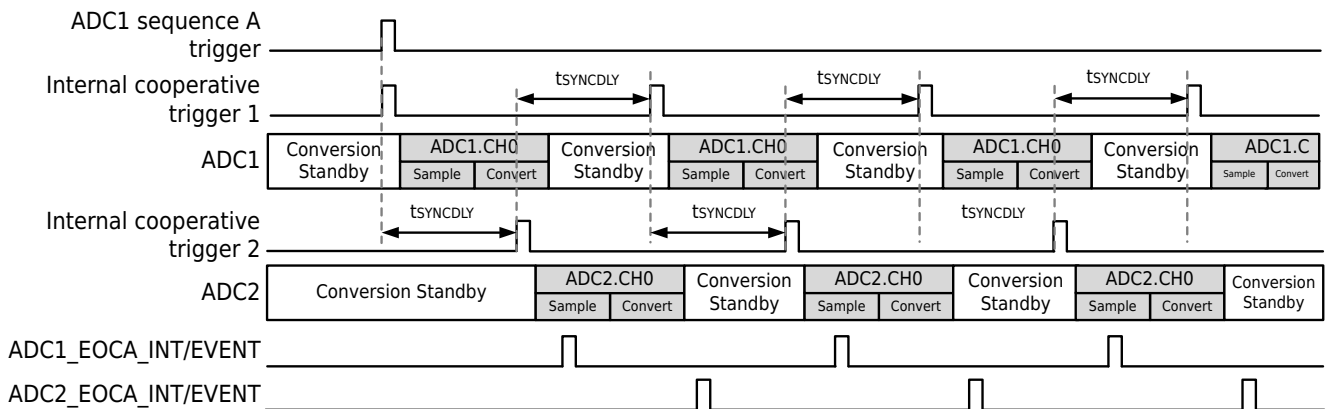


Figure 18-17 Loop Delay Trigger Mode (Two ADCs)

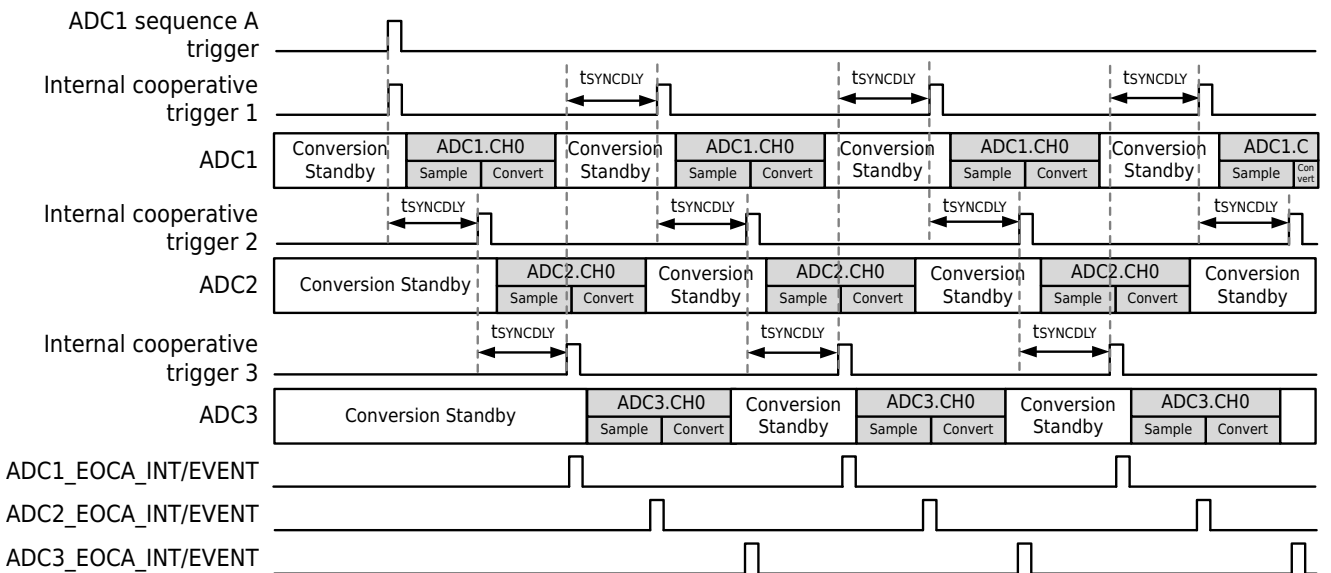


Figure 18-18 Loop Delay Trigger Mode (Three ADCs)

### Notice

When the two ADCs cooperative working, The delay time  $t_{SYNCDLY}$  and the time  $T_{SCAN}$  of a scan conversion need to satisfy:  $t_{SYNCDLY} > t_{SCAN}/2$ . When the three ADCs cooperative working, it needs to satisfy:  $t_{SYNCDLY} > t_{SCAN}/3$ . At the same time, if ADC1, ADC2, and ADC3 convert the same analog channel, the sampling time needs to be staggered, that is,  $t_{SYNCDLY} > t_{SPL}$ .

Software setup flow for this mode:

- Step1.** Write 0 to the cooperative work enable register `ADC_SYNCRR.SYNCCEN` to disable cooperative working.
- Step2.** Set ADC1, ADC2, and ADC3 units (refer to loop parallel trigger mode).
- Step3.** Set the cooperative mode control register `ADC_SYNCRR.SYNCDLY` to set the delay for each trigger.

- Step4.** Set the cooperative mode control register ADC\_SYNCCR.SYNCMD to 0b100 (Cooperative working of ADC1 and ADC2) or 0b101 (Cooperative working of ADC1, ADC2, and ADC3).
- Step5.** Write 1 to the cooperative operation enable register ADC\_SYNCCR.SYNCEN to enable cooperative working.
- Step6.** Wait for the input of the trigger source for sequence A of ADC1, and process the results after ADC1, ADC2, and ADC3 complete the conversion.

### 18.3.17 Interrupt and Events

The ADC block can generate the following kinds of event outputs. When each event occurs, if the corresponding interrupt enable register is set to active, an interrupt request is simultaneously output.

- For sequence A scan end ADC\_EOCA, corresponding to interrupt enable register ADC\_ICR.EOCAIEN
- For sequence B scan end ADC\_EOCB, corresponding to interrupt enable register ADC\_ICR.EOCBIEN
- For analog watchdog window 0 compare ADC\_CMP0, corresponding interrupt enable register ADC\_AWDCR.AWD0IEN
- For analog watchdog window 1 compare ADC\_CMP1, corresponding interrupt enable register ADC\_AWDCR.AWD1IEN

The above all event outputs can initiate other on-chip peripheral modules, including starting DMA transfer. Using DMA transfer allows continuous reading of ADC conversion results without software intervention, fully implemented by hardware, reducing CPU load. DMA settings are described in the relevant sections of the "DMA Controller (DMA) " chapter. Event signal output is independent of the interrupt enable bit control, and will be output as long as the condition occurs.

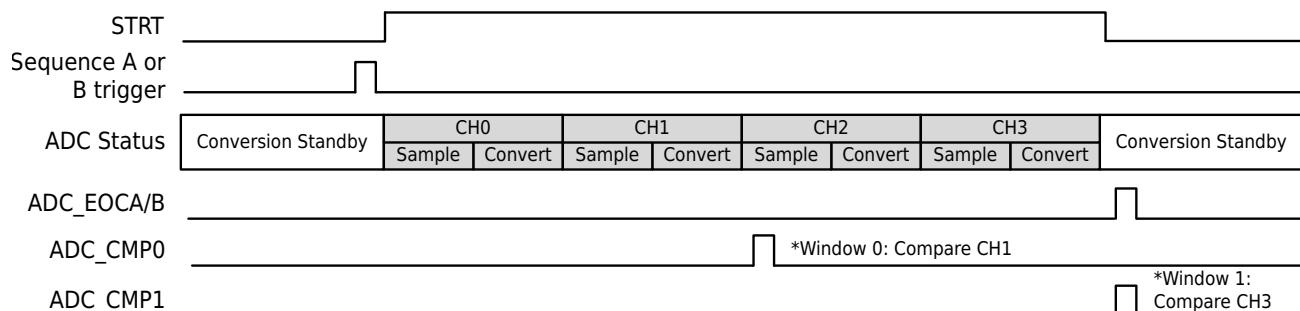


Figure 18-19 ADC Interrupt and Event Output Timing



## 18.4 Register Description

### 18.4.1 Register List

- ADC1 base address: 0x40040000
- ADC2 base address: 0x40040400
- ADC3 base address: 0x40040800

Table 18-4 Analog-to-Digital Converter (ADC) Register Offset

| Offset     | Register     | Description                                                                                                                                                                 | Width |
|------------|--------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 0x00       | ADC_STR      | Start Register                                                                                                                                                              | 8     |
| 0x02       | ADC_CR0      | Control Register 0                                                                                                                                                          | 16    |
| 0x04       | ADC_CR1      | Control Register 1                                                                                                                                                          | 16    |
| 0x06       | ADC_CR2      | Control Register 2                                                                                                                                                          | 16    |
| 0x0A       | ADC_TRGSR    | Conversion Start Trigger Register                                                                                                                                           | 16    |
| 0x0C       | ADC_CHSELRA  | Channel Selection Register A                                                                                                                                                | 32    |
| 0x10       | ADC_CHSELRB  | Channel Selection Register B                                                                                                                                                | 32    |
| 0x14       | ADC_AVCHSELR | Average Channel Selection Register                                                                                                                                          | 32    |
| 0x18       | ADC_EXCHSELR | Extended Channel Selection Register                                                                                                                                         | 8     |
| 0x20+0x1*x | ADC_SSTRx    | Sample Cycle Register (x=0-15)                                                                                                                                              | 8     |
| 0x30       | ADC_SSTRL    | Sample Cycle Register                                                                                                                                                       | 8     |
| 0x38       | ADC_CHMUXR0  | Channel Mapping Control Register 0                                                                                                                                          | 16    |
| 0x3A       | ADC_CHMUXR1  | Channel Mapping Control Register 1                                                                                                                                          | 16    |
| 0x3C       | ADC_CHMUXR2  | Channel Mapping Control Register 2                                                                                                                                          | 16    |
| 0x3E       | ADC_CHMUXR3  | Channel Mapping Control Register 3                                                                                                                                          | 16    |
| 0x44       | ADC_ISR      | Interrupt Status Register                                                                                                                                                   | 8     |
| 0x45       | ADC_ICR      | Interrupt Enable Register                                                                                                                                                   | 8     |
| 0x46       | ADC_ISCLR    | Interrupt Status Clear Register                                                                                                                                             | 8     |
| 0x4C       | ADC_SYNCRR   | Cooperative Mode Control Register                                                                                                                                           | 16    |
| 0x50+0x2*x | ADC_DRx      | ADC Data Register                                                                                                                                                           | 16    |
|            |              |  <b>Note</b><br>"x" represents the channel. For ADC1 and ADC2, x=0-15; for ADC3, x=0-19. |       |
| 0xA0       | ADC_AWDCR    | Analog Watchdog Control Register                                                                                                                                            | 16    |
| 0xA2       | ADC_AWDSR    | Analog Watchdog Status Register                                                                                                                                             | 8     |

| Offset           | Register                      | Description                                                            | Width |
|------------------|-------------------------------|------------------------------------------------------------------------|-------|
| 0xA3             | <a href="#">ADC_AWDSCLRR</a>  | Analog Watchdog Status Clear Register                                  | 8     |
| 0xA4+0x8*x+0x2*y | <a href="#">ADC_AWDxDRy</a>   | Analog Watchdog Window x Threshold Register (x=0-1, y=0-1)             | 16    |
| 0xA8+0x8*x       | <a href="#">ADC_AWDxCHSR</a>  | Analog Watchdog Window x Comparator Channel Selection Register (x=0-1) | 8     |
| 0x1A             | <a href="#">ADC_SHCR</a>      | Dedicated Sample Hold Circuit Control Register                         | 16    |
| 0xC0+0x1*x       | <a href="#">ADC_PGACRx</a>    | Programmable Gain Amplifier Control Register x (x=1-3)                 | 8     |
| 0xC4             | <a href="#">ADC_PGAVSSENr</a> | Programmable Gain Amplifier Reference Ground Selection Register        | 8     |

## 18.4.2 Start Register (ADC\_STR)

| Offset | Bit Position |   |   |   |   |   |   |      |
|--------|--------------|---|---|---|---|---|---|------|
| 0x00   | 7            | 6 | 5 | 4 | 3 | 2 | 1 | 0    |
| Reset  | 0x00         |   |   |   |   |   |   |      |
| Name   | Reserved     |   |   |   |   |   |   | STRT |
| Access |              |   |   |   |   |   |   | RW   |

| Bit | Name     | Function             | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | Access |
|-----|----------|----------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 7:1 | Reserved | Reserved             | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | -      |
| 0   | STRT     | ADC Conversion Start | <ul style="list-style-type: none"> <li>0b0: Stop conversion</li> <li>0b1: Start conversion</li> </ul> <p>Set Conditions:</p> <ol style="list-style-type: none"> <li>Software setting</li> <li>The selected trigger condition occurs</li> <li>During ADC conversion</li> </ol> <p>Clear Conditions:</p> <ol style="list-style-type: none"> <li>Software clearing 0</li> <li>Automatically clears to 0 after conversion ends</li> </ol> <div>  <b>Note</b> <ul style="list-style-type: none"> <li>Writing 1 to STRT when STRT is 0 (ADC idle) generates a software trigger to start sequence A.</li> <li>Writing 1 to STRT has no effect when STRT is 1 (ADC active).</li> <li>Writing 0 to STRT when STRT is 1 forces the ADC conversion to stop. If the ADC_TRGSR sets a value other than 0x0 and does not want the ADC to restart, set the ADC_TRGSR to 0 before writing 0 to the STRT.</li> <li>Write 0 has no effect when STRT is 0.</li> </ul> </div> | RW     |

## 18.4.3 Control Register 0 (ADC\_CR0)

**Note**

Please set this register when ADC\_STR.STRT is 0.

| Offset | Bit Position |    |    |    |    |       |   |   |      |       |        |   |          |    |   |   |
|--------|--------------|----|----|----|----|-------|---|---|------|-------|--------|---|----------|----|---|---|
| 0x02   | 15           | 14 | 13 | 12 | 11 | 10    | 9 | 8 | 7    | 6     | 5      | 4 | 3        | 2  | 1 | 0 |
| Reset  | 0x0000       |    |    |    |    |       |   |   |      |       |        |   |          |    |   |   |
| Name   | Reserved     |    |    |    |    | AVCNT |   |   | DFMT | CLREN | ACCSEL |   | Reserved | MS |   |   |
| Access |              |    |    |    |    | RW    |   |   | RW   | RW    | RW     |   |          | RW |   |   |

| Bit   | Name     | Function                      | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | Access |
|-------|----------|-------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 15:11 | Reserved | Reserved                      | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | -      |
| 10:8  | AVCNT    | Number selection              | <ul style="list-style-type: none"> <li>0b000: Average of 2 consecutive conversions</li> <li>0b001: Average of 4 consecutive conversions</li> <li>0b010: Average of 8 consecutive conversions</li> <li>0b011: Average of 16 consecutive conversions</li> <li>0b100: Average of 32 consecutive conversions</li> <li>0b101: Average of 64 consecutive conversions</li> <li>0b110: Average of 128 consecutive conversions</li> <li>0b111: Average of 256 consecutive conversions</li> </ul> | RW     |
| 7     | DFMT     | Data format                   | <ul style="list-style-type: none"> <li>0b0: Right-align conversion data</li> <li>0b1: Left-align conversion data</li> </ul> <div> <b>Note</b><br/> This bit setting has no effect in oversampling mode. </div>                                                                                                                                                                                                                                                                          | RW     |
| 6     | CLREN    | Data register automatic clear | <ul style="list-style-type: none"> <li>0b0: Disable automatic clear</li> <li>0b1: Enable automatic clear</li> </ul> <div> <b>Note</b><br/> When the CLREN bit is set, the ADC_DRx register will be automatically cleared after being read by the CPU, DMA, etc. The automatic clear function is mainly used to detect whether the data register has been updated. </div>                                                                                                                | RW     |

| Bit | Name            | Function             | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | Access |
|-----|-----------------|----------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 5:4 | ACCSEL          | Resolution selection | <ul style="list-style-type: none"> <li>● 0b00: 12-bit resolution</li> <li>● 0b01: 10-bit resolution</li> <li>● 0b10: 8-bit resolution</li> <li>● 0b11: Setting prohibited</li> </ul> <div>  <b>Note</b><br/> This should be set to 0b00 in oversampling mode. </div>                                                                                                                                                                                  | RW     |
| 3   | <i>Reserved</i> | <i>Reserved</i>      | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | -      |
| 2:0 | MS              | Mode selection       | <ul style="list-style-type: none"> <li>● 0b000: Sequence A single scan mode, sequence B is invalid</li> <li>● 0b001: Sequence A continuous scan mode, sequence B is invalid</li> <li>● 0b010: Sequence A single scan mode, sequence B single scan mode</li> <li>● 0b011: Sequence A continuous scan mode, sequence B single scan mode</li> <li>● 0b100: Sequence A data buffer mode, sequence B is invalid</li> <li>● 0b110: Sequence A data buffer mode, sequence B single scan mode</li> <li>● Others: Setting prohibited</li> </ul> | RW     |

## 18.4.4 Control Register 1 (ADC\_CR1)

**Note**

Please set this register when ADC\_STR.STRT is 0.

| Offset | Bit Position |    |    |    |    |    |   |   |   |   |   |   |   |         |          |   |
|--------|--------------|----|----|----|----|----|---|---|---|---|---|---|---|---------|----------|---|
| 0x04   | 15           | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2       | 1        | 0 |
| Reset  | 0x0000       |    |    |    |    |    |   |   |   |   |   |   |   |         |          |   |
| Name   | Reserved     |    |    |    |    |    |   |   |   |   |   |   |   | RSCHSEL | Reserved |   |
| Access |              |    |    |    |    |    |   |   |   |   |   |   |   | RW      |          |   |

| Bit  | Name     | Function                              | Description                                                                                                                                                                                                                                                                                             | Access |
|------|----------|---------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 15:3 | Reserved | Reserved                              | -                                                                                                                                                                                                                                                                                                       | -      |
| 2    | RSCHSEL  | Sequence A Re-start Channel Selection | <ul style="list-style-type: none"> <li>0b0: After being interrupted by sequence B, when sequence A restarts, it continues scanning from the interrupted channel.</li> <li>0b1: After being interrupted by sequence B, when sequence A restarts, it starts to re-scan from the first channel.</li> </ul> | RW     |
| 1:0  | Reserved | Reserved                              | -                                                                                                                                                                                                                                                                                                       | -      |

## 18.4.5 Control Register 2 (ADC\_CR2)

**Note**

Please set this register when ADC\_STR.STRT is 0.

| Offset | Bit Position |    |    |        |      |    |   |   |          |   |   |   |   |   |   |   |
|--------|--------------|----|----|--------|------|----|---|---|----------|---|---|---|---|---|---|---|
| 0x06   | 15           | 14 | 13 | 12     | 11   | 10 | 9 | 8 | 7        | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x0000       |    |    |        |      |    |   |   |          |   |   |   |   |   |   |   |
| Name   | Reserved     |    |    | OVSMOD | OVSS |    |   |   | Reserved |   |   |   |   |   |   |   |
| Access |              |    |    | RW     | RW   |    |   |   |          |   |   |   |   |   |   |   |

| Bit   | Name     | Function          | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | Access |
|-------|----------|-------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 15:13 | Reserved | Reserved          | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | -      |
| 12    | OVSMOD   | Oversampling Mode | <ul style="list-style-type: none"> <li>0b0: Normal mode</li> <li>0b1: Oversampling mode</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                 | RW     |
| 11:8  | OVSS     | Shift Selection   | <ul style="list-style-type: none"> <li>0b0000: No shift</li> <li>0b0001: Shift right by 1 bit</li> <li>0b0010: Shift right by 2 bit</li> <li>0b0011: Shift right by 3 bit</li> <li>0b0100: Shift right by 4 bit</li> <li>0b0101: Shift right by 5 bit</li> <li>0b0110: Shift right by 6 bit</li> <li>0b0111: Shift right by 7 bit</li> <li>0b1000: Shift right by 8 bit</li> <li>Others: Setting prohibited</li> </ul> <div> <b>Note</b><br/>           This bit setting has no effect when OVSMOD=0 (normal mode).         </div> | RW     |
| 7:0   | Reserved | Reserved          | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | -      |

## 18.4.6 Conversion Start Trigger Register (ADC\_TRGSR)

**Note**

Please set this register when ADC\_STR.STRT is 0.

| Offset | Bit Position |          |    |    |    |    |         |   |        |          |   |   |   |   |         |   |
|--------|--------------|----------|----|----|----|----|---------|---|--------|----------|---|---|---|---|---------|---|
| 0x0A   | 15           | 14       | 13 | 12 | 11 | 10 | 9       | 8 | 7      | 6        | 5 | 4 | 3 | 2 | 1       | 0 |
| Reset  | 0x0000       |          |    |    |    |    |         |   |        |          |   |   |   |   |         |   |
| Name   | TRGENB       | Reserved |    |    |    |    | TRGSELB |   | TRGENA | Reserved |   |   |   |   | TRGSELA |   |
| Access | RW           |          |    |    |    |    | RW      |   | RW     |          |   |   |   |   | RW      |   |

| Bit   | Name     | Function                               | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | Access |
|-------|----------|----------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 15    | TRGENB   | Sequence B Trigger Enabled             | <ul style="list-style-type: none"> <li>0b0: Disable sequence B internal or external pin trigger</li> <li>0b1: Enable sequence B internal or external pin trigger</li> </ul> <div> <b>Note</b><br/>           External pin trigger is selected as active. If ADCx_ADTRG transitions from "High" to "Low", a falling edge is detected, and the scan conversion starts. Please keep "Low" for at least 1.5*PCLK4 periods.         </div>                                                                                                                                            | RW     |
| 14:10 | Reserved | Reserved                               | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | -      |
| 9:8   | TRGSELB  | Sequence B Trigger Condition Selection | In sequence B active mode (ADC_CR0.MS[1]=1), as the trigger condition for sequence B: <ul style="list-style-type: none"> <li>0b00: ADCx_ADTRG (x=1-3, represents the ADC unit number)</li> <li>0b01: IN_TRGx0</li> <li>0b10: IN_TRGx1</li> <li>0b11: IN_TRGx0 + IN_TRGx1</li> </ul> <div> <b>Note</b><br/>           This is only valid in sequence B active mode and invalid in other modes.<br/><br/>           The interval between two triggers must be greater than or equal to the scan period <math>t_{SCAN}</math>, and if less, the trigger is disabled.         </div> | RW     |



| Bit | Name     | Function                               | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | Access |
|-----|----------|----------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 7   | TRGENA   | Sequence A Trigger Enabled             | <ul style="list-style-type: none"> <li>0b0: Disable sequence A internal or external pin trigger</li> <li>0b1: Enable sequence A internal or external pin trigger</li> </ul> <div>  <b>Note</b><br/>           External pin trigger is selected as active. If ADCx_ADTRG transitions from "High" to "Low", a falling edge is detected, and the scan conversion starts. Please keep "Low" for at least 1.5*PCLK4 periods.         </div>                                                                                                                                                                    | RW     |
| 6:2 | Reserved | Reserved                               | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | -      |
| 1:0 | TRGSELA  | Sequence A Trigger Condition Selection | Trigger conditions for sequence A: <ul style="list-style-type: none"> <li>0b00: ADCx_ADTRG (x=1-3, represents the ADC unit number)</li> <li>0b01: IN_TRGx0</li> <li>0b10: IN_TRGx1</li> <li>0b11: IN_TRGx0 + IN_TRGx1</li> </ul> <div>  <b>Note</b><br/>           Writing 1 to ADC_STR.STRT for software trigger when the ADC is idle ignores the settings of TRGENA and TRGSELA, and directly starts ADC conversion.<br/><br/>           The interval between two triggers must be greater than or equal to the scan period <math>t_{SCAN}</math>, and if less, the trigger is disabled.         </div> | RW     |

18.4.7 Channel Selection Register A (ADC\_CHSELRA)



Note

Please set this register when ADC\_STR.STRT is 0.

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0x0C   | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | CHSELA       |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access | RW           |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit  | Name   | Function                     | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | Access |
|------|--------|------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:0 | CHSELA | Conversion channel selection | <div>For sequence A channel selection, each bit represents a channel. CHSELA[x] represents channel CHx, and any combination can be selected.</div> <div><ul style="list-style-type: none"><li>0b0: Corresponding channel is not selected</li><li>0b1: Corresponding channel is selected</li></ul></div> <div><div> Notice</div><div>Please do not select the same channel for both sequences A, B.</div><div>ADC1 and ADC2 support channels 0-15, and ADC3 supports channels 0-19. Unsupported channels need to be set to 0.</div></div> | RW     |

18.4.8 Channel Selection Register B (ADC\_CHSELRB)



Note

Please set this register when ADC\_STR.STRT is 0.

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0x10   | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | CHSELB       |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access | RW           |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit  | Name   | Function                     | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | Access |
|------|--------|------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:0 | CHSELB | Conversion channel selection | <div>For sequence B channel selection, each bit represents a channel. CHSELB[x] represents channel CHx, and any combination can be selected.</div> <div><ul style="list-style-type: none"><li>0b0: Corresponding channel is not selected</li><li>0b1: Corresponding channel is selected</li></ul></div> <div><div> Notice</div><div>This is only valid in dual sequence scan mode. Please do not select the same channel for both sequences A, B.</div><div>ADC1 and ADC2 support channels 0-15, and ADC3 supports channels 0-19. Unsupported channels need to be set to 0.</div></div> | RW     |

## 18.4.9 Average Channel Selection Register (ADC\_AVCHSELR)

**Note**

Please set this register when ADC\_STR.STRT is 0.

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |  |  |  |  |  |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|--|--|--|--|--|--|
| 0x14   | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |  |  |  |  |  |  |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |  |  |  |  |  |
| Name   | AVCHSEL      |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |  |  |  |  |  |
| Access | RW           |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |  |  |  |  |  |

| Bit  | Name    | Function                  | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | Access |
|------|---------|---------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:0 | AVCHSEL | Average channel selection | <p>Each bit represents a channel, AVCHSEL[x] represents channel CHx, and any combination can be selected.</p> <ul style="list-style-type: none"> <li>● 0b0: Corresponding channel is not selected</li> <li>● 0b1: Corresponding channel is selected</li> </ul> <div>  <b>Notice</b><br/>           When AVCHSEL and the corresponding channel in ADC_CHSELRA or ADC_CHSELRB are selected simultaneously, that channel will continuously perform the set number of ADC conversions during scanning, and the conversion results will be averaged and updated into the data register. If the corresponding AVCHSEL channel is not set, the channel will perform a single conversion.<br/><br/>           ADC1 and ADC2 support channels 0-15, and ADC3 supports channels 0-19. Unsupported channels need to be set to 0.         </div> | RW     |

## 18.4.10 Extended Channel Selection Register (ADC\_EXCHSELR)

**Note**

Please set this register when ADC\_STR.STRT is 0.

| Offset | Bit Position |   |   |   |   |   |   |         |
|--------|--------------|---|---|---|---|---|---|---------|
| 0x18   | 7            | 6 | 5 | 4 | 3 | 2 | 1 | 0       |
| Reset  | 0x00         |   |   |   |   |   |   |         |
| Name   | Reserved     |   |   |   |   |   |   | EXCHSEL |
| Access |              |   |   |   |   |   |   | RW      |

| Bit | Name     | Function                   | Description                                                                                                                                                                                                                                                                                                                               | Access |
|-----|----------|----------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 7:1 | Reserved | Reserved                   | -                                                                                                                                                                                                                                                                                                                                         | -      |
| 0   | EXCHSEL  | Extended channel selection | Extended channel analog input source selection: <ul style="list-style-type: none"> <li>0b0: Select external pin</li> <li>0b1: Select internal analog signal</li> </ul> <div> <b>Note</b><br/>           Settings for internal analog signals are described in the relevant sections of the "Power Control (PWC) " chapter.         </div> | RW     |

### 18.4.11 Sample Cycle Register (ADC\_SSTRx) (x=0-15, L)


**Note**

Please set this register when ADC\_STR.STRT is 0.

Offset: 0x20+0x1\*x (SSTR0-15), 0x30 (SSTRL)

| Offset | Bit Position |   |   |   |   |   |   |   |
|--------|--------------|---|---|---|---|---|---|---|
| -      | 7            | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x0B         |   |   |   |   |   |   |   |
| Name   | SST          |   |   |   |   |   |   |   |
| Access | RW           |   |   |   |   |   |   |   |

| Bit | Name | Function            | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | Access |
|-----|------|---------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 7:0 | SST  | Sample period count | <ul style="list-style-type: none"> <li>0x00-0x04: Setting prohibited</li> <li>0x05-0xFF: Correspond to 5-255 PCLK2 clock periods respectively</li> </ul> <p>Channels CH0-15 are set by ADC_SSTRx (x=0-15), and other channels are set by ADC_SSTRL.</p> <div style="border: 1px solid black; padding: 5px; margin-top: 10px;"> <p> <b>Note</b></p> <p>When the PCLK2 frequency is 50MHz, a sampling period is 20ns, and the initial conversion state has 11 sampling periods. Register can be set to adjust sampling time when external input impedance <math>R_{AIN}</math> is too large with insufficient sampling time or PCLK2 frequency is low. The sample time should not be less than 5 periods.</p> <math display="block">SST \geq (R_{AIN} + R_{ADC}) * C_{ADC} * \ln(2^{N+2}) * f_{ADC} + 1</math> <p>Where: <math>R_{AIN}</math> represents external input impedance (<math>\Omega</math>), <math>R_{ADC}</math> represents internal sampling switch resistance (<math>\Omega</math>), <math>C_{ADC}</math> represents internal sampling and retention capacitance (F), N represents AD resolution (12/10/8), and <math>f_{ADC}</math> represents PCLK2 frequency (Hz). For specific information, please refer to the relevant sections of the "Electrical Specifications" chapter in the Datasheet.</p> </div> | RW     |


**Note**

- The sample time for the internal reference voltage should not be less than 1 $\mu$ s.
- The sample time for VBAT divider voltage should not be less than 4 $\mu$ s.
- When the channel dedicated sample hold circuit (SH) is enabled, the corresponding channel's sample time should not be less than 0.4 $\mu$ s.

### 18.4.12 Channel Mapping Control Register (ADC\_CHMUXRx) (x=0-3)


**Note**

Please set this register when ADC\_STR.STRT is 0.

#### ADC\_CHMUXR0 Register

| Offset | Bit Position |    |    |    |         |    |   |   |         |   |   |   |         |   |   |   |
|--------|--------------|----|----|----|---------|----|---|---|---------|---|---|---|---------|---|---|---|
| 0x38   | 15           | 14 | 13 | 12 | 11      | 10 | 9 | 8 | 7       | 6 | 5 | 4 | 3       | 2 | 1 | 0 |
| Reset  | 0x3210       |    |    |    |         |    |   |   |         |   |   |   |         |   |   |   |
| Name   | CH03MUX      |    |    |    | CH02MUX |    |   |   | CH01MUX |   |   |   | CH00MUX |   |   |   |
| Access | RW           |    |    |    | RW      |    |   |   | RW      |   |   |   | RW      |   |   |   |

#### ADC\_CHMUXR1 Register

| Offset | Bit Position |    |    |    |         |    |   |   |         |   |   |   |         |   |   |   |
|--------|--------------|----|----|----|---------|----|---|---|---------|---|---|---|---------|---|---|---|
| 0x3A   | 15           | 14 | 13 | 12 | 11      | 10 | 9 | 8 | 7       | 6 | 5 | 4 | 3       | 2 | 1 | 0 |
| Reset  | 0x7654       |    |    |    |         |    |   |   |         |   |   |   |         |   |   |   |
| Name   | CH07MUX      |    |    |    | CH06MUX |    |   |   | CH05MUX |   |   |   | CH04MUX |   |   |   |
| Access | RW           |    |    |    | RW      |    |   |   | RW      |   |   |   | RW      |   |   |   |

#### ADC\_CHMUXR2 Register

| Offset | Bit Position |    |    |    |         |    |   |   |         |   |   |   |         |   |   |   |
|--------|--------------|----|----|----|---------|----|---|---|---------|---|---|---|---------|---|---|---|
| 0x3C   | 15           | 14 | 13 | 12 | 11      | 10 | 9 | 8 | 7       | 6 | 5 | 4 | 3       | 2 | 1 | 0 |
| Reset  | 0xBA98       |    |    |    |         |    |   |   |         |   |   |   |         |   |   |   |
| Name   | CH11MUX      |    |    |    | CH10MUX |    |   |   | CH09MUX |   |   |   | CH08MUX |   |   |   |
| Access | RW           |    |    |    | RW      |    |   |   | RW      |   |   |   | RW      |   |   |   |

## ADC\_CHMUXR3 Register

| Offset | Bit Position |    |    |    |         |    |   |   |         |   |   |   |         |   |   |   |
|--------|--------------|----|----|----|---------|----|---|---|---------|---|---|---|---------|---|---|---|
| 0x3E   | 15           | 14 | 13 | 12 | 11      | 10 | 9 | 8 | 7       | 6 | 5 | 4 | 3       | 2 | 1 | 0 |
| Reset  | 0xFEDC       |    |    |    |         |    |   |   |         |   |   |   |         |   |   |   |
| Name   | CH15MUX      |    |    |    | CH14MUX |    |   |   | CH13MUX |   |   |   | CH12MUX |   |   |   |
| Access | RW           |    |    |    | RW      |    |   |   | RW      |   |   |   | RW      |   |   |   |

| Bit   | Name   | Function                             | Description                                                                                                                                                                                                                                                                                                                                                                                               | Access |
|-------|--------|--------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 15:12 | CHxMUX | Channel x Mapping Selection (x=0-15) | Select the ADC unit mapping target for CHx. For set value options, refer to <a href="#">Mapping Table of CHx for Different ADC Units</a> .<br><br> <b>Notice</b><br>For non-existent channels, the corresponding bits read as 0 and should be written as 0.<br><br>Please do not set non-existent analog input channels. | RW     |
| 11:8  | CHxMUX |                                      |                                                                                                                                                                                                                                                                                                                                                                                                           | RW     |
| 7:4   | CHxMUX |                                      |                                                                                                                                                                                                                                                                                                                                                                                                           | RW     |
| 3:0   | CHxMUX |                                      |                                                                                                                                                                                                                                                                                                                                                                                                           | RW     |

Table 18-30 Table of Different ADC Unit CHx Mapping Relationships

| Set Value | ADC1 Mapping Objects              | ADC2 Mapping Objects              | ADC3 Mapping Objects             |
|-----------|-----------------------------------|-----------------------------------|----------------------------------|
| 0x0       | ADC123_IN0                        | ADC123_IN0                        | ADC123_IN0                       |
| 0x1       | ADC123_IN1                        | ADC123_IN1                        | ADC123_IN1                       |
| 0x2       | ADC123_IN2                        | ADC123_IN2                        | ADC123_IN2                       |
| 0x3       | ADC123_IN3                        | ADC123_IN3                        | ADC123_IN3                       |
| 0x4       | ADC12_IN4                         | ADC12_IN4                         | ADC3_IN4                         |
| 0x5       | ADC12_IN5                         | ADC12_IN5                         | ADC3_IN5                         |
| 0x6       | ADC12_IN6                         | ADC12_IN6                         | ADC3_IN6                         |
| 0x7       | ADC12_IN7                         | ADC12_IN7                         | ADC3_IN7                         |
| 0x8       | ADC12_IN8                         | ADC12_IN8                         | ADC3_IN8                         |
| 0x9       | ADC12_IN9                         | ADC12_IN9                         | ADC3_IN9                         |
| 0xA       | ADC123_IN10                       | ADC123_IN10                       | ADC123_IN10                      |
| 0xB       | ADC123_IN11                       | ADC123_IN11                       | ADC123_IN11                      |
| 0xC       | ADC123_IN12                       | ADC123_IN12                       | ADC123_IN12                      |
| 0xD       | ADC123_IN13                       | ADC123_IN13                       | ADC123_IN13                      |
| 0xE       | ADC12_IN14                        | ADC12_IN14                        | ADC3_IN14                        |
| 0xF       | ADC12_IN15/Internal analog signal | ADC12_IN15/Internal analog signal | ADC3_IN15/Internal analog signal |



## 18.4.13 Interrupt Status Register (ADC\_ISR)

| Offset | Bit Position |   |   |         |          |   |       |       |
|--------|--------------|---|---|---------|----------|---|-------|-------|
| 0x44   | 7            | 6 | 5 | 4       | 3        | 2 | 1     | 0     |
| Reset  | 0x00         |   |   |         |          |   |       |       |
| Name   | Reserved     |   |   | SASTPDF | Reserved |   | EOCBF | EOCAF |
| Access |              |   |   | RO      |          |   | RO    | RO    |

| Bit | Name            | Function                            | Description                                                                                                                                                                                                                                                                                                                                                                                                                                 | Access |
|-----|-----------------|-------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 7:5 | <i>Reserved</i> | <i>Reserved</i>                     | -                                                                                                                                                                                                                                                                                                                                                                                                                                           | -      |
| 4   | SASTPDF         | Sequence A Interrupt Flag           | <ul style="list-style-type: none"> <li>0b0: Sequence A is not interrupted</li> <li>0b1: Sequence A is interrupted</li> </ul> <div>  <b>Note</b><br/>           In dual sequence scan mode, this bit is set to 1 when sequence A is interrupted by sequence B with higher priority during scanning.         </div>                                          | RO     |
| 3:2 | <i>Reserved</i> | <i>Reserved</i>                     | -                                                                                                                                                                                                                                                                                                                                                                                                                                           | -      |
| 1   | EOCBF           | Sequence B Conversion Complete Flag | <ul style="list-style-type: none"> <li>0b0: The selected channels in sequence B have not been scanned completely</li> <li>0b1: All selected channels in sequence B have been scanned completely</li> </ul> <div>  <b>Note</b><br/>           This bit is set to 1 after all selected channels in sequence B have been scanned completely.         </div> | RO     |
| 0   | EOCAF           | Sequence A Conversion Complete Flag | <ul style="list-style-type: none"> <li>0b0: The selected channels in sequence A have not been scanned completely</li> <li>0b1: All selected channels in sequence A have been scanned completely</li> </ul> <div>  <b>Note</b><br/>           This bit is set to 1 after all selected channels in sequence A have been scanned completely.         </div> | RO     |

## 18.4.14 Interrupt Enable Register (ADC\_ICR)

| Offset | Bit Position |   |   |   |   |   |         |         |
|--------|--------------|---|---|---|---|---|---------|---------|
| 0x45   | 7            | 6 | 5 | 4 | 3 | 2 | 1       | 0       |
| Reset  | 0x03         |   |   |   |   |   |         |         |
| Name   | Reserved     |   |   |   |   |   | EOCBIEN | EOCAIEN |
| Access |              |   |   |   |   |   | RW      | RW      |

| Bit | Name     | Function                                         | Description                                                                                               | Access |
|-----|----------|--------------------------------------------------|-----------------------------------------------------------------------------------------------------------|--------|
| 7:2 | Reserved | Reserved                                         | -                                                                                                         | -      |
| 1   | EOCBIEN  | Sequence B Conversion Complete Interrupt Enabled | <ul style="list-style-type: none"> <li>0b0: Interrupt disabled</li> <li>0b1: Interrupt enabled</li> </ul> | RW     |
| 0   | EOCAIEN  | Sequence A Conversion Complete Interrupt Enabled | <ul style="list-style-type: none"> <li>0b0: Interrupt disabled</li> <li>0b1: Interrupt enabled</li> </ul> | RW     |

## 18.4.15 Interrupt Status Clear Register (ADC\_ISCLRR)

| Offset | Bit Position |   |   |             |          |   |           |           |
|--------|--------------|---|---|-------------|----------|---|-----------|-----------|
| 0x46   | 7            | 6 | 5 | 4           | 3        | 2 | 1         | 0         |
| Reset  | 0x00         |   |   |             |          |   |           |           |
| Name   | Reserved     |   |   | CLR SASTPDF | Reserved |   | CLR EOCBF | CLR EOCAF |
| Access |              |   |   | WO          |          |   | WO        | WO        |

| Bit | Name        | Function                                  | Description                                                                     | Access |
|-----|-------------|-------------------------------------------|---------------------------------------------------------------------------------|--------|
| 7:5 | Reserved    | Reserved                                  | -                                                                               | -      |
| 4   | CLR SASTPDF | Sequence A Interrupt Flag Reset           | Write 1 to clear the SASTPDF bit, write 0 has no effect; read always returns 0. | WO     |
| 3:2 | Reserved    | Reserved                                  | -                                                                               | -      |
| 1   | CLR EOCBF   | Sequence B Conversion Complete Flag Reset | Write 1 to clear the EOCBF bit, write 0 has no effect; read always returns 0.   | WO     |
| 0   | CLR EOCAF   | Sequence A Conversion Complete Flag Reset | Write 1 to clear the EOCAF bit, write 0 has no effect; read always returns 0.   | WO     |

## 18.4.16 Cooperative Mode Control Register (ADC\_SYNCRR)

**Note**

This register is only available in the master ADC (i.e., ADC1) and is not present in other ADC units.

| Offset | Bit Position |    |    |    |    |    |   |   |          |             |   |   |          |   |   |        |
|--------|--------------|----|----|----|----|----|---|---|----------|-------------|---|---|----------|---|---|--------|
| 0x4C   | 15           | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7        | 6           | 5 | 4 | 3        | 2 | 1 | 0      |
| Reset  | 0x0C00       |    |    |    |    |    |   |   |          |             |   |   |          |   |   |        |
| Name   | SYNCDLY      |    |    |    |    |    |   |   | Reserved | SYNCMD[2:0] |   |   | Reserved |   |   | SYNCEN |
| Access | RW           |    |    |    |    |    |   |   |          | RW          |   |   |          |   |   | RW     |

| Bit  | Name        | Function                       | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | Access |
|------|-------------|--------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 15:8 | SYNCDLY     | Synchronization Delay Time     | <p>In delayed trigger mode, the start-up delay time of the two ADCs is <math>t_{\text{SYNCDLY}}</math>.</p> <ul style="list-style-type: none"> <li>● 0x00: Setting prohibited</li> <li>● 0x01: <math>t_{\text{SYNCDLY}} = \text{PCLK2 clock period} \times 1</math></li> <li>● .....</li> <li>● 0xFF: <math>t_{\text{SYNCDLY}} = \text{PCLK2 clock period} \times 255</math></li> </ul> <div> <b>Note</b><br/> Please set this register when SYNCEN is 0.<br/> Set a reasonable delay time based on the sample time and conversion time of each ADC to avoid increased errors caused by multiple ADCs being in sampling state simultaneously and to prevent synchronization failure due to triggers occurring before the ADC conversion is complete. The recommended settings are as follows: <ul style="list-style-type: none"> <li>● Single delay trigger mode: <math>t_{\text{SYNCDLY}} &gt; t_{\text{SPL}}</math></li> <li>● For two ADCs in loop delay trigger mode: <math>t_{\text{SYNCDLY}} &gt; t_{\text{SPL}}</math>, and <math>t_{\text{SYNCDLY}} &gt; t_{\text{SCAN}}/2</math></li> <li>● Three ADC loop delay trigger mode: <math>t_{\text{SYNCDLY}} &gt; t_{\text{SPL}}</math>, and <math>t_{\text{SYNCDLY}} &gt; t_{\text{SCAN}}/3</math></li> <li>● Single parallel trigger mode: This register setting has no effect.</li> <li>● Loop parallel trigger mode: <math>t_{\text{SYNCDLY}} &gt; t_{\text{SCAN}}</math></li> </ul> </div> | RW     |
| 7    | Reserved    | Reserved                       | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | -      |
| 6:4  | SYNCMD[2:0] | Synchronization Mode Selection | <p>SYNCMD[2]:</p> <ul style="list-style-type: none"> <li>● 0b0: Single trigger</li> <li>● 0b1: Loop trigger</li> </ul> <p>SYNCMD[1]:</p> <ul style="list-style-type: none"> <li>● 0b0: Delay trigger mode</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | RW     |

| Bit | Name            | Function                       | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | Access |
|-----|-----------------|--------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 6:4 | SYNCMD[2:0]     | Synchronization Mode Selection | <ul style="list-style-type: none"> <li>0b1: Parallel trigger mode</li> </ul> SYNCMD[0]: <ul style="list-style-type: none"> <li>0b0: ADC1 and ADC2 work synchronously, ADC3 works independently</li> <li>0b1: ADC1, ADC2, and ADC3 work synchronously</li> </ul> <div>  <b>Note</b><br/>           Please set this register when SYNCEN is 0.<br/>           When using a single trigger, please set the ADC to be synchronized to sequence A single scan, or sequence A continuous scan mode.<br/>           When using loop trigger mode, set the ADC to sequence A single scan mode.         </div>                                                                                                                                          | RW     |
| 3:1 | <i>Reserved</i> | <i>Reserved</i>                | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | -      |
| 0   | SYNCEN          | Synchronous Mode License       | <ul style="list-style-type: none"> <li>0b0: Synchronization mode disabled</li> <li>0b1: Synchronization mode enabled</li> </ul> <div>  <b>Note</b><br/>           Synchronization mode only supports sequence A.<br/>           Before SYNCEN writes 1, please turn off sequence B of several ADCs involved in synchronization (ADC_CR0.MS[1]=0), and select the same number of channels for sequence A, and set the same channel sampling time ADC_SSTRx. In order to avoid subsequent synchronization failures causing by the inconsistent scanning time <math>t_{SCAN}</math> of each ADC.<br/>           When the software writes 0 to ADC1_STR.STRT to force the conversion to stop, SYNCEN is automatically cleared to 0.         </div> | RW     |

### 18.4.17 ADC Data Register (ADC\_DRx)

This register is a read-only register used to store the ADC conversion data for each channel. The storage method of conversion result data varies depending on the oversampling mode, data alignment, and conversion resolution.

Offset:  $0x50 + 0x2 \times x$  ("x" represents the channel. For ADC1 and ADC2,  $x=0-15$ ; for ADC3,  $x=0-19$ )

#### Oversampling Mode

| Offset | Bit Position |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| -      | 15           | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x0000       |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | AD[15:0]     |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access | RO           |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

#### Data Right Alignment - 12-bit Resolution

| Offset | Bit Position |    |    |    |          |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----------|----|---|---|---|---|---|---|---|---|---|---|
| -      | 15           | 14 | 13 | 12 | 11       | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x0000       |    |    |    |          |    |   |   |   |   |   |   |   |   |   |   |
| Name   | Reserved     |    |    |    | AD[11:0] |    |   |   |   |   |   |   |   |   |   |   |
| Access |              |    |    |    | RO       |    |   |   |   |   |   |   |   |   |   |   |

#### Data Right Alignment - 10-bit Resolution

| Offset | Bit Position |    |    |    |    |    |         |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|---------|---|---|---|---|---|---|---|---|---|
| -      | 15           | 14 | 13 | 12 | 11 | 10 | 9       | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x0000       |    |    |    |    |    |         |   |   |   |   |   |   |   |   |   |
| Name   | Reserved     |    |    |    |    |    | AD[9:0] |   |   |   |   |   |   |   |   |   |
| Access |              |    |    |    |    |    | RO      |   |   |   |   |   |   |   |   |   |

**Data Right Alignment - 8-bit Resolution**

| Offset | Bit Position |    |    |    |    |    |   |   |         |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|---|---|---------|---|---|---|---|---|---|---|
| -      | 15           | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7       | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x0000       |    |    |    |    |    |   |   |         |   |   |   |   |   |   |   |
| Name   | Reserved     |    |    |    |    |    |   |   | AD[7:0] |   |   |   |   |   |   |   |
| Access |              |    |    |    |    |    |   |   | RO      |   |   |   |   |   |   |   |

**Data Left Alignment - 12-bit Resolution**

| Offset | Bit Position |    |    |    |    |    |   |   |   |   |   |   |          |   |   |   |
|--------|--------------|----|----|----|----|----|---|---|---|---|---|---|----------|---|---|---|
| -      | 15           | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3        | 2 | 1 | 0 |
| Reset  | 0x0000       |    |    |    |    |    |   |   |   |   |   |   |          |   |   |   |
| Name   | AD[11:0]     |    |    |    |    |    |   |   |   |   |   |   | Reserved |   |   |   |
| Access | RO           |    |    |    |    |    |   |   |   |   |   |   |          |   |   |   |

**Data Left Alignment - 10-bit Resolution**

| Offset | Bit Position |    |    |    |    |    |   |   |   |   |          |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|---|---|---|---|----------|---|---|---|---|---|
| -      | 15           | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5        | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x0000       |    |    |    |    |    |   |   |   |   |          |   |   |   |   |   |
| Name   | AD[9:0]      |    |    |    |    |    |   |   |   |   | Reserved |   |   |   |   |   |
| Access | RO           |    |    |    |    |    |   |   |   |   |          |   |   |   |   |   |

**Data Left Alignment - 8-bit Resolution**

| Offset | Bit Position |    |    |    |    |    |   |   |          |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|---|---|----------|---|---|---|---|---|---|---|
| -      | 15           | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7        | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x0000       |    |    |    |    |    |   |   |          |   |   |   |   |   |   |   |
| Name   | AD[7:0]      |    |    |    |    |    |   |   | Reserved |   |   |   |   |   |   |   |
| Access | RO           |    |    |    |    |    |   |   |          |   |   |   |   |   |   |   |

## 18.4.18 Analog Watchdog Control Register (ADC\_AWDCR)

| Offset | Bit Position |    |    |    |    |    |       |   |          |        |         |        |          |        |         |        |
|--------|--------------|----|----|----|----|----|-------|---|----------|--------|---------|--------|----------|--------|---------|--------|
| 0xA0   | 15           | 14 | 13 | 12 | 11 | 10 | 9     | 8 | 7        | 6      | 5       | 4      | 3        | 2      | 1       | 0      |
| Reset  | 0x0000       |    |    |    |    |    |       |   |          |        |         |        |          |        |         |        |
| Name   | Reserved     |    |    |    |    |    | AWDCM |   | Reserved | AWD1MD | AWD1IEN | AWD1EN | Reserved | AWD0MD | AWD0IEN | AWD0EN |
| Access |              |    |    |    |    |    | RW    |   |          | RW     | RW      | RW     |          | RW     | RW      | RW     |

| Bit   | Name     | Function                              | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | Access |
|-------|----------|---------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 15:10 | Reserved | Reserved                              | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | -      |
| 9:8   | AWDCM    | Watchdog Window Combination Selection | <ul style="list-style-type: none"> <li>0b00: Window combination inactive, ADC_CMP1 outputs independent comparison results for Window 1</li> <li>0b01: Window combination active, ADC_CMP1 outputs logical OR of comparison results for Window 0 and Window 1</li> <li>0b10: Window combination active, ADC_CMP1 outputs logical AND of comparison results for Window 0 and Window 1</li> <li>0b11: Window combination active, ADC_CMP1 outputs logical XOR of comparison results for Window 0 and Window 1</li> </ul> <div>  <b>Note</b><br/>           When using the window combination function, both Window 0 and Window 1 must be enabled (i.e., AWD0EN is written to 1, AWD1EN is written to 1). If the comparison channels selected for Window 0 and Window 1 are different, please ensure that during the scan conversion process, the channel selected for Window 1 is converted after the channel selected for Window 0. The ADC_CMP1 interrupt or event is output when the conversion of the channel selected for Window 1 ends.         </div> | RW     |
| 7     | Reserved | Reserved                              | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | -      |
| 6     | AWD1MD   | Compare Mode of Watchdog Window 1     | Window 1 Protected Area Selection: <ul style="list-style-type: none"> <li>0b0: Conversion result &lt; AWD1DR0, or conversion result &gt; AWD1DR1</li> <li>0b1: Conversion result ≥ AWD1DR0, and conversion result ≤ AWD1DR1</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | RW     |
| 5     | AWD1IEN  | Watchdog Window 1 Interrupt Enabled   | <ul style="list-style-type: none"> <li>0b0: Watchdog Window 1 comparison interrupt ADC_CMP1 inactive</li> <li>0b1: Watchdog Window 1 comparison interrupt ADC_CMP1 enabled</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | RW     |



| Bit | Name            | Function                            | Description                                                                                                                                                                                                                                                            | Access |
|-----|-----------------|-------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 4   | AWD1EN          | Watchdog Window 1 Compare Enabled   | <ul style="list-style-type: none"> <li>0b0: Watchdog Window 1 comparison function inactive</li> <li>0b1: Watchdog Window 1 comparison function enabled</li> </ul>                                                                                                      | RW     |
| 3   | <i>Reserved</i> | <i>Reserved</i>                     | -                                                                                                                                                                                                                                                                      | -      |
| 2   | AWD0MD          | Compare Mode of Watchdog Window 0   | Window 0 Protected Area Selection: <ul style="list-style-type: none"> <li>0b0: Conversion result &lt; AWD0DR0, or conversion result &gt; AWD0DR1</li> <li>0b1: Conversion result <math>\geq</math> AWD0DR0, and conversion result <math>\leq</math> AWD0DR1</li> </ul> | RW     |
| 1   | AWD0IEN         | Watchdog Window 0 Interrupt Enabled | <ul style="list-style-type: none"> <li>0b0: Watchdog Window 0 comparison interrupt ADC_CMP0 inactive</li> <li>0b1: Watchdog Window 0 comparison interrupt ADC_CMP0 enabled</li> </ul>                                                                                  | RW     |
| 0   | AWD0EN          | Watchdog Window 0 Compare Enabled   | <ul style="list-style-type: none"> <li>0b0: Watchdog Window 0 comparison function inactive</li> <li>0b1: Watchdog Window 0 comparison function enabled</li> </ul>                                                                                                      | RW     |

## 18.4.19 Analog Watchdog Status Register (ADC\_AWDSR)

| Offset | Bit Position |   |   |        |          |   |       |       |
|--------|--------------|---|---|--------|----------|---|-------|-------|
| 0xA2   | 7            | 6 | 5 | 4      | 3        | 2 | 1     | 0     |
| Reset  | 0x00         |   |   |        |          |   |       |       |
| Name   | Reserved     |   |   | AWDCMF | Reserved |   | AWD1F | AWD0F |
| Access |              |   |   | RO     |          |   | RO    | RO    |

| Bit | Name            | Function                                       | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | Access |
|-----|-----------------|------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 7:5 | <i>Reserved</i> | <i>Reserved</i>                                | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | -      |
| 4   | AWDCMF          | Watchdog window combination compare status bit | <ul style="list-style-type: none"> <li>0b0: Compare result of Window 0 and Window 1 does not meet the combination condition</li> <li>0b1: Compare result of Window 0 and Window 1 meets the combination condition</li> </ul> <div>  <b>Note</b><br/>           When the watchdog window combination comparison function is enabled (i.e., ADC_AWDCR.AWDCM[1:0]=0b01/0b10/0b11), this bit is set to 1 when the conversion of the channel selected for Window 1 ends and the comparison result of Window 0 and Window 1 meets the combination condition.         </div> | RO     |
| 3:2 | <i>Reserved</i> | <i>Reserved</i>                                | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | -      |
| 1   | AWD1F           | Watchdog window 1 comparative status position  | <ul style="list-style-type: none"> <li>0b0: Conversion result does not meet the comparison condition</li> <li>0b1: Conversion result meets the comparison condition</li> </ul> <div>  <b>Note</b><br/>           This bit is set to 1 when the conversion of the channel selected for Window 1 ends and the conversion result meets the comparison condition.         </div>                                                                                                                                                                                        | RO     |
| 0   | AWD0F           | Watchdog window 0 comparative status position  | <ul style="list-style-type: none"> <li>0b0: Conversion result does not meet the comparison condition</li> <li>0b1: Conversion result meets the comparison condition</li> </ul> <div>  <b>Note</b><br/>           This bit is set to 1 when the conversion of the channel selected for Window 0 ends and the conversion result meets the comparison condition.         </div>                                                                                                                                                                                        | RO     |

## 18.4.20 Analog Watchdog Status Clear Register (ADC\_AWDSCCLR)

| Offset | Bit Position |   |   |           |          |   |         |         |
|--------|--------------|---|---|-----------|----------|---|---------|---------|
| 0xA3   | 7            | 6 | 5 | 4         | 3        | 2 | 1       | 0       |
| Reset  | 0x00         |   |   |           |          |   |         |         |
| Name   | Reserved     |   |   | CLRAWDCMF | Reserved |   | CLRAW1F | CLRAW0F |
| Access |              |   |   | WO        |          |   | WO      | WO      |

| Bit | Name      | Function                                             | Description                                                                     | Access |
|-----|-----------|------------------------------------------------------|---------------------------------------------------------------------------------|--------|
| 7:5 | Reserved  | Reserved                                             | -                                                                               | -      |
| 4   | CLRAWDCMF | Watchdog Window Combination Compare Status Bit Reset | Write 1 to reset AWD1F status bit, write 0 has no effect; read always returns 0 | WO     |
| 3:2 | Reserved  | Reserved                                             | -                                                                               | -      |
| 1   | CLRAW1F   | Watchdog Window 1 Compare Status Bit Reset           | Write 1 to reset AWD1F status bit, write 0 has no effect; read always returns 0 | WO     |
| 0   | CLRAW0F   | Watchdog Window 0 Compare Status Bit Reset           | Write 1 to reset AWD0F status bit, write 0 has no effect; read always returns 0 | WO     |

### 18.4.21 Analog Watchdog Window x Threshold Register (ADC\_AWDxDRy) (x=0-1, y=0-1)

Offset:  $0xA4 + 0x8 * x + 0x2 * y$

| Offset | Bit Position                       |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|------------------------------------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| -      | 15                                 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x0000 (AWDxDR0); 0xFFFF (AWDxDR1) |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | AWDDR                              |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access | RW                                 |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit  | Name  | Function         | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | Access |
|------|-------|------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 15:0 | AWDDR | Comparative Data | <p>AWD0DR0 sets the low threshold for Window 0, and AWD0DR1 sets the high threshold for Window 0.</p> <p>AWD1DR0 sets the low threshold for Window 1, and AWD1DR1 sets the high threshold for Window 1.</p> <p>AWD0DR0, AWD0DR1, AWD1DR0 and AWD1DR1 vary depending on the alignment (right-aligned or left-aligned data) and resolution (12-bit, 10-bit, or 8-bit).</p> <ul style="list-style-type: none"> <li>● Right-aligned data-12-bit resolution: Low 12 bits [11:0] are available</li> <li>● Right-aligned data-10-bit resolution: Low 10 bits [9:0] are available</li> <li>● Right-aligned data-8-bit resolution: Low 8 bits [7:0] are available</li> <li>● Left-aligned data-12-bit resolution: High 12 bits [15:4] are available</li> <li>● Left-aligned data-10-bit resolution: High 10 bits [15:6] are available</li> <li>● Left-aligned data-8-bit resolution: High 8 bits [15:8] are available</li> </ul> | RW     |

### 18.4.22 Analog Watchdog Window x Comparator Channel Selection Register (ADC\_AWDxCHSR) (x=0-1)

Offset: 0xA8+0x8\*x

| Offset | Bit Position |   |   |       |   |   |   |   |
|--------|--------------|---|---|-------|---|---|---|---|
| -      | 7            | 6 | 5 | 4     | 3 | 2 | 1 | 0 |
| Reset  | 0x00         |   |   |       |   |   |   |   |
| Name   | Reserved     |   |   | AWDCH |   |   |   |   |
| Access |              |   |   | RW    |   |   |   |   |

| Bit | Name            | Function                           | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | Access |
|-----|-----------------|------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 7:5 | <i>Reserved</i> | <i>Reserved</i>                    | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | -      |
| 4:0 | AWDCH           | Watchdog Compare Channel Selection | Select the comparison channel for Window x. <ul style="list-style-type: none"> <li>● 0x00: CH0 (Channel 0)</li> <li>● 0x01: CH1 (Channel 1)</li> <li>● .....</li> <li>● 0x0F: CH15 (Channel 15)</li> <li>● .....</li> <li>● 0x13: CH19 (Channel 19)</li> <li>● Others: Setting prohibited</li> </ul> <div>  <b>Notice</b><br/>           Please do not set non-existent channels. ADC1 and ADC2 support channels 0-15, and ADC3 supports channels 0-19.         </div> | RW     |

## 18.4.23 Dedicated Sample Hold Circuit Control Register (ADC\_SHCR)

**Note**

- Unit 1 (ADC1) supports 3 channels for SH1-3. Please refer to [Mapping Relationship with Physical Channels](#). Register ADC\_SHCR.SHSEL[0] corresponds to SH1, SHSEL[1] corresponds to SH2, and SHSEL[2] corresponds to SH3.
- Unit 2 (ADC2) does not support SH.
- Unit 3 (ADC3) does not support SH.

| Offset | Bit Position |    |    |    |    |       |   |   |       |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|-------|---|---|-------|---|---|---|---|---|---|---|
| 0x1A   | 15           | 14 | 13 | 12 | 11 | 10    | 9 | 8 | 7     | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x0018       |    |    |    |    |       |   |   |       |   |   |   |   |   |   |   |
| Name   | Reserved     |    |    |    |    | SHSEL |   |   | SHSST |   |   |   |   |   |   |   |
| Access |              |    |    |    |    | RW    |   |   | RW    |   |   |   |   |   |   |   |

| Bit   | Name     | Function                                        | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | Access |
|-------|----------|-------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 15:11 | Reserved | Reserved                                        | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | -      |
| 10:8  | SHSEL    | Dedicated Sample Hold Circuit Channel Selection | Corresponding to SH3-SH1: <ul style="list-style-type: none"> <li>● 0b0: Bypass the dedicated sample hold circuit</li> <li>● 0b1: Enable the dedicated sample hold circuit</li> </ul> <div> <b>Note</b><br/>           Each bit corresponds to one channel.<br/><br/>           In dual sequence scan mode, do not use the dedicated sample hold circuit in sequence A. This is because sequence A may be interrupted by sequence B, resulting in unexpected conversion results by the inconsistent sampling times for each dedicated sample hold circuit.         </div> | RW     |
| 7:0   | SHSST    | Sample Time Selection                           | <ul style="list-style-type: none"> <li>● 0x00-0x03: Setting prohibited</li> <li>● 0x04: PCLK2 Clock Cycle×4</li> <li>● .....</li> <li>● 0xFF: PCLK2 Clock Cycle×255</li> </ul> <div> <b>Note</b><br/>           When PCLK2 operates at 50MHz, one period is 20ns. The initial period setting is 24 periods. When the input impedance is high, resulting in insufficient sample time, or when the PCLK2 frequency is low, the sample time can be adjusted. The sample time should not be less than 0.4μs.         </div>                                                  | RW     |

### 18.4.24 Programmable Gain Amplifier Control Register (ADC\_PGACRx) (x=1-3)



#### Note

- Unit 1 (ADC1) supports 3 channels for PGA1-3. Please refer to [Mapping Relationship with Physical Channels](#). The ADC1\_PGACR1-3 registers correspond to the control of PGA1-3, respectively.
- Unit 2 (ADC2) supports 1 channel for PGA4. Please refer to [Mapping Relationship with Physical Channels](#). The ADC2\_PGACR1 register corresponds to the control of PGA4.
- Unit 3 (ADC3) does not support PGA.

Offset: 0xC0+0x1\*x

| Offset | Bit Position |   |   |   |        |   |   |   |
|--------|--------------|---|---|---|--------|---|---|---|
| -      | 7            | 6 | 5 | 4 | 3      | 2 | 1 | 0 |
| Reset  | 0x00         |   |   |   |        |   |   |   |
| Name   | PGAGAIN      |   |   |   | PGACTL |   |   |   |
| Access | RW           |   |   |   | RW     |   |   |   |

| Bit | Name    | Function          | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | Access |
|-----|---------|-------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 7:4 | PGAGAIN | Gain Setting      | <ul style="list-style-type: none"> <li>● 0b0000: ×2.000</li> <li>● 0b0001: ×2.133</li> <li>● 0b0010: ×2.286</li> <li>● 0b0011: ×2.667</li> <li>● 0b0100: ×2.909</li> <li>● 0b0101: ×3.200</li> <li>● 0b0110: ×3.556</li> <li>● 0b0111: ×4.000</li> <li>● 0b1000: ×4.571</li> <li>● 0b1001: ×5.333</li> <li>● 0b1010: ×6.400</li> <li>● 0b1011: ×8.000</li> <li>● 0b1100: ×10.667</li> <li>● 0b1101: ×16.000</li> <li>● 0b1110: ×32.000</li> <li>● 0b1111: Setting prohibited</li> </ul> | RW     |
| 3:0 | PGACTL  | Amplifier Control | <ul style="list-style-type: none"> <li>● 0b0000: Amplifier disabled</li> <li>● 0b1110: Amplifier enabled, signal amplified according to the PGAGAIN[3:0] setting</li> <li>● Others: Setting prohibited</li> </ul>                                                                                                                                                                                                                                                                       | RW     |

### 18.4.25 Programmable Gain Amplifier Reference Ground Selection Register (ADC\_PGAVSS-SENr)


**Note**

- Units 1 (ADC1) and 2 (ADC2) support PGA.
- Unit 3 (ADC3) does not support PGA.

| Offset | Bit Position |   |   |   |   |           |   |   |
|--------|--------------|---|---|---|---|-----------|---|---|
| 0xC4   | 7            | 6 | 5 | 4 | 3 | 2         | 1 | 0 |
| Reset  | 0x00         |   |   |   |   |           |   |   |
| Name   | Reserved     |   |   |   |   | PGAVSSSEN |   |   |
| Access |              |   |   |   |   | RW        |   |   |

| Bit | Name      | Function                             | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | Access |
|-----|-----------|--------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 7:3 | Reserved  | Reserved                             | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | -      |
| 2:0 | PGAVSSSEN | Amplifier Reference Ground Selection | <ul style="list-style-type: none"> <li>● 0b0: Use external pin PGAx_VSS as the PGA reference ground</li> <li>● 0b1: Use analog ground AVSS as the PGA reference ground</li> </ul> Where: <ul style="list-style-type: none"> <li>● For Unit 1, ADC1_PGAVSSSEN.PGAVSSSEN[0] selects the PGA1 reference ground as PGA123_VSS or AVSS.</li> <li>● For Unit 1, ADC1_PGAVSSSEN.PGAVSSSEN[1] selects the PGA2 reference ground as PGA123_VSS or AVSS.</li> <li>● For Unit 1, ADC1_PGAVSSSEN.PGAVSSSEN[2] selects the PGA3 reference ground as PGA123_VSS or AVSS.</li> <li>● For Unit 2, ADC2_PGAVSSSEN.PGAVSSSEN[0] selects the PGA4 reference ground as PGA4_VSS or AVSS.</li> <li>● For Unit 2, ADC2_PGAVSSSEN.PGAVSSSEN[2:1] does not exist.</li> </ul> | RW     |



## 18.5 Precautions

### 18.5.1 Precautions for Reading Data Registers

Please access the ADC data registers (ADC\_DR) in half-word units. Do not access data registers in byte units.

### 18.5.2 Precautions for Handling Scan Completion Interrupts

When two consecutive scan conversions are performed on the same channel, during the period from the completion of interrupt handling in the first conversion to the completion of interrupt handling in the second conversion, if the CPU does not read the first converted data in time, the second converted data will overwrite the first converted data.

### 18.5.3 Precautions for Module Stop and Low Power Settings

The ADC module can be stopped and power consumption reduced by setting the PWC\_FCG register. The ADC is initially in the stopped state. To start the ADC module, first set the corresponding bit in the PWC\_FCG register to cancel the stop, and wait for 1 $\mu$ s before starting the ADC conversion.

Before setting the module to stop, please confirm that the ADC is in the conversion stop state, i.e., the ADC\_STR.STRT bit is 0.

Before setting the system to enter STOP mode, please set the ADC to module Stop mode first.

For specific descriptions, please refer to the relevant sections of the "Action Mode and Low Power Mode" section in the "Power Control (PWC) " chapter.

### 18.5.4 ADC Conversion Analog Channel Input Pin Settings

When setting a chip pin as an ADC analog channel input, please first disable the digital function (PCRxy.DDIS) of the corresponding pin. For specific descriptions, please refer to the "General-Purpose I/O (GPIO)" chapter.

### 18.5.5 Noise Control

To prevent abnormal voltages such as surges from damaging the analog input pin, it is recommended to use the protection circuit shown in the "Electrical Specifications" chapter in the Datasheet.

## 19 Digital-to-Analog Converter (DAC)

### 19.1 Introduction

This MCU is equipped with 2 12-bit digital-to-analog converter (DAC) unit, DAC1 and DAC2. Each DAC unit includes two DAC conversion channels that can operate independently or synchronously. The analog voltage output range has two selectable settings. Each conversion channel has an output amplifier to drive external loads directly without an external operational amplifier. Independent pin inputs for reference voltages VREFH and VREFL can improve conversion accuracy.

### 19.2 Features

- 2 DAC units, each with 2 DAC conversion channels.
- 12-bit conversion data can be configured as left- or right-aligned.
- Both conversion channels in the same DAC module can perform synchronous conversion.
- The data computing unit (DCU) can output triangular and sawtooth waves.
- The output can be used as the negative-end voltage for the voltage comparator (CMP).
- The output has amplification function and can drive external loads directly.
- The ADC conversion priority mode reduces interference during ADC conversion.
- Independent pin inputs for reference voltages VREFH/VREFL

### 19.3 Functional Description

#### 19.3.1 Function Block Diagram

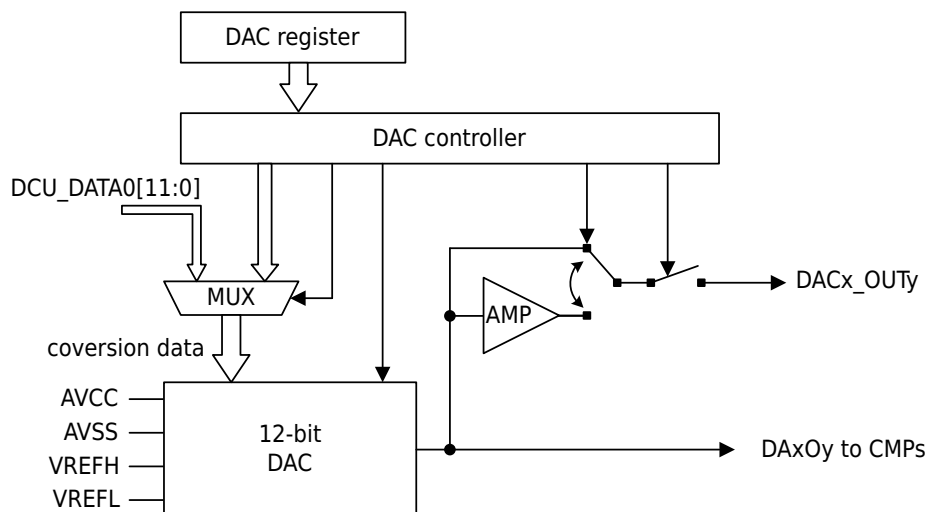


Figure 19-1 DAC Conversion Channel Block Diagram

Table 19-1 DAC pin

| Pin Name | Input/Output | Function            |
|----------|--------------|---------------------|
| AVCC     | Input        | Analog Power Supply |
| AVSS     | Input        | Analog Ground       |
| VREFH    | Input        | Reference Voltage   |
| VREFL    | Input        | Reference Ground    |

| Pin Name  | Input/Output | Function                                  |
|-----------|--------------|-------------------------------------------|
| DACx_OUTy | Output       | DAC Converts Analog Output (x=1-2, y=1-2) |

**Note**

In subsequent sections of this chapter, x refers to the unit number, and y refers to the channel number.

### 19.3.2 DAC Conversion

Each conversion channel can perform DAC conversion independently. When DAC\_DACR.DAyE is set to 1, DAC conversion starts for the corresponding channel and the conversion result is output from DACx\_OUTy. If only the conversion result is used as the negative input voltage of the voltage comparator CMP, the DACx\_OUTy output can be turned off through the DAC\_DAOCR register. The DAC conversion action timing is as follows:

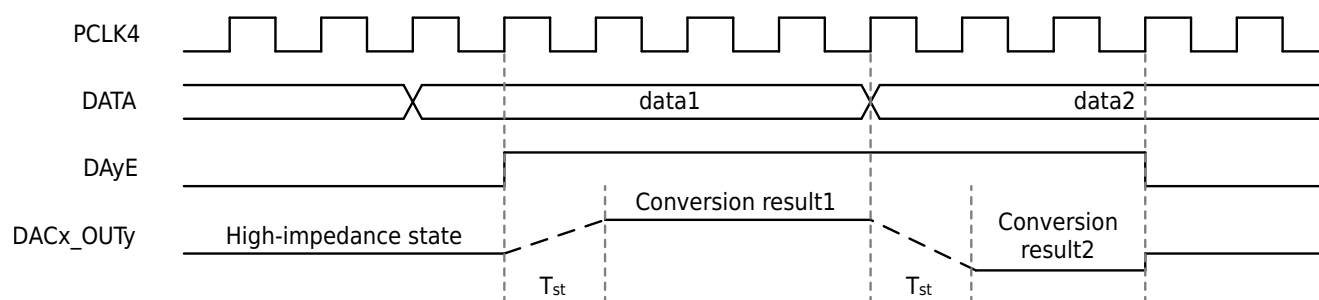


Figure 19-2 DAC Conversion Diagram

Using DAC1 Channel 1 for single-channel conversion as an example:

1. Set DAC conversion data (DAC\_DADR1) and data format (DAC\_DACR.DPSEL).
2. Set DAC\_DACR.DA1E to 1 to start DAC conversion. After the conversion time  $T_{st}$ , the conversion result is output from port DAC1\_OUT1 and is held until the value of DAC\_DADR1 is rewritten or DAC\_DACR.DA1E is set to 0. The output analog voltage can be calculated using the following formula:

$$DAC_{output} = \frac{\text{Conversion}}{4096} \times VREFH$$

When DAC\_DACR.DA1E is 1, rewriting DAC\_DADR1 triggers a new DAC conversion. In the same way, after conversion time  $T_{st}$ , the new conversion results are output from DAC1\_OUT1.

3. Setting DAC\_DACR.DA1E to 0 disables DAC1 Channel 1, and DAC1\_OUT1 outputs a high-impedance state.

### 19.3.3 Synchronization Conversion

Synchronized conversion of both channels in the same DAC unit can be achieved through synchronized updates of conversion data.

Taking the two-channel synchronized conversion using DAC1 as an example, this section describes the synchronization conversion process.

1. Set DAC conversion data (DAC\_DADR1, DAC\_DADR2) and data format (DAC\_DACR.DPSEL).
2. Set DAC\_DACR.DAE to 1 to start synchronous DAC conversion for both channels. After conversion time  $T_{st}$ , the conversion results are output from DAC1\_OUT1 and DAC1\_OUT2 and remain until DADR is rewritten or DAC\_DACR.DAE is set to 0.

- When DAC\_DACR.DAE is 1, simultaneously rewriting DAC\_DADRx1 and DAC\_DADRx2 triggers a new DAC conversion for both channels. In the same way, after conversion time  $T_{st}$ , the new conversion results are output from DAC1\_OUT1 and DAC1\_OUT2.
- Setting DAC\_DACR.DAE to 0 disables both channels, and DAC1\_OUT1 and DAC1\_OUT2 output a high-impedance state.

### 19.3.4 External Data Conversion

In addition to converting data in DAC\_DADRx, the DAC can also convert data from the data computing unit (DCU). DAC1 Channel 1 and Channel 2 correspond to DCU units 1 and 2, respectively, and DAC2 Channel 1 and Channel 2 correspond to DCU units 3 and 4, respectively. Setting DAC\_DACR.EXTDSL to 1 selects the lower 12 bits of data register 0 in the corresponding DCU unit for conversion.

When converting external data, the output amplifier and ADC conversion priority mode are disabled.

### 19.3.5 ADC Conversion Priority Mode

When the DAC starts conversion, a transient inrush current may occur on the analog power supply, which can interfere with ongoing ADC conversions. The ADC conversion priority mode actively avoids this interference by changing the update timing of DAC conversion data.

Setting DAC\_DAADPCR.ADPEN to 1 enables ADC conversion priority mode:

- If DAC\_DADRx is rewritten during ADC conversion, DAC conversion does not start immediately but waits until ADC conversion is complete. The maximum delay from rewriting DAC\_DADRx to starting DAC conversion is one ADC conversion period, during which the DAC\_DADRx value and analog output do not match.
- If DAC\_DADRx is rewritten when ADC is stopped (ADC\_STR.STRT=0), DAC conversion starts after 2 ADCLK periods.

DAC\_DAADPCR.ADCSLn (n=1-3) is used to select the ADC channel with priority for ADC conversion. The ADC conversion priority mode action timing is as follows:

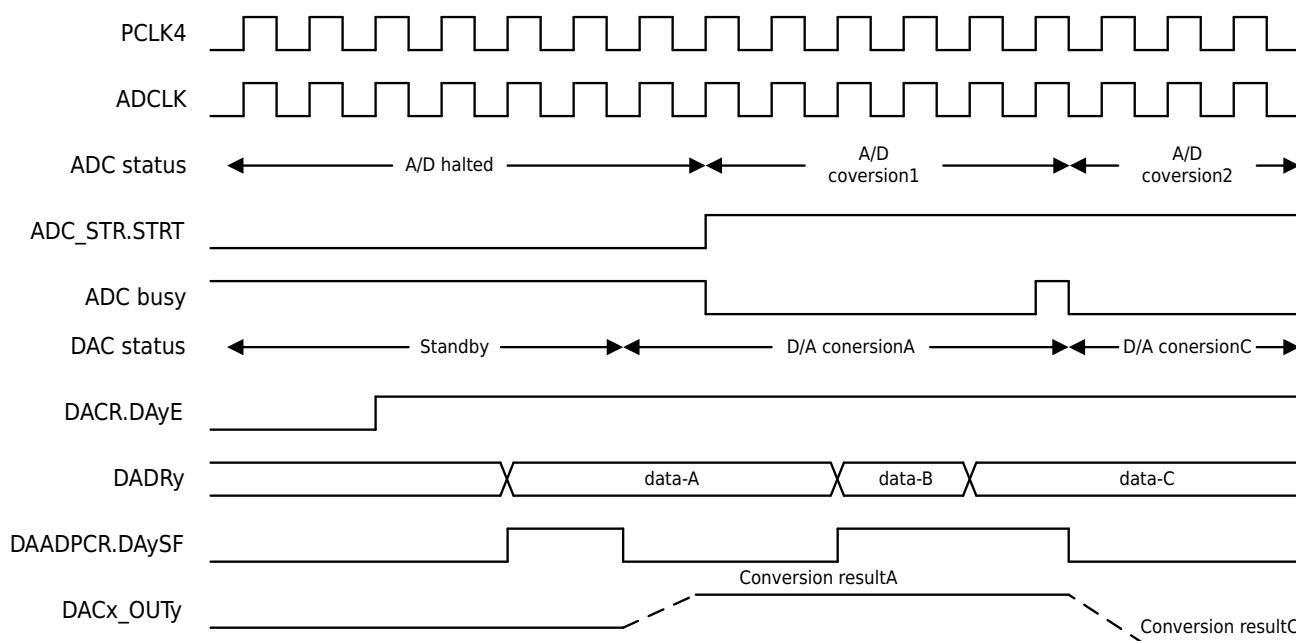


Figure 19-3 ADC Conversion Priority Mode Action Schematic Diagram

**Note**

When converting external data (DAC\_DACR.EXTDSLy=1), ADC conversion priority mode is disabled.

The following takes DAC1 channel 1 as an example to introduce the steps for setting the ADC conversion priority mode.

1. Confirm ADC is stopped and set DAC\_DAADPCR.ADCSL1.
2. Confirm ADC is stopped and set DAC\_DACR.DA1E to 1.
3. Write conversion data to DAC\_DADR1.
  - When conversion data A is written to DAC\_DADR1, ADC\_STR.STRT=0 (ADC is stopped), and DAC conversion starts after 2 ADCLK periods.
  - When conversion data B is written to DAC\_DADR1, ADC\_STR.STRT=1 (ADC is converting), and DAC conversion starts after ADC conversion is complete. Conversion data C is written to DAC\_DADR1 before DAC conversion starts, so the final converted data is C, not B.

To avoid data loss, check DAC\_DAADPCR.DAySF register to confirm the current data conversion is complete before rewriting DAC\_DADRx.

## 19.4 Register Description

### 19.4.1 Register List

- DAC1 base address: 0x40041000
- DAC2 base address: 0x40041400

Table 19-2 Digital-to-Analog Converter (DAC) Register Offset

| Offset | Register    | Description                              | Width |
|--------|-------------|------------------------------------------|-------|
| 0x00   | DAC_DADR1   | Data Register 1                          | 16    |
| 0x02   | DAC_DADR2   | Data Register 2                          | 16    |
| 0x04   | DAC_DACR    | Control Register                         | 16    |
| 0x06   | DAC_DAADPCR | ADC Conversion Priority Control Register | 16    |
| 0x1C   | DAC_DAOCR   | Analog Output Control Register           | 16    |

### 19.4.2 Data Register (DAC\_DADRx) (y=1-2)

This register stores the data for DAC conversion, with one register per channel. As long as DAC conversion is enabled, the value in DAC\_DADRx is converted to an analog voltage and output from the analog port. 32-bit operations on DAC\_DADR1 can set the conversion data for both channels in the unit simultaneously.

Offset: 0x00 (DAC\_DADR1), 0x02 (DAC\_DADR2)

#### Data Right-aligned (DACR.DPSEL=0)

| Offset | Bit Position |    |    |    |      |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|------|----|---|---|---|---|---|---|---|---|---|---|
| -      | 15           | 14 | 13 | 12 | 11   | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x0000       |    |    |    |      |    |   |   |   |   |   |   |   |   |   |   |
| Name   | Reserved     |    |    |    | DADR |    |   |   |   |   |   |   |   |   |   |   |
| Access |              |    |    |    |      |    |   |   |   |   |   |   |   |   |   |   |

| Bit   | Name     | Function            | Description                       | Access |
|-------|----------|---------------------|-----------------------------------|--------|
| 15:12 | Reserved | Reserved            | -                                 | -      |
| 11:0  | DADR     | Transformation Data | Transformation data setting value | RW     |

#### Data Left-aligned (DACR.DPSEL=1)

| Offset | Bit Position |    |    |    |    |    |   |   |   |   |   |   |          |   |   |   |
|--------|--------------|----|----|----|----|----|---|---|---|---|---|---|----------|---|---|---|
| -      | 15           | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3        | 2 | 1 | 0 |
| Reset  | 0x0000       |    |    |    |    |    |   |   |   |   |   |   |          |   |   |   |
| Name   | DADR         |    |    |    |    |    |   |   |   |   |   |   | Reserved |   |   |   |
| Access |              |    |    |    |    |    |   |   |   |   |   |   |          |   |   |   |

| Bit  | Name     | Function            | Description                       | Access |
|------|----------|---------------------|-----------------------------------|--------|
| 15:4 | DADR     | Transformation Data | Transformation data setting value | RW     |
| 3:0  | Reserved | Reserved            | -                                 | -      |

## 19.4.3 Control Register (DAC\_DACR)

| Offset | Bit Position |    |    |         |         |        |        |       |          |   |   |   |      |      |     |   |
|--------|--------------|----|----|---------|---------|--------|--------|-------|----------|---|---|---|------|------|-----|---|
| 0x04   | 15           | 14 | 13 | 12      | 11      | 10     | 9      | 8     | 7        | 6 | 5 | 4 | 3    | 2    | 1   | 0 |
| Reset  | 0x0000       |    |    |         |         |        |        |       |          |   |   |   |      |      |     |   |
| Name   | Reserved     |    |    | EXTDSL2 | EXTDSL1 | DAAMP2 | DAAMP1 | DPSEL | Reserved |   |   |   | DA2E | DA1E | DAE |   |
| Access |              |    |    | RW      | RW      | RW     | RW     | RW    |          |   |   |   | RW   | RW   | RW  |   |

| Bit   | Name     | Function                           | Description                                                                                                                                                                              | Access |
|-------|----------|------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 15:13 | Reserved | Reserved                           | -                                                                                                                                                                                        | -      |
| 12    | EXTDSL2  | Channel 2 External Data Selection  | <ul style="list-style-type: none"> <li>0b0: Selects DAC_DADR2 as the conversion data for Channel 2.</li> <li>0b1: Selects external data as the conversion data for Channel 2.</li> </ul> | RW     |
| 11    | EXTDSL1  | Channel 1 External Data Selection  | <ul style="list-style-type: none"> <li>0b0: Selects DAC_DADR1 as the conversion data for Channel 1.</li> <li>0b1: Selects external data as the conversion data for Channel 1.</li> </ul> | RW     |
| 10    | DAAMP2   | Channel 2 Output Amplifier Enabled | <ul style="list-style-type: none"> <li>0b0: Disable Channel 2 output amplifier.</li> <li>0b1: Enable Channel 2 output amplifier.</li> </ul>                                              | RW     |
| 9     | DAAMP1   | Channel 1 Output Amplifier Enabled | <ul style="list-style-type: none"> <li>0b0: Disable Channel 1 output amplifier.</li> <li>0b1: Enable Channel 1 output amplifier.</li> </ul>                                              | RW     |
| 8     | DPSEL    | Data Register Format Selection     | <ul style="list-style-type: none"> <li>0b0: Right-aligned format</li> <li>0b1: Left-aligned format</li> </ul>                                                                            | RW     |
| 7:3   | Reserved | Reserved                           | -                                                                                                                                                                                        | -      |
| 2     | DA2E     | Channel 2 Enabled                  | Bit setting valid when DAE=0: <ul style="list-style-type: none"> <li>0b0: Channel 2 disabled</li> <li>0b1: Channel 2 enabled</li> </ul>                                                  | RW     |
| 1     | DA1E     | Channel 1 Enabled                  | Bit setting valid when DAE=0: <ul style="list-style-type: none"> <li>0b0: Channel 1 disabled</li> <li>0b1: Channel 1 enabled</li> </ul>                                                  | RW     |
| 0     | DAE      | Total Channel Enabled              | <ul style="list-style-type: none"> <li>0b0: Channels 1, 2 disabled</li> <li>0b1: Channels 1, 2 enabled</li> </ul>                                                                        | RW     |



**Note**

- This register controls DAC conversion and DAC output; see the "[Analog Output Control Register \(DAC\\_DAOCR\)](#)" description for details.
- When DA1E=0 and DAE=0, channel 1 stops DAC conversion, the analog output is disabled regardless of DAAMP1, and the port presents a high-impedance state. When DA2E=0 and DAE=0, channel 2 stops DAC conversion, the analog output is disabled regardless of DAAMP2, and the port presents a high-impedance state.
- When using ADC conversion priority mode, set DA2E, DA1E, and DAE when ADC is stopped (ADC\_STR.STRT=0). To avoid accidental ADC startup, set the ADC trigger selection to software trigger.

## 19.4.4 Analog Output Control Register (DAC\_DAOCR)

| Offset | Bit Position |         |          |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|---------|----------|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0x06   | 15           | 14      | 13       | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x0000       |         |          |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | DAODIS2      | DAODIS1 | Reserved |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access | RW           | RW      |          |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit  | Name     | Function                      | Description                                                                                                                                     | Access |
|------|----------|-------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 15   | DAODIS2  | DACx_OUT2 Port Output Control | <ul style="list-style-type: none"> <li>0b0: Disable port output of analog voltage</li> <li>0b1: Enable port output of analog voltage</li> </ul> | RW     |
| 14   | DAODIS1  | DACx_OUT1 Port Output Control | <ul style="list-style-type: none"> <li>0b0: Disable port output of analog voltage</li> <li>0b1: Enable port output of analog voltage</li> </ul> | RW     |
| 13:0 | Reserved | Reserved                      | -                                                                                                                                               | -      |

Table 19-11 DAC Conversion and Analog Output Control

| DAE | DAyE | DAAMP | DAODIS | DAC Conversion | Output Amplifier | DACx_OUTy output     | DAO Output           |
|-----|------|-------|--------|----------------|------------------|----------------------|----------------------|
| 0b0 | 0b0  | -     | -      | Stop           | Stop             | High Impedance State | High Impedance State |
|     | 0b1  | 0b0   | 0b0    | Start          | Stop             | Normal Output        | Normal Output        |
|     |      |       | 0b1    | Start          | Stop             | High Impedance State | Normal Output        |
|     |      | 0b1   | 0b0    | Start          | Start            | Amplified Output     | Normal Output        |
|     |      |       | 0b1    | Start          | Start            | High Impedance State | Normal Output        |
| 0b1 | 0b0  | 0b0   | 0b0    | Start          | Stop             | Normal Output        | Normal Output        |
|     |      |       | 0b1    | Start          | Stop             | High Impedance State | Normal Output        |
|     |      | 0b1   | 0b0    | Start          | Start            | Amplified Output     | Normal Output        |
|     |      |       | 0b1    | Start          | Start            | High Impedance State | Normal Output        |
|     | 0b1  | 0b0   | 0b0    | Start          | Stop             | Normal Output        | Normal Output        |
|     |      |       | 0b1    | Start          | Stop             | High Impedance State | Normal Output        |
|     |      | 0b1   | 0b0    | Start          | Start            | Amplified Output     | Normal Output        |
|     |      |       | 0b1    | Start          | Start            | High Impedance State | Normal Output        |

## 19.4.5 ADC Conversion Priority Control Register (DAC\_DAADPCR)

| Offset | Bit Position |          |    |    |    |    |       |       |          |   |   |   |   |        |        |        |
|--------|--------------|----------|----|----|----|----|-------|-------|----------|---|---|---|---|--------|--------|--------|
| 0x1C   | 15           | 14       | 13 | 12 | 11 | 10 | 9     | 8     | 7        | 6 | 5 | 4 | 3 | 2      | 1      | 0      |
| Reset  | 0x0000       |          |    |    |    |    |       |       |          |   |   |   |   |        |        |        |
| Name   | ADPEN        | Reserved |    |    |    |    | DA2SF | DA1SF | Reserved |   |   |   |   | ADCSL3 | ADCSL2 | ADCSL1 |
| Access | RW           |          |    |    |    |    | RO    | RO    |          |   |   |   |   | RW     | RW     | RW     |

| Bit   | Name            | Function                               | Description                                                                                                                                                                                       | Access |
|-------|-----------------|----------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 15    | ADPEN           | ADC Conversion Priority Mode Selection | <ul style="list-style-type: none"> <li>0b0: ADC conversion priority mode disabled</li> <li>0b1: ADC conversion priority mode enabled</li> </ul>                                                   | RW     |
| 14:10 | <i>Reserved</i> | <i>Reserved</i>                        | -                                                                                                                                                                                                 | -      |
| 9     | DA2SF           | Channel 2 Data Update Status           | This flag bit only changes in ADC conversion priority mode: <ul style="list-style-type: none"> <li>0b0: Channel 2 data update complete</li> <li>0b1: Channel 2 data update in progress</li> </ul> | RO     |
| 8     | DA1SF           | Channel 1 Data Update Status           | This flag bit only changes in ADC conversion priority mode: <ul style="list-style-type: none"> <li>0b0: Channel 1 data update complete</li> <li>0b1: Channel 1 data update in progress</li> </ul> | RO     |
| 7:3   | <i>Reserved</i> | <i>Reserved</i>                        | -                                                                                                                                                                                                 | -      |
| 2     | ADCSL3          | ADC_3 Priority Selection               | <ul style="list-style-type: none"> <li>0b0: Does not select ADC_3 as the ADC conversion priority object</li> <li>0b1: Selects ADC_3 as the ADC conversion priority object</li> </ul>              | RW     |
| 1     | ADCSL2          | ADC_2 Priority Selection               | <ul style="list-style-type: none"> <li>0b0: Does not select ADC_2 as the ADC conversion priority object</li> <li>0b1: Select ADC_2 as the ADC conversion priority object</li> </ul>               | RW     |
| 0     | ADCSL1          | ADC_1 Priority Selection               | <ul style="list-style-type: none"> <li>0b0: Does not select ADC_1 as the ADC conversion priority object</li> <li>0b1: Select ADC_1 as the ADC conversion priority object</li> </ul>               | RW     |

**Note**

- This register controls ADC conversion priority mode.
- When ADPEN is set to 1, ADC conversion priority mode is enable; select the ADC unit for priority when setting ADPEN.
- DA1SF and DA2SF are DA conversion status flag bits, reflecting whether the data in the current channel data register has been converted.

## 19.5 Precautions

### 19.5.1 Module Stop Function Settings

The module stop control register can set the DAC to be active or stopped. After system reset, the DAC is stopped and must be released from the module stop state before accessing the register. For specific descriptions, please refer to the relevant sections of the "Action Mode and Low Power Mode" section in the "Power Supply Control (PWC) " chapter.

### 19.5.2 DAC Action in Module Stop State

If the DAC enters the module stop state during conversion, the analog output is maintained, and the current through the analog power supply remains the same as during conversion. To further reduce power consumption in the module stop state, set both DAYE and DAE bits in DAC\_DACR register to 0.

### 19.5.3 DAC Action in Stop Low Power Mode

If the system enters Stop mode during DAC conversion, the analog output is maintained, and the current through the analog power supply remains the same as during conversion. To further reduce power consumption in Stop mode, set both DAYE and DAE bits in DAC\_DACR register to 0.

### 19.5.4 DAC Action in Power-Down Low Power Mode

If the system enters Power-down mode during DAC conversion, the analog output is set to a high-impedance state.

### 19.5.5 Using the Output Amplifier

Use the following initialization procedure when using the output amplifier.

**Step1.** Write all 0s to DAC\_DADRY.

**Step2.** Set DAC\_DACR.DAAMPY to 1.

**Step3.** Set DAC\_DACR.DAE or DAYE to 1.

**Step4.** Wait 3 $\mu$ s before writing the conversion data to DAC\_DADRY.

Turning off the DAC stops the amplifier. To use the amplifier again, repeat steps 1-4.

## 20 On-chip Temperature Sensor (OTS)

### 20.1 Introduction

The On-chip Temperature Sensor (hereinafter referred to as OTS) can obtain the temperature inside the chip to support the reliable operation of the system. OTS provides a set of temperature-related digital quantities, and the temperature value can be obtained through calculation. When not in use, it can be turned off by the module stop function to reduce system power consumption.

### 20.2 Functional Description

#### 20.2.1 Function Block Diagram

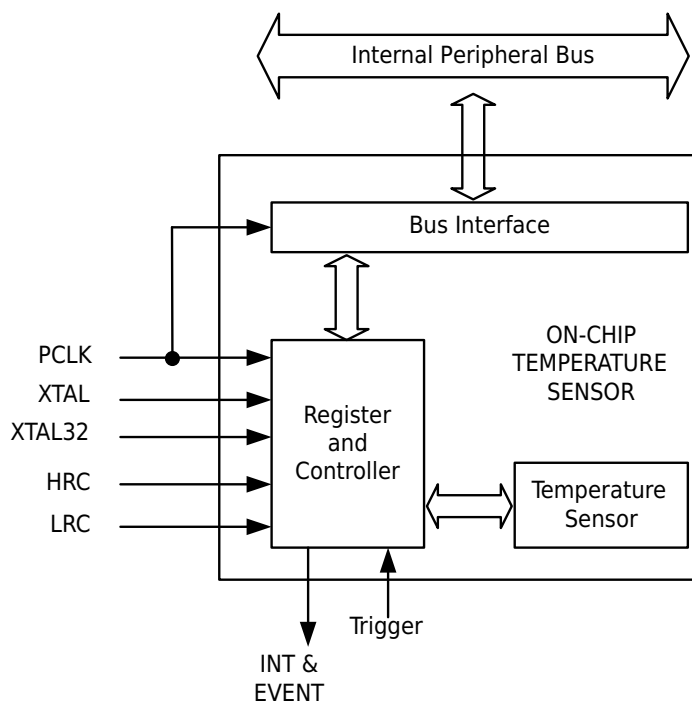


Figure 20-1 OTS Function Block Diagram

#### 20.2.2 Usage Method

Before using OTS to obtain the internal temperature of the chip, it must be calibrated first, and the measurement accuracy is related to the calibration method. Please refer to the following calibration experiment instructions for calibration. When using, please turn off the module stop function and start the internal low-speed clock LRC. At the same time, please choose to start the internal high-speed clock HRC, external high-speed clock XTAL and external low-speed clock XTAL32 according to the usage. After setting the register OTS\_CTL.OTSST to "1", the temperature measurement starts, and the OTSST bit will be automatically cleared after the temperature measurement is completed. After confirming that OTSST is "0", read the temperature parameters in registers OTS\_DR1, OTS\_DR2, and OTS\_ECR, and use the following calculation formula to obtain the temperature value.

$$T = K \times (1.7/D1 - 1/D2) \times E_{hrc} + M$$

Parameter Description:

- T: Temperature (°C)
- K: Temperature slope (determined by calibration experiment)

- D1: Temperature data 1 (read from register OTS\_DR1)
- D2: Temperature data 2 (read from register OTS\_DR2)
- $E_{hrc}$ : HRC frequency error compensation data (read from register OTS\_ECR)
- M: Temperature offset (determined by calibration experiment)

The register OTS\_CTL.OTSCK is used to select the temperature measurement clock. When the HRC action is selected, its frequency error may affect the accuracy of the final calculated temperature. To eliminate this error, start XTAL32 before measuring the temperature and use  $E_{hrc}$  when calibrating and calculating the temperature. When selecting XTAL action, ignore  $E_{hrc}$  and show it as a constant 1.

**Calibration Experiment:** Perform a calibration experiment at two determined temperatures and calculate K and M.

$$K = (T_2 - T_1) / (A_2 - A_1)$$

$$M = T_1 - K \times A_1 = T_2 - K \times A_2$$

Parameter Description:

- T1: Experimental temperature 1
- T2: Experimental temperature 2
- A1:  $(1.7/D1_{T1} - 1/D2_{T1}) \times E_{hrcT1}$   
 $D1_{T1}$ ,  $D1_{T1}$ ,  $E_{hrcT1}$  are D1, D2,  $E_{hrc}$  measured at temperature T1 respectively.
- A2:  $(1.7/D1_{T2} - 1/D2_{T2}) \times E_{hrcT2}$   
 $D1_{T2}$ ,  $D1_{T2}$ ,  $E_{hrcT2}$  are D1, D2,  $E_{hrc}$  measured at temperature T2 respectively.

For reference, three sets of temperature data are preset inside the chip, and the data of two temperature points can be arbitrarily selected for calibration. Since the test environment temperature may vary, the preset data is for reference only.

Table 20-1 OTS Preset Temperature Data

| Testing Temperature       | Reading Method          |                         | Testing Conditions |
|---------------------------|-------------------------|-------------------------|--------------------|
|                           | Temperature Data 1 (D1) | Temperature Data 2 (D2) |                    |
| $T_A = -40^\circ\text{C}$ | OTS_PDR3[15:0]          | OTS_PDR3[31:16]         | XTAL=8MHz          |
| $T_A = 25^\circ\text{C}$  | OTS_PDR1[15:0]          | OTS_PDR1[31:16]         | XTAL=8MHz          |
| $T_A = 125^\circ\text{C}$ | OTS_PDR2[15:0]          | OTS_PDR2[31:16]         | XTAL=8MHz          |

$E_{hrc}$  is a calibration value to eliminate the influence of HRC error on the measured temperature. It can be read from register OTS\_ECR. The usage is as follows:

Table 20-2  $E_{hrc}$  Usage and Setting Method

| Action Clock | Calibration Experiment (Calculate K, M) |                                   | Temperature Measurement           |
|--------------|-----------------------------------------|-----------------------------------|-----------------------------------|
| (OTSCK)      | Use Preset Temperature Data             | Re-acquire Temperature Data       | (Calculate T)                     |
| XTAL         | $E_{hrc} = 1$                           | $E_{hrc} = 1$                     | $E_{hrc} = 1$                     |
| HRC          | $E_{hrc} = f_{hrc}(MHz) / 0.032768$     | $E_{hrc} = \text{OTS\_ECR}[15:0]$ | $E_{hrc} = \text{OTS\_ECR}[15:0]$ |

**Note**

$f_{\text{hrc}}$  is the standard frequency of HRC, that is, 16MHz or 20MHz.

The register OTS\_CTL.TSSTP is used to select whether to turn off the analog temperature sensor after the temperature measurement is completed. The initial value of TSSTP is "0", which means that the analog temperature sensor is turned on after a temperature measurement is completed, so that the stabilization time of the analog temperature sensor from off to on will be automatically skipped in the next temperature measurement. To turn off the analog temperature sensor after each temperature measurement, please set TSSTP to "1".

The temperature measurement can be triggered by other of peripheral events. When using it, please set the trigger target of the trigger source to OTS. When the temperature measurement is completed, an event that triggers the start of other of peripherals or generated. When using it, please set the register OTS\_TRG to select the trigger target. Please set the register OTS\_CTL.OTSIE to "1" when using the temperature measurement completion interrupt.

## 20.3 Register Description

### 20.3.1 Register List

OTS Preset Data Register Base Address: 0x40010600

Table 20-3 OTS Preset Data Register Offset

| Offset | Register                 | Description                            | Width |
|--------|--------------------------|----------------------------------------|-------|
| 0xE0   | <a href="#">OTS_PDR1</a> | OTS Preset Temperature Data Register 1 | 32    |
| 0xF4   | <a href="#">OTS_PDR2</a> | OTS Preset Temperature Data Register 2 | 32    |
| 0xF8   | <a href="#">OTS_PDR3</a> | OTS Preset Temperature Data Register 3 | 32    |

OTS Base Address: 0x4004A800

Table 20-4 On-chip Temperature Sensor (OTS) Register Offset

| Offset | Register                | Description                     | Width |
|--------|-------------------------|---------------------------------|-------|
| 0x00   | <a href="#">OTS_CTL</a> | OTS Control Register            | 16    |
| 0x02   | <a href="#">OTS_DR1</a> | OTS Data Register 1             | 16    |
| 0x04   | <a href="#">OTS_DR2</a> | OTS Data Register 2             | 16    |
| 0x06   | <a href="#">OTS_ECR</a> | OTS Error Compensation Register | 16    |



## 20.3.2 OTS Control Register (OTS\_CTL)

| Offset | Bit Position |    |    |    |    |    |   |   |   |   |   |   |       |       |       |       |
|--------|--------------|----|----|----|----|----|---|---|---|---|---|---|-------|-------|-------|-------|
| 0x00   | 15           | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3     | 2     | 1     | 0     |
| Reset  | 0x0000       |    |    |    |    |    |   |   |   |   |   |   |       |       |       |       |
| Name   | Reserved     |    |    |    |    |    |   |   |   |   |   |   | TSSTP | OTSIE | OTSCK | OTSST |
| Access |              |    |    |    |    |    |   |   |   |   |   |   | RW    | RW    | RW    | RW    |

| Bit  | Name     | Function                           | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | Access |
|------|----------|------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 15:4 | Reserved | Reserved                           | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | -      |
| 3    | TSSTP    | Analog Temperature Sensor Disabled | Select whether to automatically turn off the analog temperature sensor after the temperature measurement is completed <ul style="list-style-type: none"> <li>0b0: Analog temperature sensor enabled</li> <li>0b1: Analog temperature sensor disabled</li> </ul>                                                                                                                                                                                                                                                                                                                                                           | RW     |
| 2    | OTSIE    | Interrupt Enabled Bit              | <ul style="list-style-type: none"> <li>0b0: Disable temperature measurement end interrupt request</li> <li>0b1: Enable temperature measurement end interrupt request</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                           | RW     |
| 1    | OTSCK    | Clock Select Bit                   | <ul style="list-style-type: none"> <li>0b0: Select external high-speed clock (XTAL) action</li> <li>0b1: Select internal high-speed clock (HRC) action</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                         | RW     |
| 0    | OTSST    | Temperature Measurement Start Bit  | <ul style="list-style-type: none"> <li>0b0: Stop temperature measurement</li> <li>0b1: Start temperature measurement</li> </ul> <div style="border: 1px solid black; padding: 5px; margin-top: 10px;"> <p> <b>Note</b></p> <p>Set "1" condition:</p> <ol style="list-style-type: none"> <li>Software set to "1"</li> <li>Hardware trigger set to "1"</li> </ol> <p>Clear "0" condition:</p> <ol style="list-style-type: none"> <li>Software clear "0"</li> <li>Automatically clear "0" after temperature measurement</li> </ol> </div> | RW     |

## 20.3.3 OTS Data Register 1 (OTS\_DR1)

| Offset | Bit Position |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0x02   | 15           | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x0000       |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | TSDC         |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access | RO           |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit  | Name | Function            | Description                                                                                                                                                                                                                                                                         | Access |
|------|------|---------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 15:0 | TSDC | Temperature Data D1 | Temperature data D1, automatically updated after temperature measurement is completed<br><div>  <b>Note</b><br/>           Please confirm that OTS_CTL.OTSST is "0" before reading.         </div> | RO     |

## 20.3.4 OTS Data Register 2 (OTS\_DR2)

| Offset | Bit Position |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0x04   | 15           | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x0000       |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | TSDC         |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access | RO           |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit  | Name | Function            | Description                                                                                                                                                                                                                                                                           | Access |
|------|------|---------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 15:0 | TSDC | Temperature Data D2 | Temperature data D2, automatically updated after temperature measurement is completed<br><div>  <b>Note</b><br/>           Please confirm that OTS_CTL.OTSST is "0" before reading.         </div> | RO     |

### 20.3.5 OTS Error Compensation Register (OTS\_ECR)

| Offset | Bit Position |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0x06   | 15           | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x0000       |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | TSEC         |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access | RO           |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit  | Name | Function     | Description                                                                                                                                                                                                                                                               | Access |
|------|------|--------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 15:0 | TSEC | Error Factor | Error factor $E_{hrc}$ , automatically updated after temperature measurement is completed<br><br><div>  <b>Note</b><br/> Please confirm that OTS_CTL.OTSST is "0" before reading. </div> | RO     |

### 20.3.6 OTS Preset Temperature Data Register 1 (OTS\_PDR1)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |       |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0xE0   | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15    | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | -            |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |       |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | TSPD2        |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | TSPD1 |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access | RO           |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RO    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit   | Name  | Function                   | Description                                                                                                                                                                                                                                                                                                        | Access |
|-------|-------|----------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:16 | TSPD2 | Preset Temperature Data D2 | Preset temperature data for use during calibration<br>This data is measured at 8MHz (test temperature 25°C). Please convert it with the actual frequency of the action clock selected by OTSCK before use.<br>$D2 = TSPD2 \times f_{OTSCK} / 8$ , $f_{OTSCK}$ is the frequency of the actual action clock, in MHz. | RO     |
| 15:0  | TSPD1 | Preset Temperature Data D1 | Preset temperature data for use during calibration<br>This data is measured at 8MHz (test temperature 25°C). Please convert it with the actual frequency of the action clock selected by OTSCK before use.<br>$D1 = TSPD1 \times f_{OTSCK} / 8$ , $f_{OTSCK}$ is the frequency of the actual action clock, in MHz. | RO     |

## 20.3.7 OTS Preset Temperature Data Register 2 (OTS\_PDR2)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |       |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0xF4   | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15    | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | -            |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |       |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | TSPD2        |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | TSPD1 |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access | RO           |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RO    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit   | Name  | Function                   | Description                                                                                                                                                                                                                                                                                                         | Access |
|-------|-------|----------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:16 | TSPD2 | Preset Temperature Data D2 | Preset temperature data for use during calibration<br>This data is measured at 8MHz (test temperature 125°C). Please convert it with the actual frequency of the action clock selected by OTSCK before use.<br>$D2 = TSPD2 \times f_{OTSCK} / 8$ , $f_{OTSCK}$ is the frequency of the actual action clock, in MHz. | RO     |
| 15:0  | TSPD1 | Preset Temperature Data D1 | Preset temperature data for use during calibration<br>This data is measured at 8MHz (test temperature 125°C). Please convert it with the actual frequency of the action clock selected by OTSCK before use.<br>$D1 = TSPD1 \times f_{OTSCK} / 8$ , $f_{OTSCK}$ is the frequency of the actual action clock, in MHz. | RO     |

## 20.3.8 OTS Preset Temperature Data Register 3 (OTS\_PDR3)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |       |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0xF8   | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15    | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | -            |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |       |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | TSPD2        |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | TSPD1 |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access | RO           |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RO    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit   | Name  | Function                   | Description                                                                                                                                                                                                                                                                                                         | Access |
|-------|-------|----------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:16 | TSPD2 | Preset Temperature Data D2 | Preset temperature data for use during calibration<br>This data is measured at 8MHz (test temperature -40°C). Please convert it with the actual frequency of the action clock selected by OTSCK before use.<br>$D2 = TSPD2 \times f_{OTSCK} / 8$ , $f_{OTSCK}$ is the frequency of the actual action clock, in MHz. | RO     |
| 15:0  | TSPD1 | Preset Temperature Data D1 | Preset temperature data for use during calibration<br>This data is measured at 8MHz (test temperature -40°C). Please convert it with the actual frequency of the action clock selected by OTSCK before use.<br>$D1 = TSPD1 \times f_{OTSCK} / 8$ , $f_{OTSCK}$ is the frequency of the actual action clock, in MHz. | RO     |

## 21 Advanced Control Timer (Timer6)

### 21.1 Introduction

Advanced Control Timer 6 (Timer6) is a high-performance timer with 32/16-bit counting width, offering a rich and flexible combination of configurations suitable for various complex application scenarios. It supports multiple interrupts, events, and PWM outputs. This timer supports both sawtooth and triangular waveform modes and can generate various types of PWM signals (edge-aligned independent PWM, center-aligned independent PWM, center-aligned complementary PWM, and center-aligned asymmetric PWM, etc.). It enables both software and hardware synchronization between units (synchronized start, stop, clear, refresh, etc.). All reference registers support buffering (single-level and double-level buffering). Timer6 also supports pulse width and period measurement, 2-phase and 3-phase quadrature encoding count, and EMB control, units 1 through 5 and unit 7 support PTO (Pulse Train Output). This series of products is equipped with 8 units of Timer6 (unit 1 to unit 4 are 32-bit timers; unit 5 to unit 8 are 16-bit timers).

### 21.2 Features

The following table outlines the basic functions and features of Timer6.

**Table 21-1 Basic Functions and Features of Timer6**

|                  |                                   |
|------------------|-----------------------------------|
| Waveform Mode    | Sawtooth Wave (count up and down) |
|                  | Triangle Wave (count up and down) |
| Basic Function   | Capture Input                     |
|                  | Software Synchronization          |
|                  | Hardware Synchronization          |
|                  | Buffer Function                   |
|                  | Pulse Width Measurement           |
|                  | Period Measurement                |
|                  | Quadrature Encoding Count         |
|                  | General PWM output                |
|                  | EMB Control                       |
|                  | PTO Output                        |
| Interrupt Output | Count Compare Match Interrupt     |
|                  | Count Period Match Interrupt      |
|                  | Dead Zone Error Interrupt         |
| Event Output     | Count Compare Match Event         |
|                  | Count Period Match Event          |

## 21.3 Functional Description

### 21.3.1 Timer6 Block Diagram

The basic block diagram of Timer6 is shown below. In the diagram, the symbol "<t>" denotes the unit number, which ranges from 1 to 8. In the remainder of this section, "<t>" will consistently refer to the unit number and will not be further explained.

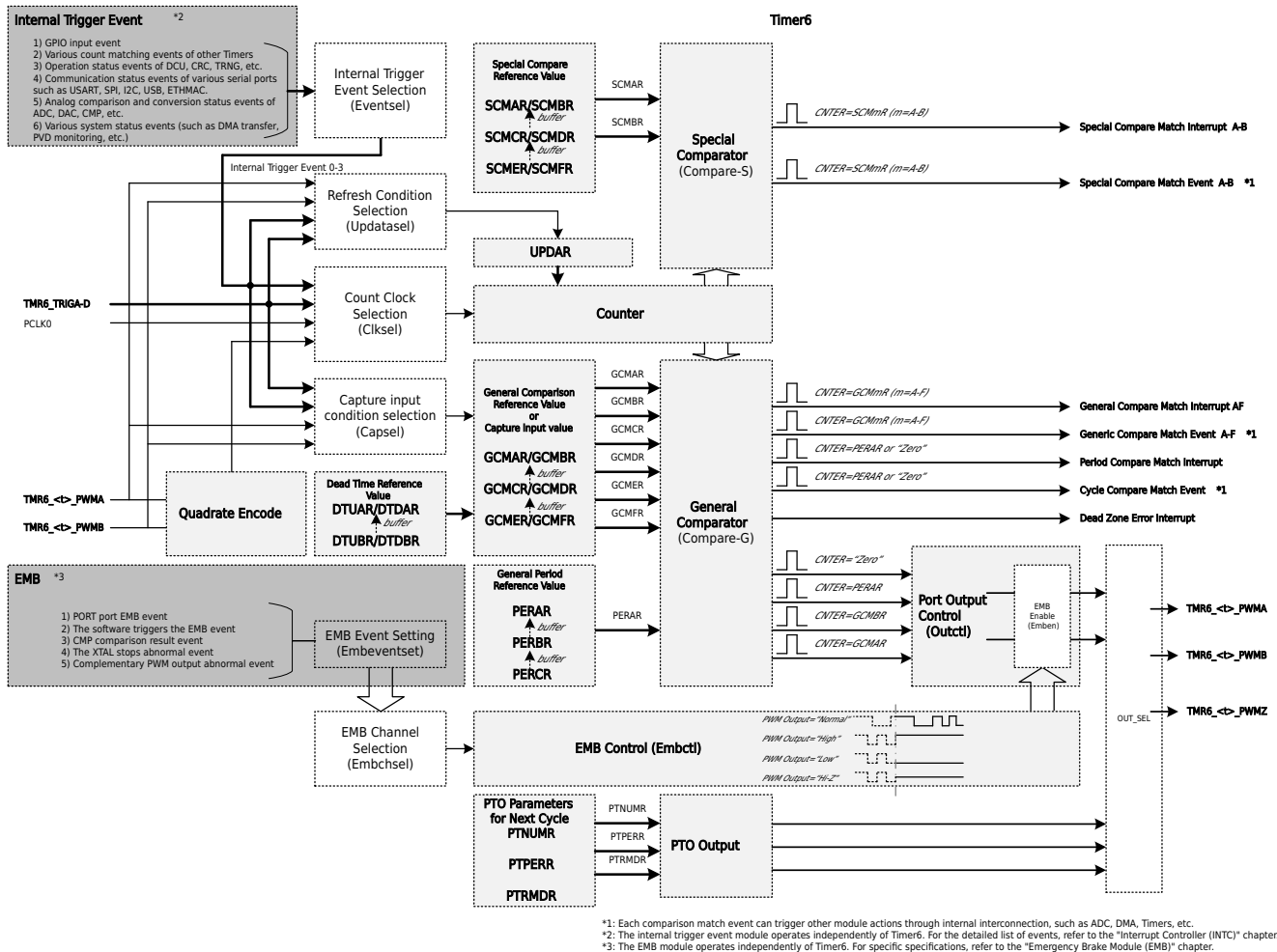


Figure 21-1 Timer6 Basic Block Diagram

Table 21-2 : Timer6 Port List shows the input and output port list of Timer6.

Table 21-2 Timer6 Port List

| Port Name     | Direction    | Function                                                                                                                                                                                                               |
|---------------|--------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| TMR6_<t>_PWMA | Input/Output | <ul style="list-style-type: none"> <li>Quadrature encoder count clock input, capture input, compare output, or PTO A/B phase output port</li> <li>Hardware start, stop, clear, refresh condition input port</li> </ul> |
| TMR6_<t>_PWMB |              |                                                                                                                                                                                                                        |
| TMR6_TRIGA    | Input/Output | <ul style="list-style-type: none"> <li>Hardware count clock input port or capture input port</li> <li>Hardware start, stop, clear, refresh condition input port</li> <li>PTO Z phase output port</li> </ul>            |
| TMR6_TRIGB    |              |                                                                                                                                                                                                                        |
| TMR6_TRIGC    |              |                                                                                                                                                                                                                        |
| TMR6_TRIGD    |              |                                                                                                                                                                                                                        |

## 21.3.2 Basic Actions

### 21.3.2.1 Waveform Mode

Timer6 has two basic counting wave modes: Sawtooth wave mode and Triangle wave mode. The basic waveforms of the two waveform modes are shown below.

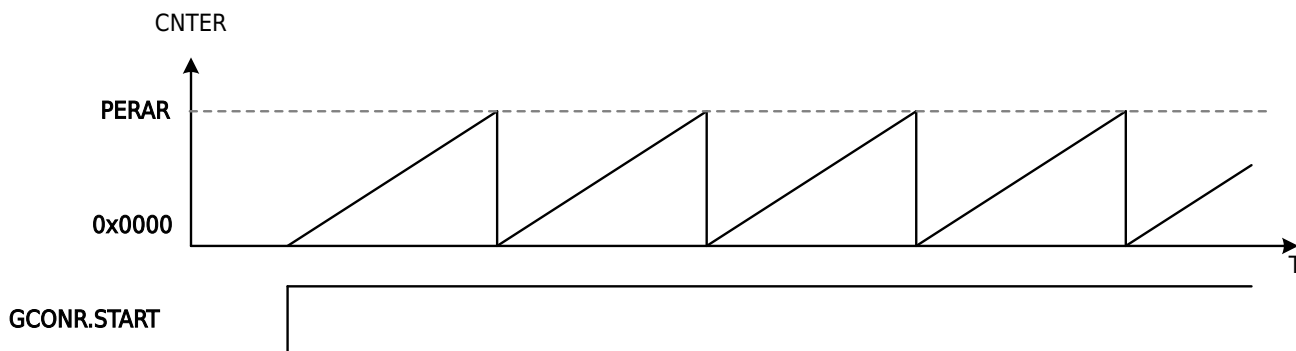


Figure 21-2 Sawtooth Waveform (Counting Up)

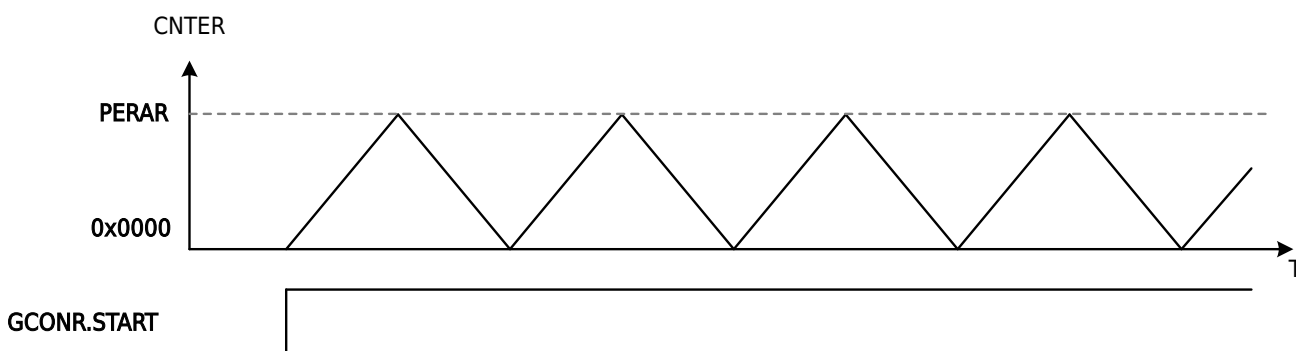


Figure 21-3 Triangle Waveform

### 21.3.2.2 Compare Output

Each unit's Timer6 has two compare output ports (TMR6\_<t>PWMA and TMR6\_<t>PWMB), which can output specified levels when the counter value matches the compare reference. The GCMAR and GCMBR registers correspond to the compare references for TMR6\_<t>PWMA and TMR6\_<t>PWMB, respectively. When the timer counter value equals GCMAR, the TMR6\_<t>PWMA or TMR6\_<t>PWMB port outputs the specified level.

The levels of the TMR6\_<t>PWMA port at the start of counting, stop of counting, counter overflow, and compare match can be set via the PCNAR.STACA, PCNAR.STPCA, PCNAR.OVFCA, PCNAR.UDFCA, PCNAR.CMA<B>U<D>A, and PCNAR.FORCA bits of the port control register (PCNAR).

The levels of the TMR6\_<t>PWMB port at the start of counting, stop of counting, counter overflow, and compare match can be set via the PCNBR.STACB, PCNBR.STPCB, PCNBR.OVFCEB, PCNBR.UDFCB, PCNBR.CMA<B>U<D>B, and PCNBR.FORCB bits of the port control register (PCNBR). The figure below illustrates an example of the compare output behavior.



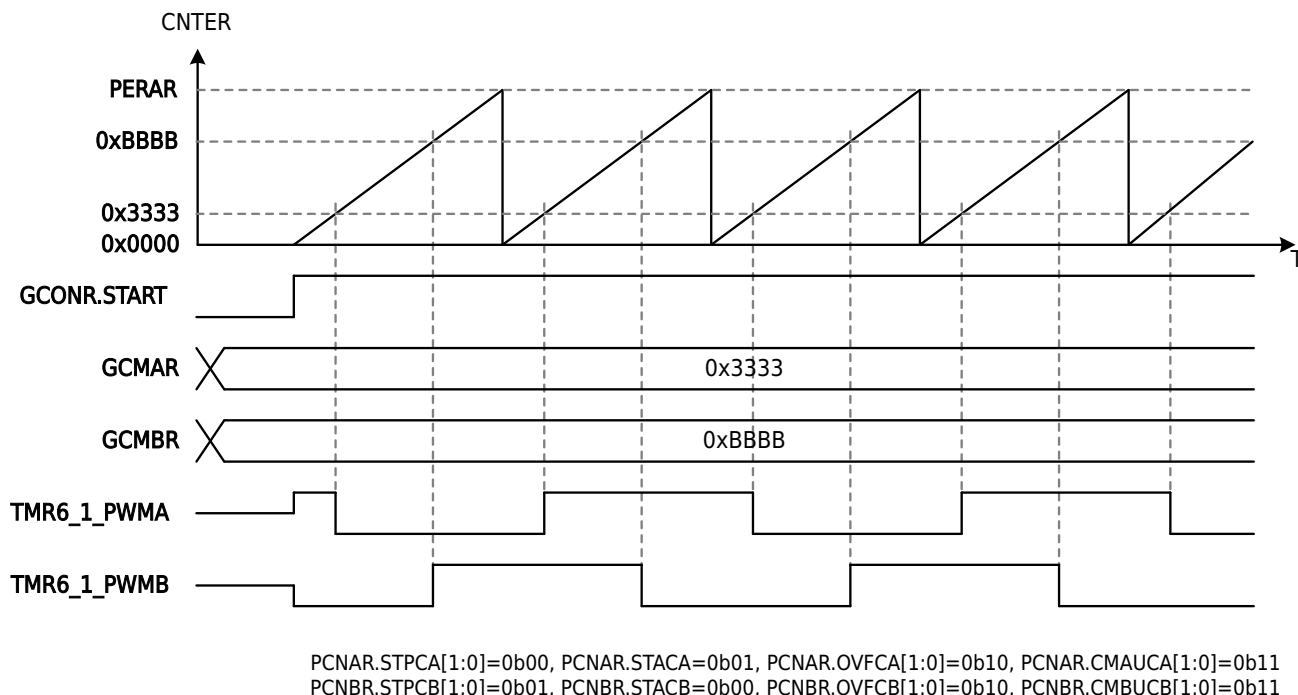


Figure 21-4 Compare Output Action

### 21.3.2.3 Capture Input

Each unit features a capture input function and includes two sets of capture registers (GCMAR, GCMBR) for storing the current counter value upon capture. By setting the PCNA<B>.CAPMDA<B> bit to 1, the capture input function is enabled. When the specified capture condition is met and becomes active, the current counter value is stored in the corresponding capture register (GCMAR, GCMBR).

The conditions for each set of capture inputs of each unit can be internal trigger event inputs 0-3, port inputs of TMR6\_TRIGA-D, port inputs of TMR6\_<t>\_PWMA or TMR6\_<t>\_PWMB, and TMR6\_<t>\_XOR (port input XOR of TMR6\_<t>\_PWMA and TMR6\_<t>\_PWMB), etc. The specific condition selection can be set through the hardware capture event selection registers (HCPAR, HCPBR). The following figure shows an example of the capture input action.

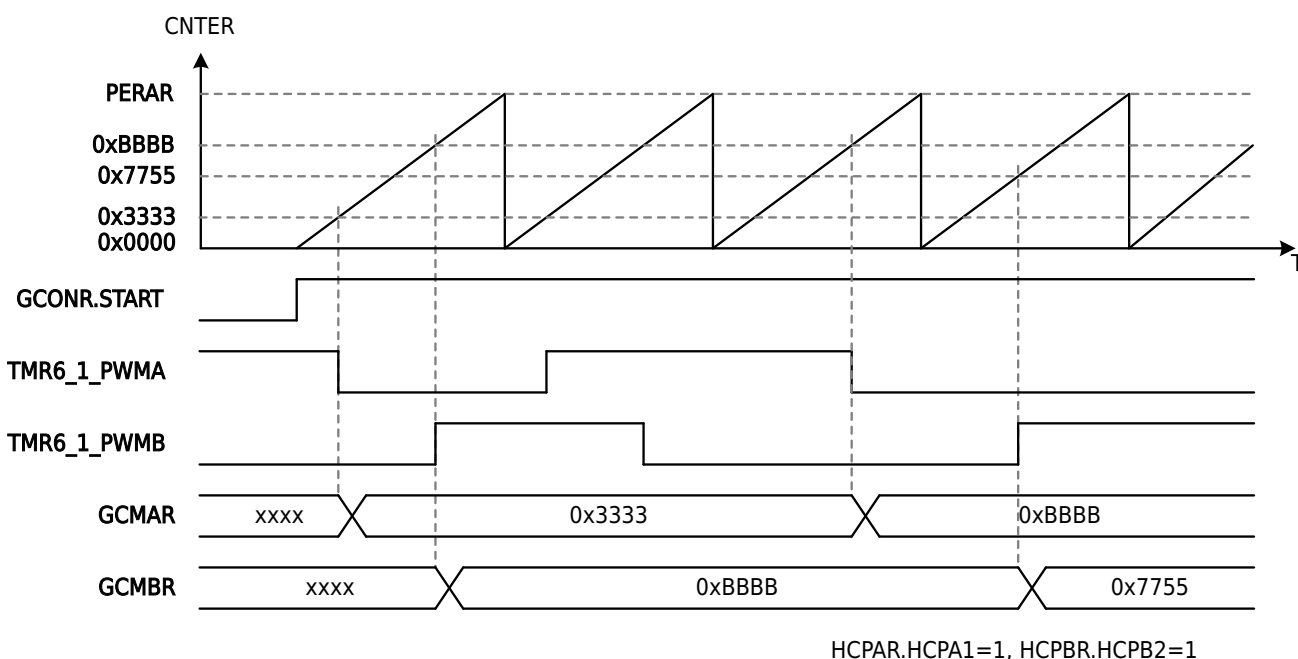


Figure 21-5 Capture Input Action

### 21.3.3 Clock Source Selection

The Timer6 count clock can be selected from the following sources:

1. Divided PCLK0 by 1, 2, 4, 8, 16, 32, 64, 128, 256, 512, or 1024 (configured via GCONR.CKDIV)
2. Internal trigger event inputs 0-3 (configured via HCUPR.HCUPn or HCDOR.HCDOn) (n=8-11)
3. Port inputs of TMR6\_TRIGA-D (configured via HCUPR.HCUPn or HCDOR.HCDOn) (n=16-23)
4. Quadrature encoder input of TMR6\_<t>PWMA and TMR6<t>\_PWMB (configured via HCUPR.HCUPn or HCDOR.HCDOn) (n=0-7)

When the clock source selection is set to 1, the timer operates in software counting mode. When set to others, the timer operates in hardware counting mode. Each count clock source is independent and can be enabled or disabled individually. When the hardware clocks are selected, the software clock is automatically disabled.



#### Note

In hardware count mode, the GCONR.CKDIV must be set to 0.

### 21.3.4 Count Direction

The counting direction of Timer6 can be changed via software. The method for changing the count direction varies slightly depending on the waveform mode.

#### 21.3.4.1 Sawtooth Wave Count Direction

In sawtooth wave mode, the counting direction can be configured either during timer operation or while the timer is stopped.

While counting up, setting GCONR.DIR=0b0 (count down) causes the timer to switch to down-counting mode after reaching an overflow. While counting down, setting GCONR.DIR=0b1 (count up) causes the timer to switch to up-counting mode after reaching an underflow.

When the timer is stopped, the GCONR.DIR setting can be applied immediately by also setting GCONR.FODIR=0b1. The setting of GCONR.DIR can be reflected in the counting direction immediately. After starting the counting, count according to the set GCONR.DIR. If GCONR.FODIR=0b0 is set together with GCONR.DIR, the setting of GCONR.DIR is reflected in the count from the start of the count until it overflows or underflows.

#### 21.3.4.2 Triangular Wave Count Direction

In triangle wave mode, setting the counting direction during operation is disabled. The timer automatically reverses the counting direction upon reaching the peak or valley of the count.

When the timer is stopped, setting the GCONR.DIR bit along with GCONR.FODIR=0b1, the setting of GCONR.DIR can be reflected in the counting direction immediately. After starting the counting, count according to the set GCONR.DIR. If GCONR.DIR is set with GCONR.FODIR=0b0, after the counting starts until the overflow or underflow occurs, the counting direction is automatically changed, and setting the GCONR.DIR bit is invalid.

### 21.3.5 Counter Refresh

Timer6 includes a General Refresh Value Register (UPDAR), which allows real-time refreshing of the General Counter Value Register (COUNTER) value either while the timer is stopped or during counting.

The conditions for refreshing the counter value are configured through the Hardware Refresh Event Selection Register (HUPDR) or the Software Synchronous Refresh Control Register (SUPDR). When a con-

figured refresh event is triggered, the value in the General Counter Value Register (CNTER) is updated with the value specified in the General Refresh Value Register (UPDAR). The figure below illustrates an example of hardware refresh action during counting in sawtooth wave mode.

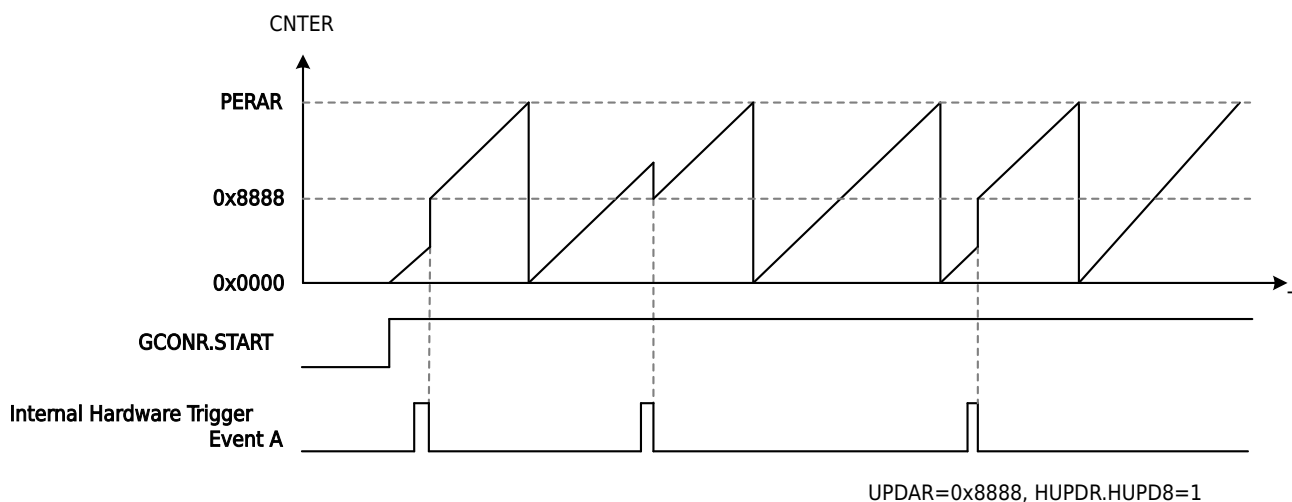


Figure 21-6 Hardware Refresh Action

### 21.3.6 Digital Filtering

The TMR6\_<t>PWMA, TMR6\_<t>PWMB, and TMR6\_TRIGA-D port inputs are equipped with digital filtering functionality. The filter for the TMR6\_<t>PWMA and TMR6\_<t>PWMB ports can be enabled by setting the corresponding enable bits in the Filter Control Register (FCNCR). The filter reference clock used when the filter is active can also be configured via FCNCR. The TMR6\_TRIGA-D ports are shared among multiple units and their filter functionality can be enabled by setting the appropriate enable bits in the Common Filter Control Register (FCNTR). The filter reference clock used when the filter is active can also be configured via FCNTR.

When three consecutive identical levels are sampled on the port based on the filter sampling reference clock, the level is considered active and passed to the internal module. The fewer than three consecutive identical levels are treated as external noise and ignored and not be passed to the internal module. An example of this filtering behavior is shown in the figure below.

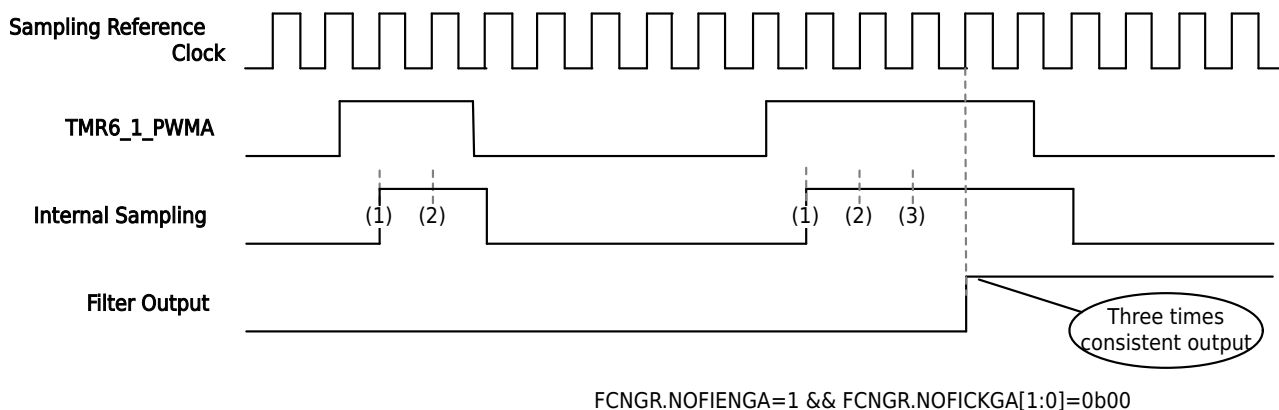


Figure 21-7 Digital Filtering Function of Capture Input Ports

## 21.3.7 Software Synchronization

### Software Synchronization Start

Each unit can achieve synchronous start of the target unit counter (COUNTER) by setting the relevant bits of the Software Synchronous Start Control Register (SSTAR).

In addition, the product is equipped with a count synchronization enable register TMR\_SYNENRx (x=1-2) to synchronously start all the timers (including Timer0, Timer2, Timer4, Timer6, TimerA) carried by the product. For usage details and register descriptions, refer to the General Timer (Timer0) chapter.

### Software Synchronization Stop

By setting the relevant bits in the Software Synchronization Stop Control Register (SSTPR), the target unit counter (COUNTER) can be synchronously stopped.

### Software Synchronization Clear

By setting the relevant bits in the Software Synchronization Clear Control Register (SCLRR), the target unit counter (COUNTER) can be synchronously cleared.

### Software Synchronization Refresh

By setting the relevant bits in the Software Synchronization Refresh Control Register (SUPDR), the target unit counter (COUNTER) can be synchronously refreshed.

As shown in the figure below, setting SSTAR.SSTA1=SSTAR.SSTA2=SSTAR.SSTA3=1 enables software-synchronized start for units 1-3. Similarly, setting SSTPR.SSTP1=SSTPR.SSTP2=SSTPR.SSTP3=1 enables software-synchronized stop for units 1-3.

The software synchronization action related registers (SSTAR, SSTPR, SCLRR and SUPDR) are a group of registers independent of the unit and shared among various units. Each bit of this group of registers is enable only when writing 1, and writing 0 has no effect. When the SSTAR register is read, the timer status of each unit (counting stopped or counting) is read. When the SSTPR, SCLRR, or SUPDR is read, zero is read.

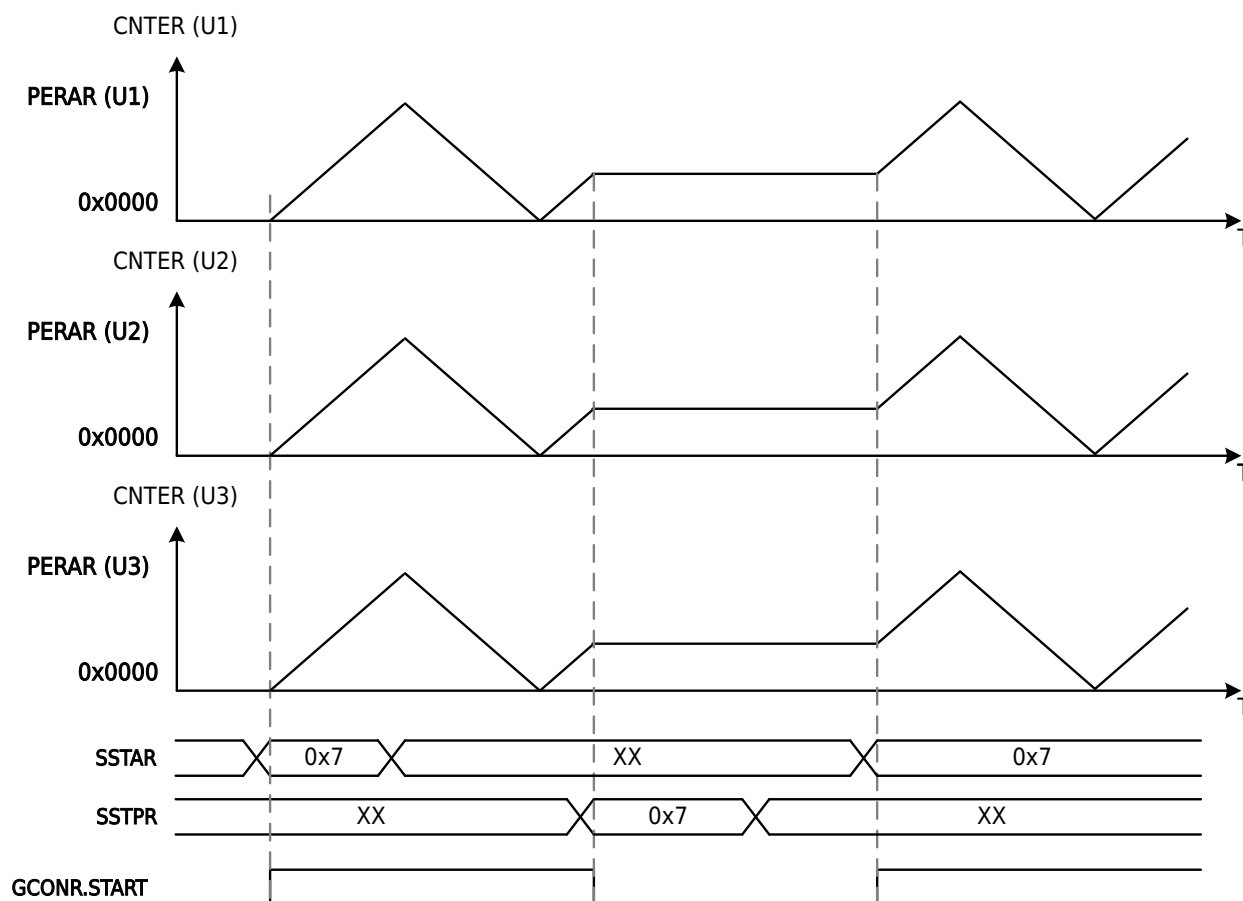


Figure 21-8 Software Synchronization Action

### 21.3.8 Hardware Synchronization

Each unit is equipped with two special general input ports (TMR6\_<t>\_PWMA and TMR6\_<t>\_PWMB), and all units share 4 general external trigger input ports (TMR6\_TRIGA, TMR6\_TRIGB, TMR6\_TRIGC, and TMR6\_TRIGD), as well as 4 internal trigger event input conditions that enable hardware-synchronized operations across units.

Internal trigger event sources can be selected by configuring the corresponding bits in the Hardware Trigger Event Selection Registers, as described in the "Automatic Operation System (AOS)" chapter titled "Timer6 Hardware Trigger Event Selection Register (TMR6\_TRGSELx) (x=0-3)". For detailed event-to-source mapping, refer to the "Interrupt Controller (INTC)" chapter. When using internal trigger functions, the peripheral trigger function enable bit in PWC\_FCG0 Register must first be set to 1.

#### Hardware Synchronization Start

Each unit can choose to start the timer by hardware, and the unit with the same hardware starting condition can be started synchronously when the starting condition is valid. The hardware start condition is configured in the Hardware Start Event Selection Register (HSTAR).

#### Hardware Synchronization Stop

Each unit can choose to stop the timer by hardware, and the unit with the same hardware stop condition can be synchronized to stop when the stop condition is valid. The hardware stop condition is configured in the Hardware Stop Event Selection Register (HSTOPR).

### Hardware Synchronization Clear

Each unit can choose to clear the timer by hardware, and select the unit with the same hardware clear condition to achieve synchronous clear when the clear condition is valid. The hardware clear condition is configured in the Hardware Clear Event Selection Register (HCLRR).

### Hardware Synchronization Refresh

Each unit can choose to refresh the timer by hardware, and select the unit with the same hardware refresh conditions, synchronous refreshing can be achieved when the refresh conditions are valid. The hardware refresh condition is set in the Hardware Refresh Event Selection Register (HUPDR).

### Hardware Synchronization Capture Input

Each unit can choose to implement the capture input function by hardware, and the units with the same capture input function condition can synchronize the capture input when the capture input function condition is valid. The hardware capture input conditions are set using the Hardware Capture Event Selection Registers (HCPAR, HCPBR).

### Hardware Synchronization Count

Each unit can choose to use the hardware input as the CLOCK for counting. By selecting units with the same hardware counting conditions, synchronous counting can be achieved when the hardware counting clock is valid. The count conditions are configured via the Hardware Count Up Event Selection Register (HCUPR) and Hardware Count Down Event Selection Register (HCDOR).

Selecting a hardware synchronous count function only designates the external clock source. It does not affect the start, stop, or clear operations of the timer. These actions must be configured separately.

The diagram below illustrates a hardware synchronization example across Units 1-3.

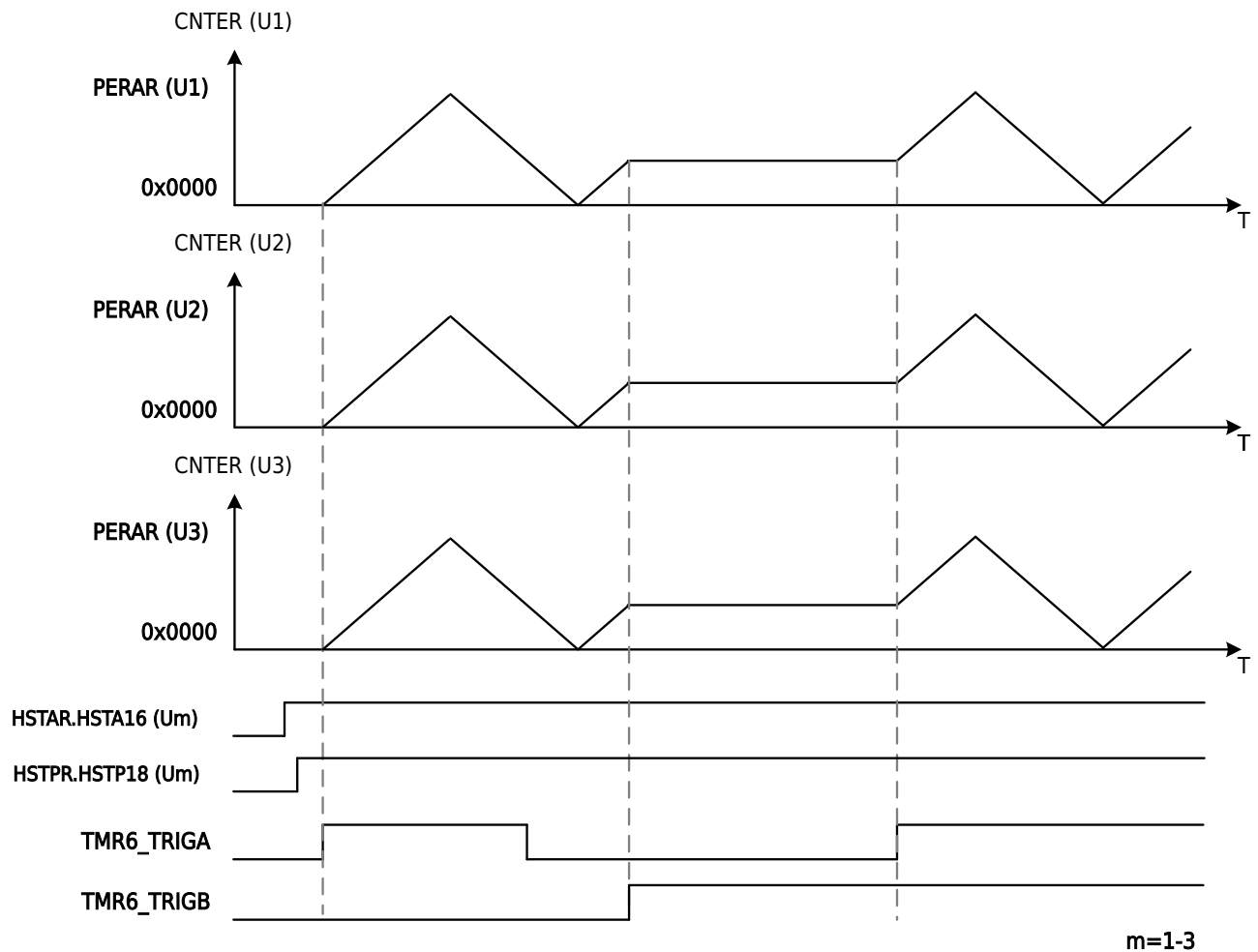


Figure 21-9 Hardware Synchronization Action

### 21.3.9 Pulse Width Measurement

When using the hardware trigger related functions (refer to "[Hardware Synchronization](#)") of the TMR6\_TRIGA-D port and the TMR6\_<t>\_PWMA-B port, each unit can realize independent pulse width measurement functions.

For example, by configuring the hardware start condition of the counter as the rising edge of TMR6\_TRIGA, and setting the hardware clear condition, stop condition, and the capture input condition for the GCMAR register to the falling edge of TMR6\_TRIGA, continuous pulse width measurement can be achieved. The corresponding operation is illustrated in the figure below.

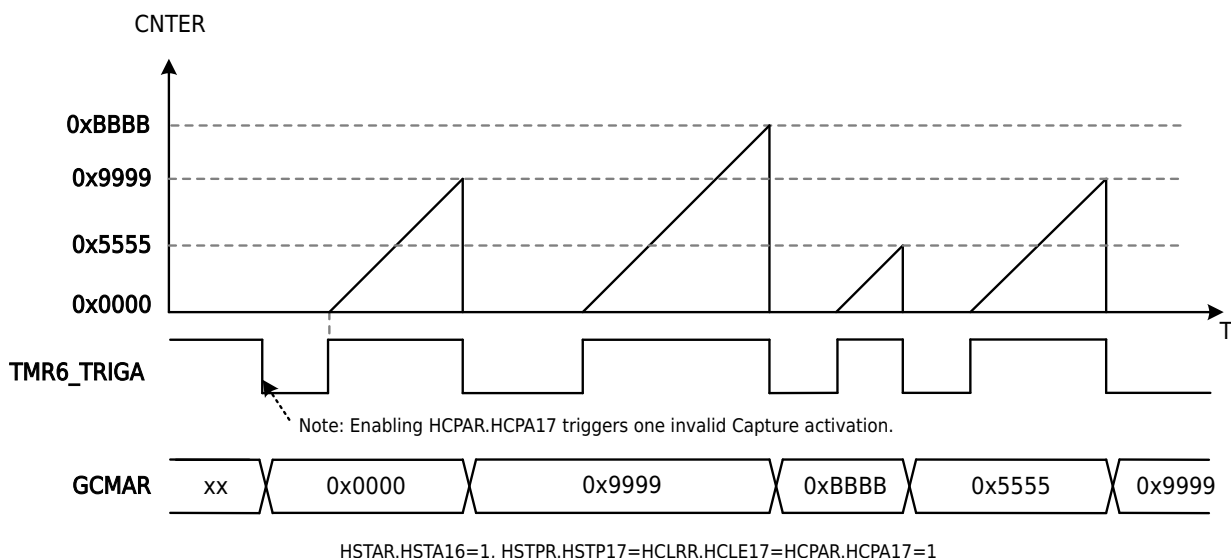


Figure 21-10 Pulse Width Measurement

### 21.3.10 Period Measurement

When using the hardware trigger related functions (refer to "[Hardware Synchronization](#)") of the TMR6\_TRIGA-Dport and the TMR6\_<t>\_PWMA-B port, each unit can realize independent period measurement functions.

For example, by configuring the hardware start condition, hardware clear condition, and the capture input condition for the GCMBR register all to the rising edge of TMR6\_TRIGB, continuous period measurement can be achieved. The corresponding operation is illustrated in the figure below.

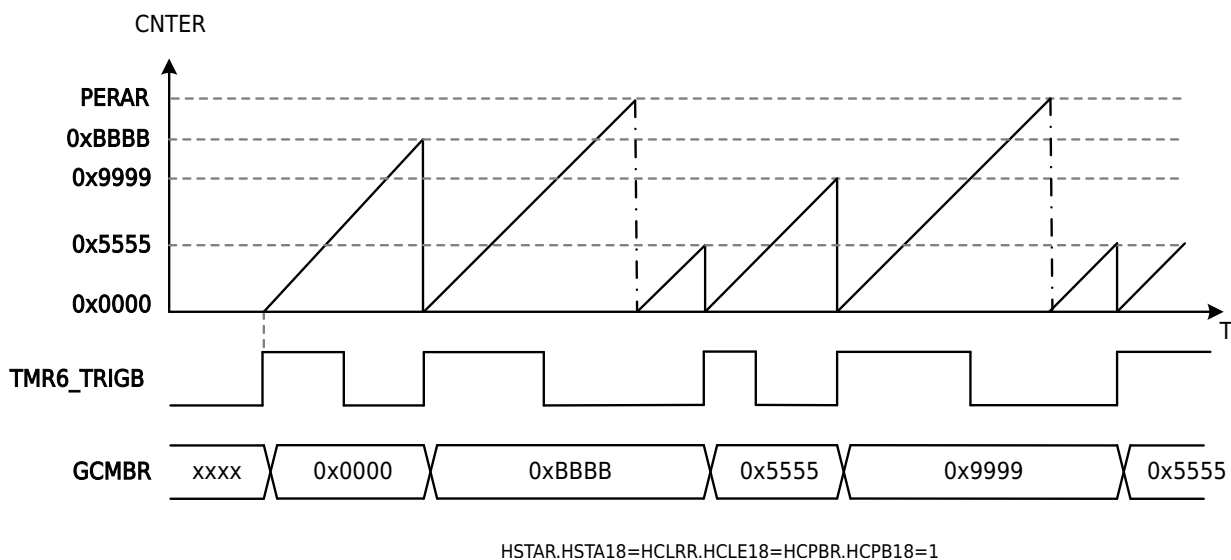


Figure 21-11 Period Measurement

### 21.3.11 Buffer Function

Timer6 supports buffer function for general period reference value, general compare reference value, special compare reference value, dead-time reference value, and more. This allows for hardware-based updates to the period, duty period, and dead time during counting. The period reference value, general compare reference value, and special compare reference value support single or double buffer function, while the dead-time reference value supports single buffer function.



### 21.3.11.1 Single Buffer Action

Single buffer action refers to configuring the buffer control register (BCONR.BE-NA<B><P><SPA><SPB>=1, BCONR.BSEA<B><P><SPA><SPB>=0) and the dead-time control register (DCONR.DTBENU<D>=1), at the buffer transfer time point, select the following events to occur:

1. The value in the General Period Reference Value Buffer Register (PERBR) is automatically transferred to the General Period Reference Value Register (PERAR).
2. The values in the General Compare Reference Value Buffer Registers (GCMCR, GCMDR) are automatically transferred to the General Compare Reference Value Registers (GCMAR, GCMBR) during compare output operations.
3. The values in the General Compare Reference Value Registers (GCMAR, GCMBR) are automatically transferred to the General Compare Reference Value Buffer Registers (GCMCR, GCMDR) during capture input operations.
4. The values in the Special Compare Reference Value Buffer Registers (SCMCR, SCMDR) are automatically transferred to the Special Compare Reference Value Registers (SCMAR, SCMBR).
5. The values in the Dead-Time Reference Value Buffer Registers (DTUBR, DTDBR) are automatically transferred to the Dead-Time Reference Value Registers (DTUAR, DTDAR).

The figure below shows the timing diagram of single buffer operation for general compare reference value registers during a compare output operation in Unit 1 (PCNAR.CAPMDA=0). From the diagram, it can be seen that changing the General Compare Reference Value Register (GCMAR) value during counting allows adjustment of the output duty period, while changing the General Period Reference Value Register (PERAR) value adjusts the output period.

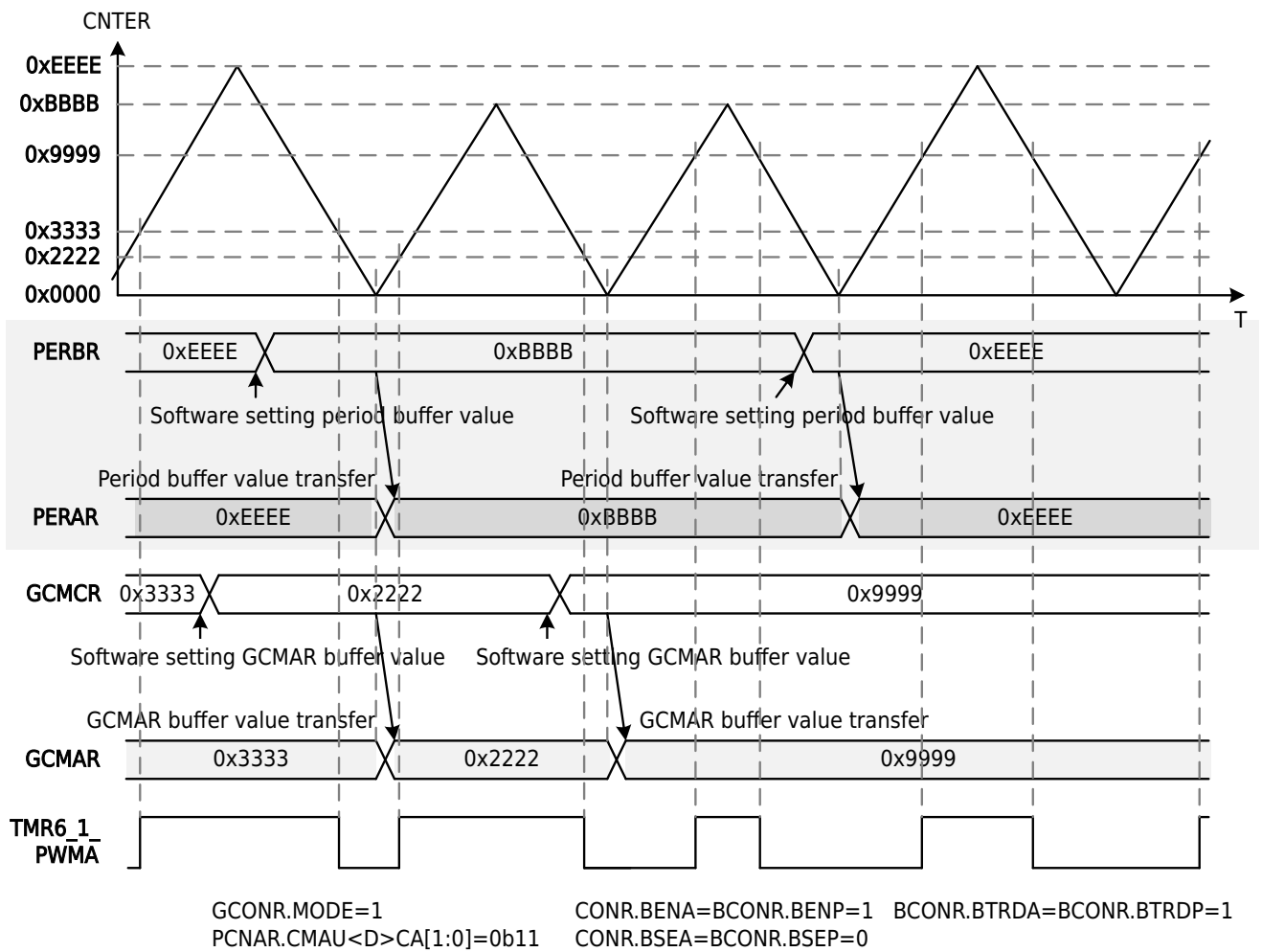


Figure 21-12 Timing Diagram of Compare Output in Single Buffer Mode

### 21.3.11.2 Double Buffer Action

Double buffer operation refers to the configuration where both BCONR.BENA<B><P><SPA><SPB>=1 and BCONR.BSEA<B><P><SPA><SPB>=1 are set. At the buffer transfer timing point, the following events occur:

1. The value in the General Period Reference Value Buffer Register (PERBR) is automatically transferred to the General Period Reference Value Register (PERAR), and the value in the General Period Reference Value Double Buffer Register (PERCR) is automatically transferred to General Period Reference Value Buffer Register (PERBR).
2. The values in the General Compare Reference Value Buffer Registers (GCMCR, GCMDR) are automatically transferred to the General Compare Reference Value Registers (GCMAR, GCMBR), and the values in the General Compare Reference Value Double Buffer Registers (GCMER, GCMFR) are automatically transferred to General Compare Reference Value Buffer Registers (GCMCR, GCMDR) during compare output operations.
3. The values in the General Compare Reference Value Buffer Registers (GCMCR, GCMDR) are automatically transferred to General Compare Reference Value Double Buffer Registers (GCMER, GCMFR), and the values in General Compare Reference Value Registers (GCMAR, GCMBR) are automatically transferred to General Compare Reference Value Buffer Registers (GCMCR, GCMDR) during capture input operations.
4. The values in the Special Compare Reference Value Buffer Registers (SCMCR, SCMDR) are automatically transferred to the Special Compare Reference Value Registers (SCMAR, SCMBR), and the values

in the Special Compare Reference Value Double Buffer Registers (SCMER, SCMFR) are automatically transferred to Special Compare Reference Value Buffer Registers (SCMCR, SCMDR).

The diagram below illustrates the timing of capture input in double buffer mode, triggered by Internal Trigger Event 0.

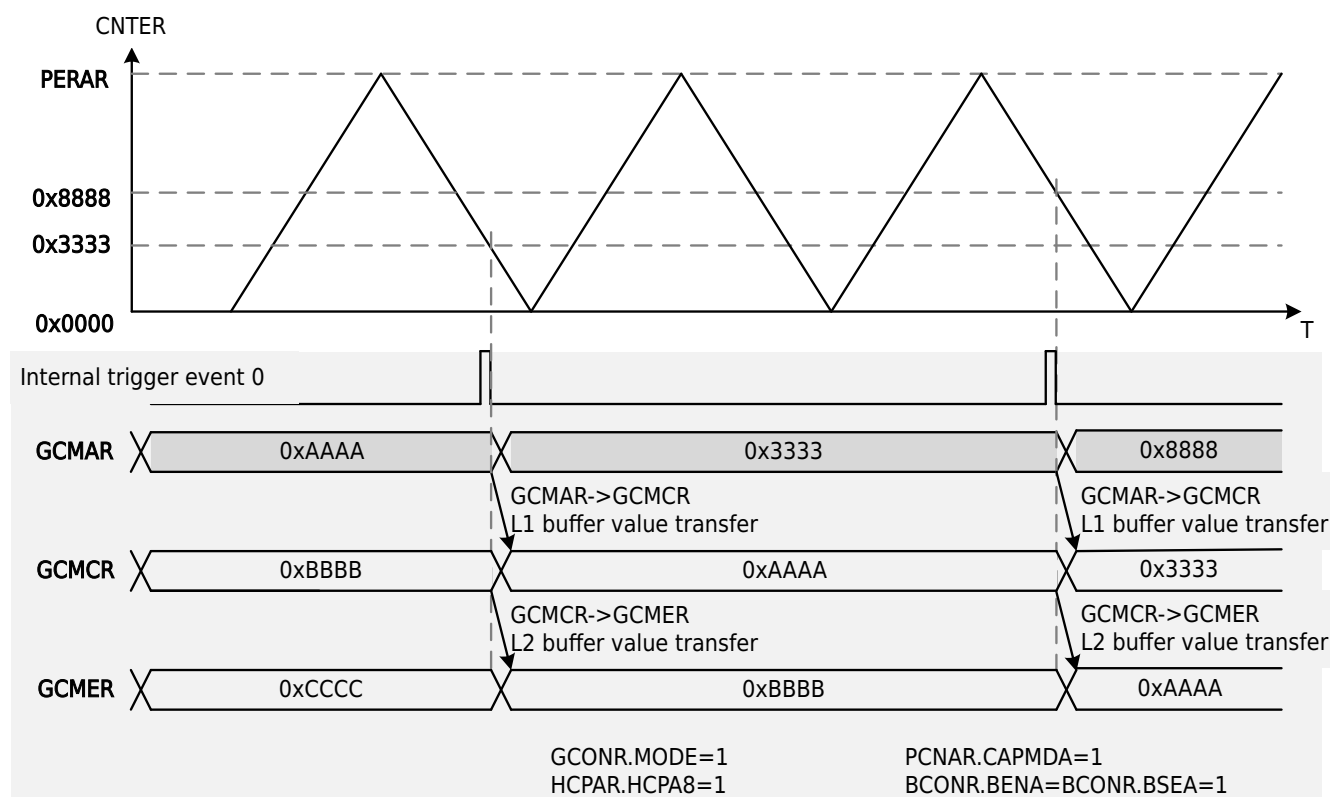


Figure 21-13 Capture Input Timing in Double Buffer Mode

### 21.3.11.3 Buffer Transfer Node

#### Compare Output (Sawtooth Wave) Buffer Transfer

When the buffer function is enabled (BCONR.BENA<B><P><SPA><SPB>=1, DCONR.DTBENU<D>=1) and compare output counting (PCNA<B>R.CAPMDA<B>=0) and the sawtooth wave count mode (GCONR.MODE=0), buffer transfers for the general period reference value, general compare reference value, special compare reference value, and dead-time reference value occur at either the counter overflow in up-counting mode or the counter underflow in down-counting mode.

The diagram below illustrates the buffer operation during up-counting in sawtooth mode:

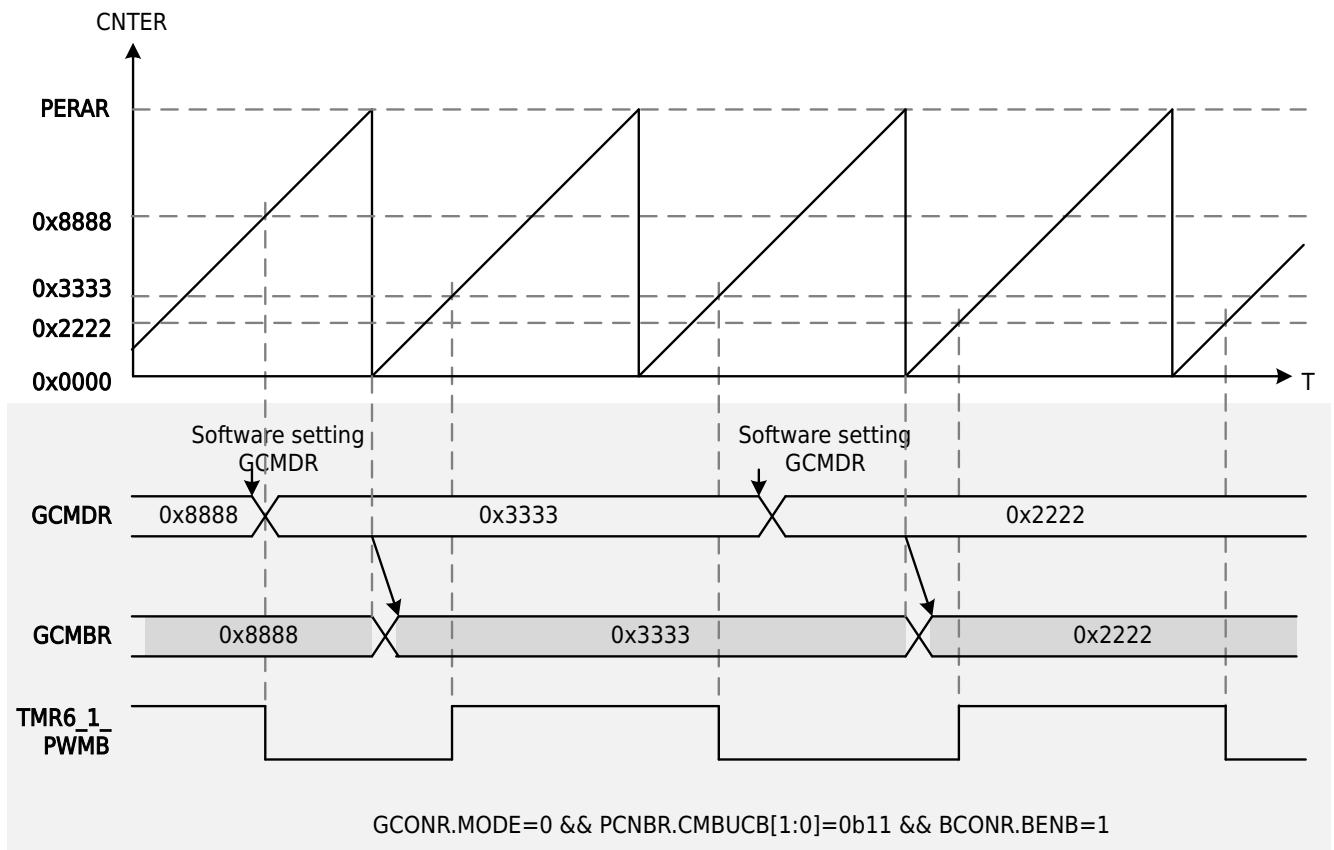


Figure 21-14 Count Buffer Operation in Sawtooth Mode

**Note**

- If a clear operation occurs in sawtooth mode, it is treated as a counter overflow. At this moment, each reference will be transferred based on the corresponding buffer configuration (single buffer, double buffer, etc.).
- In hardware count mode, a clear operation is also treated as a counter overflow. In such cases, only the general period and general compare reference value are transferred based on the corresponding buffer configuration (single buffer, double buffer, overflow transfer, underflow transfer, etc.); other references are not transferred.

**Compare Output (Triangle Wave) Buffer Transfer**

When the buffer function is enabled (BCONR.BENA<B><P><SPA><SPB>=1, DCONR.DTBENU<D>=1) and compare output count (PCNA<B>R.CAPMDA<B>=0) and triangle wave count mode (GCONR.MODE=1), the buffer transfer timing for general period reference value, general compare reference value, special compare reference value, and dead-time reference value is controlled by corresponding control bits in the registers. When BCONR.BTRUA<B><P><SPA><SPB>=1 or DCONR.DTBTRU=1, buffer transfer occurs at the peak point of the triangle wave. When BCONR.BTRDA<B><P><SPA><SPB>=1 or DCONR.DTBTRD=1, buffer transfer occurs at the valley point of the triangle wave.

Figure 21-15 : Count Buffer Operation 1 in Triangle Mode shows the buffer action when the triangular wave count reaches the valley point. Figure 21-16 : Count Buffer Operation 2 in Triangle Mode is a diagram shows the buffer action when the triangular wave count reaches the peak point and valley point.

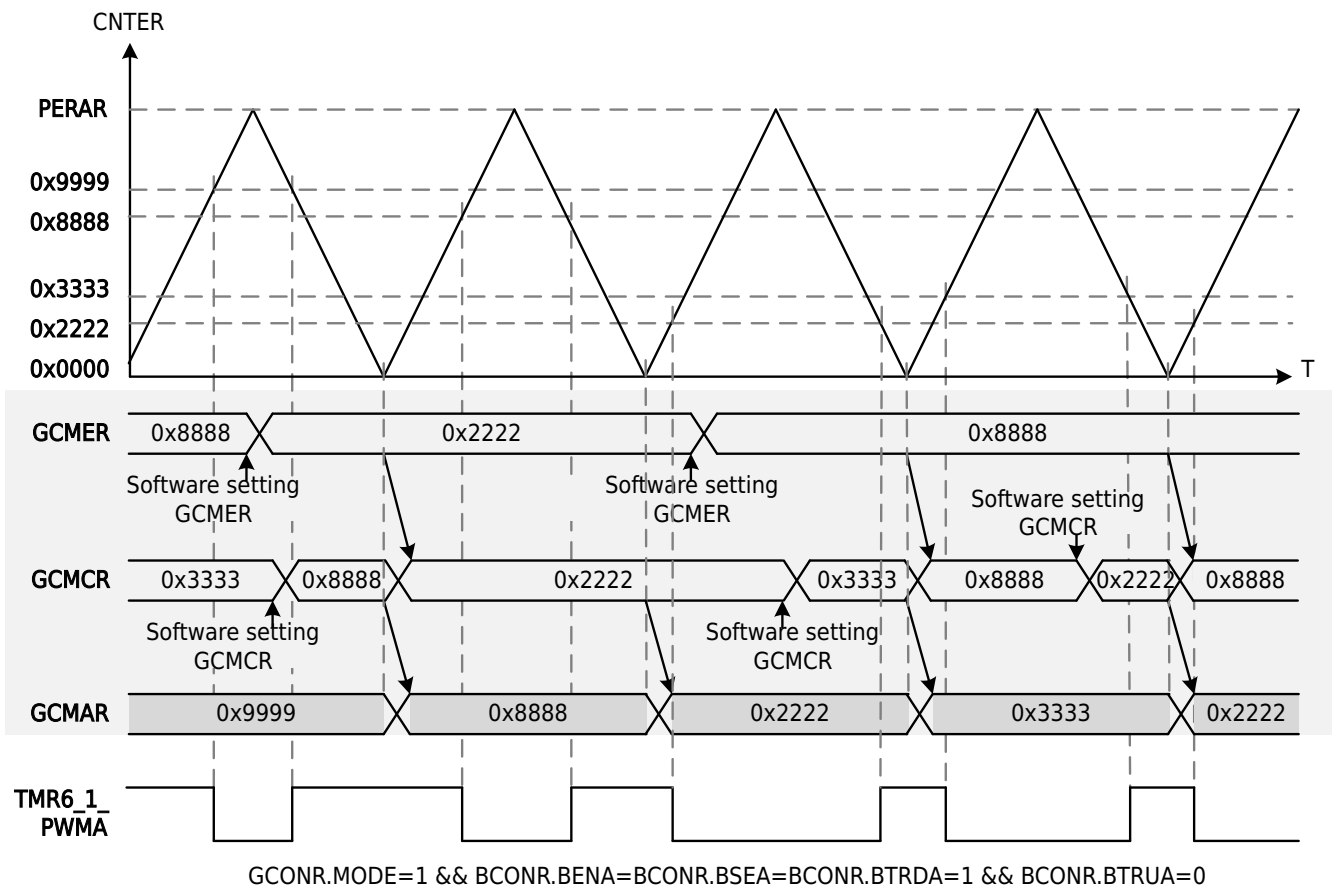


Figure 21-15 Count Buffer Operation 1 in Triangle Mode

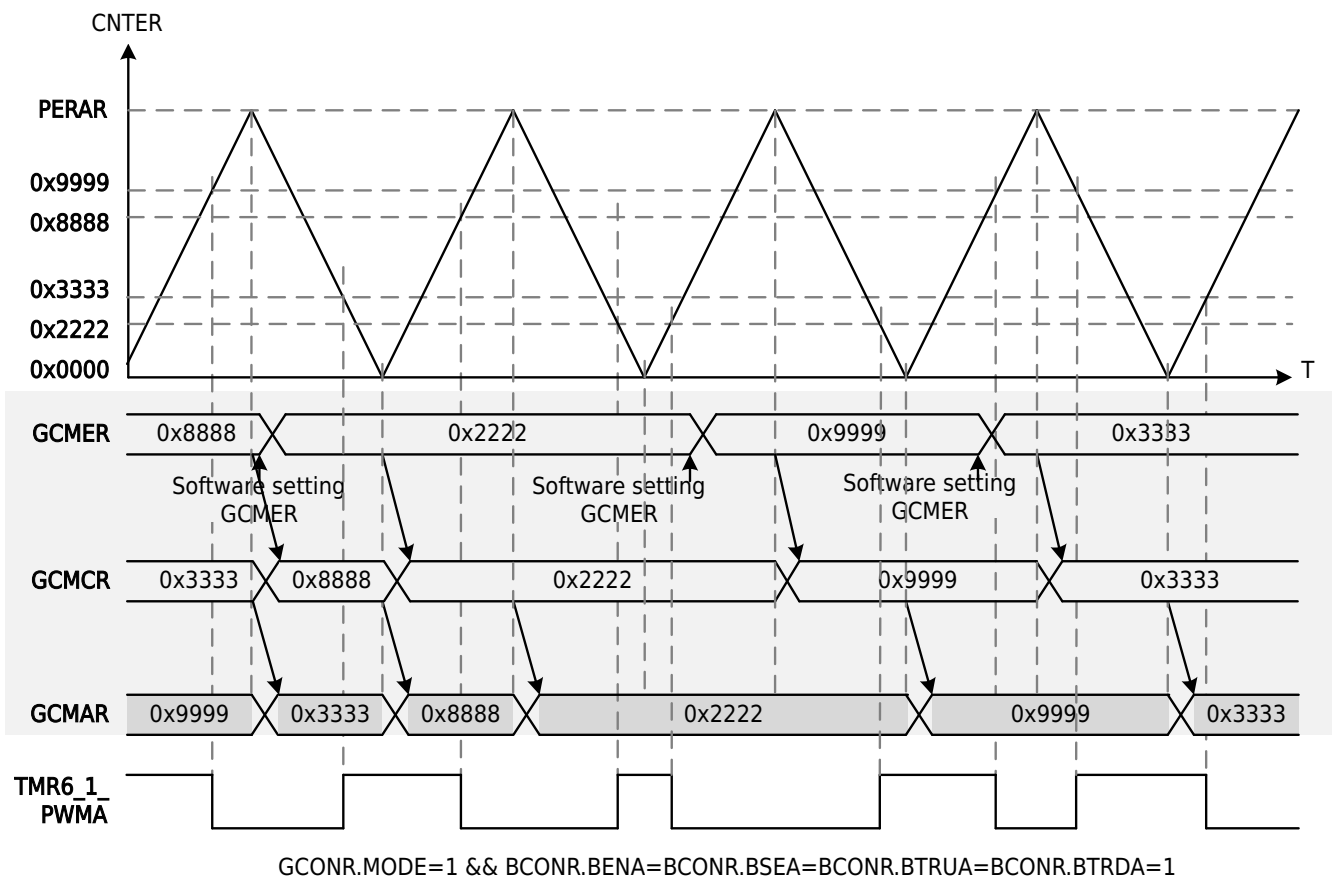


Figure 21-16 Count Buffer Operation 2 in Triangle Mode

## Capture Input Buffer Transfer

When capture input action ( $PCNA\langle B\rangle.R.CAPMDA\langle B\rangle=1$ ) is valid, the general compare reference value support buffer transfer. If buffer configuration is enable ( $BCONR.BENA\langle B\rangle=1$ ), a buffer transfer occurs at each capture input action. Capture input action supports both single buffer and double buffer modes (which are controlled by the  $BCONR.BSEA\langle B\rangle$  setting).

## 21.3.12 General PWM Output

### 21.3.12.1 Single-Edge Aligned Independent PWM Output

In sawtooth wave count mode ( $GCONR.MODE=0$ ), each unit's two ports  $TMR6\_<t>\_PWMA$  and  $TMR6\_<t>\_PWMB$  can independently output PWM signals. When the same output level transition is configured at the counter overflow point, single-edge aligned independent PWM output can be achieved.

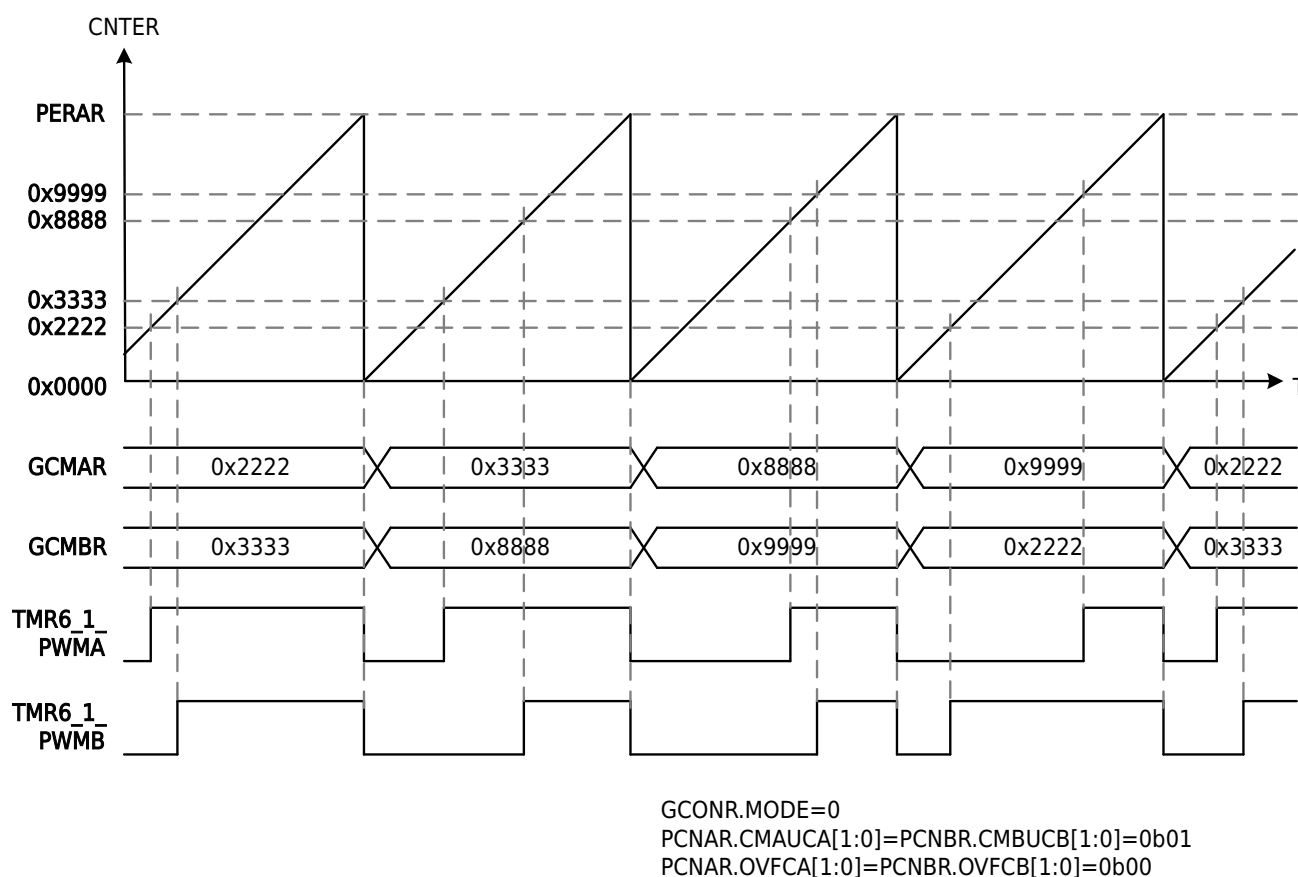


Figure 21-17 Single-Edge Aligned Independent PWM

### 21.3.12.2 Double-Edge Symmetrical Independent PWM Output

In triangle wave count mode ( $GCONR.MODE=1$ ), each unit's two ports  $TMR6\_<t>\_PWMA$  and  $TMR6\_<t>\_PWMB$  can independently output PWM signals. When level changes are configured at the count comparison match points, and the levels remain unchanged at the count peak and valley points, double-edge symmetrical independent PWM output can be achieved. The following figure shows that the  $TMR6\_<t>\_PWMA$  port and  $TMR6\_<t>\_PWMB$  port implement double-edge symmetrical independent output PWM.

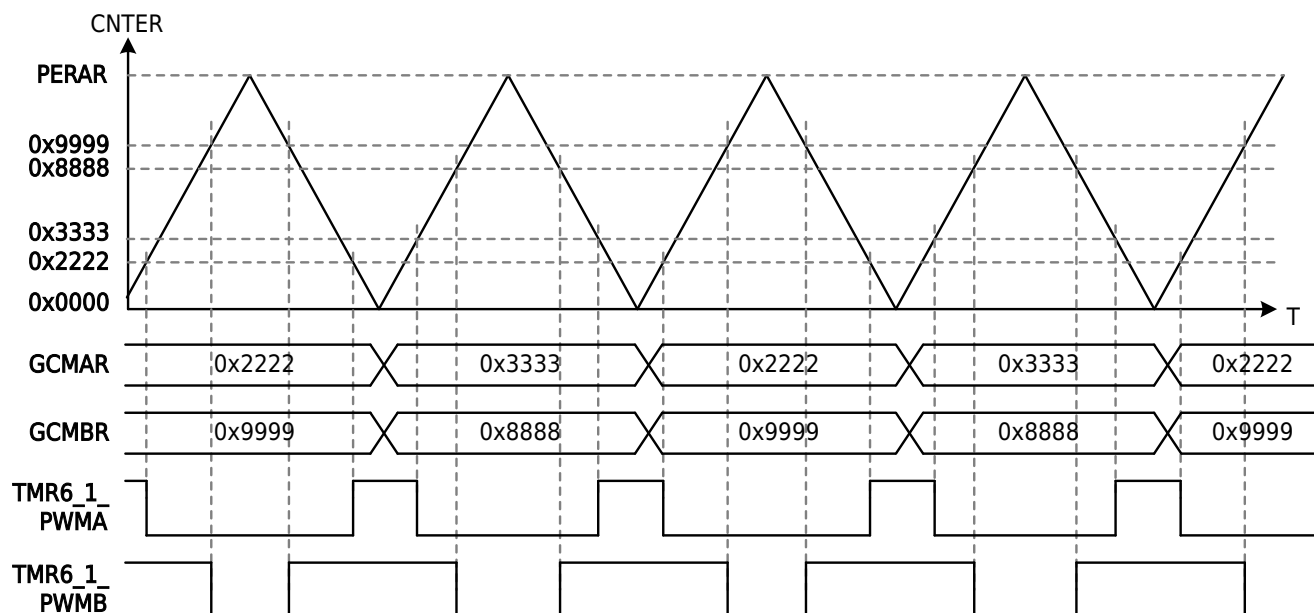


Figure 21-18 Double-Edge Symmetrical independent PWM

### 21.3.12.3 Double-Edge Symmetrical Complementary PWM Output

In triangle wave count mode (GCONR.MODE=1), by configuring the level states for the ports during count start, comparison match, and count overflow, a pair of complementary PWM waveforms can be output on the TMR6\_<t>\_PWMA and TMR6\_<t>\_PWMB ports.

According to the different ways of assigning the GCMBR reference value, the output of double-edge symmetrical complementary PWM can be divided into "software-set GCMBR complementary PWM output" and "hardware-set GCMBR complementary PWM output".

#### Software-Set GCMBR Complementary PWM Output

Software setting means that in triangle wave mode, the general compare reference value register (GCMBR) used for the waveform output of the TMR6\_<t>\_PWMB port is directly written by the CPU, etc., and has no direct relationship with the value of GCMAR.

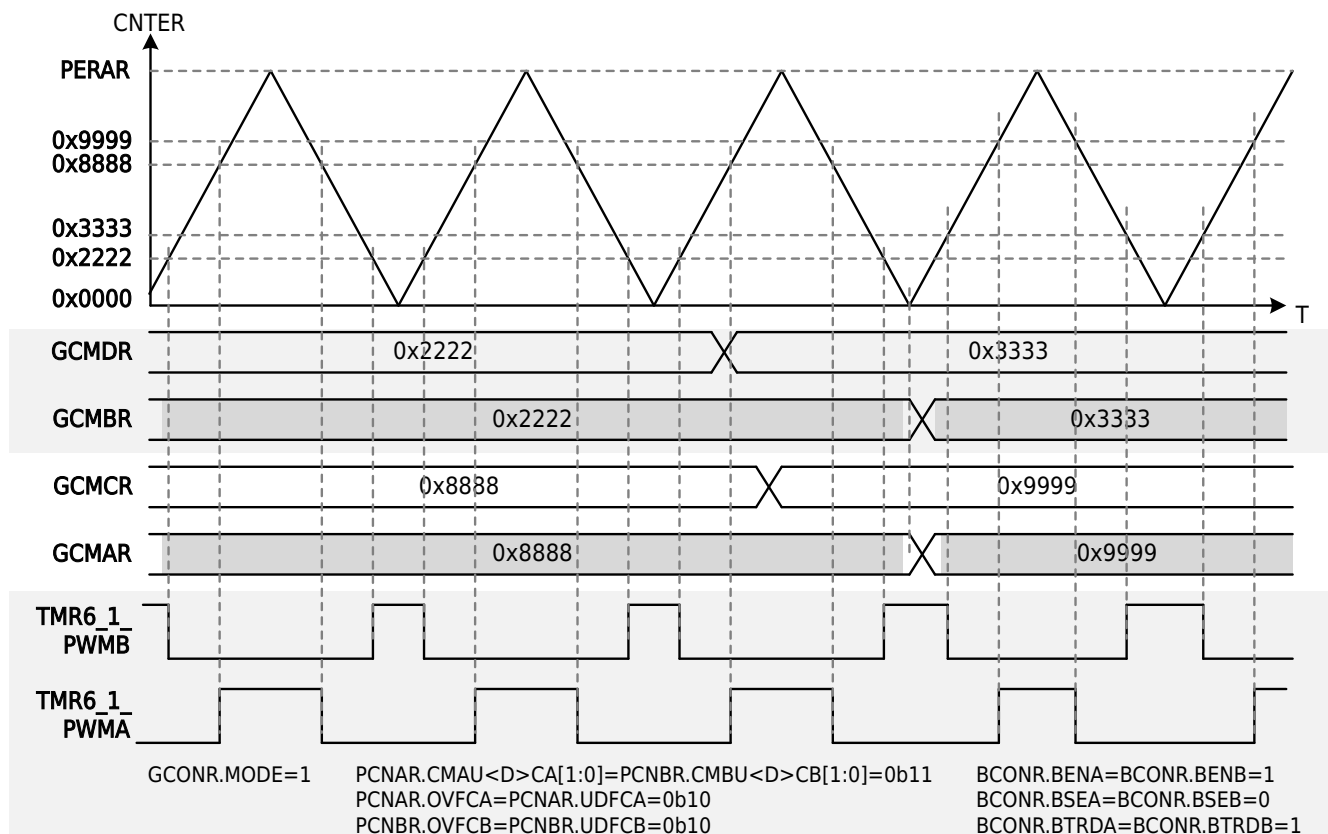


**Figure 21-19 Software-Set GCMBR Complementary PWM Output**

## Hardware-Set GCMBR Complementary PWM Output

The hardware setting means that in the triangle wave mode, the value of the general compare reference value register (GCMBR) used for the waveform output of the TMR6\_<t>\_PWMB port is determined by the value operation of the general compare reference value register (GCMAR) and the dead-time reference value register (DTU<D>AR) .

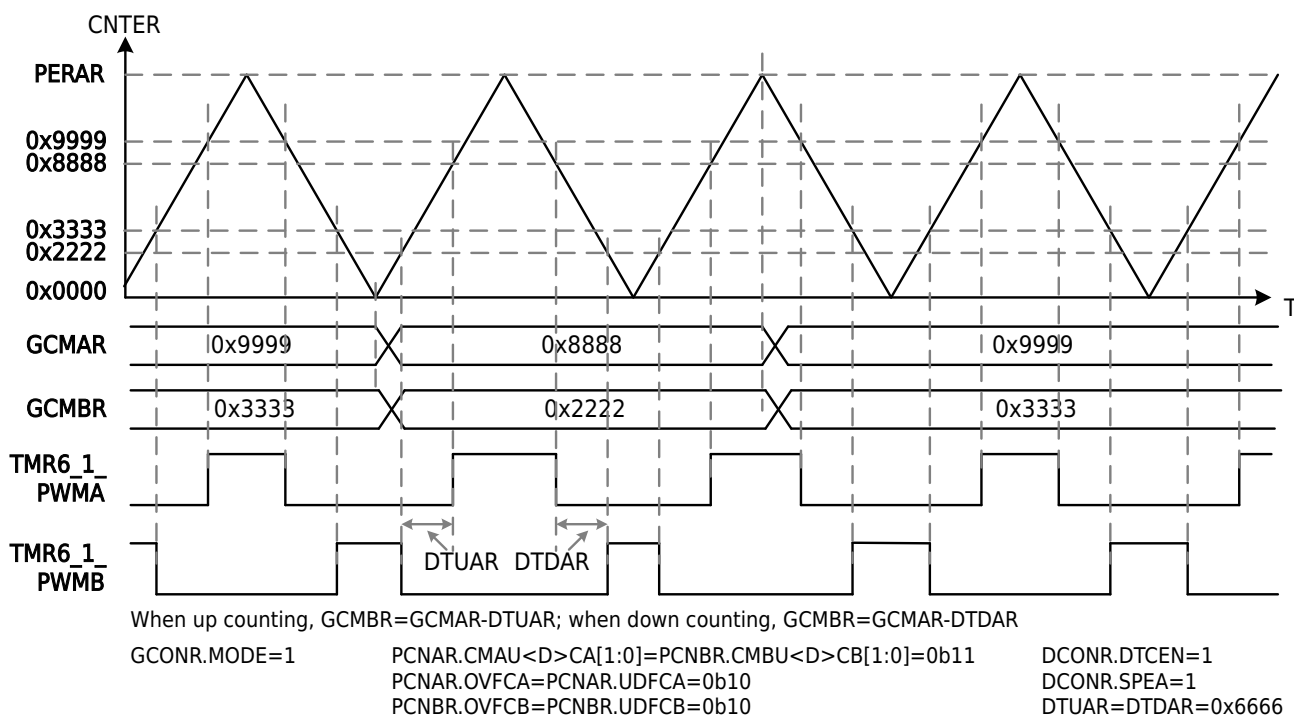


**Figure 21-20 Hardware-Set GCMBR Complementary PWM Output**



### 21.3.12.4 Double-Edge Asymmetrical PWM Output

In triangle wave count mode (GCONR.MODE=1), each unit's two ports, TMR6\_<t>\_PWMA and TMR6\_<t>\_PWMB, are capable of independently outputting PWM waveforms. Each port can change its output levels according to the corresponding comparison results based on the General Compare Reference Value Registers (GCMAR, GCMBR). If during the up-counting and down-counting phases of the counter, the level transitions on the TMR6\_<t>\_PWMA port are controlled by the comparison results of the GCMAR and GCMBR registers, a double-edge asymmetrical PWM output can be achieved on the TMR6\_<t>\_PWMA port. Below is the diagram for the double-edge asymmetrical PWM output on the TMR6\_<t>\_PWMA port:

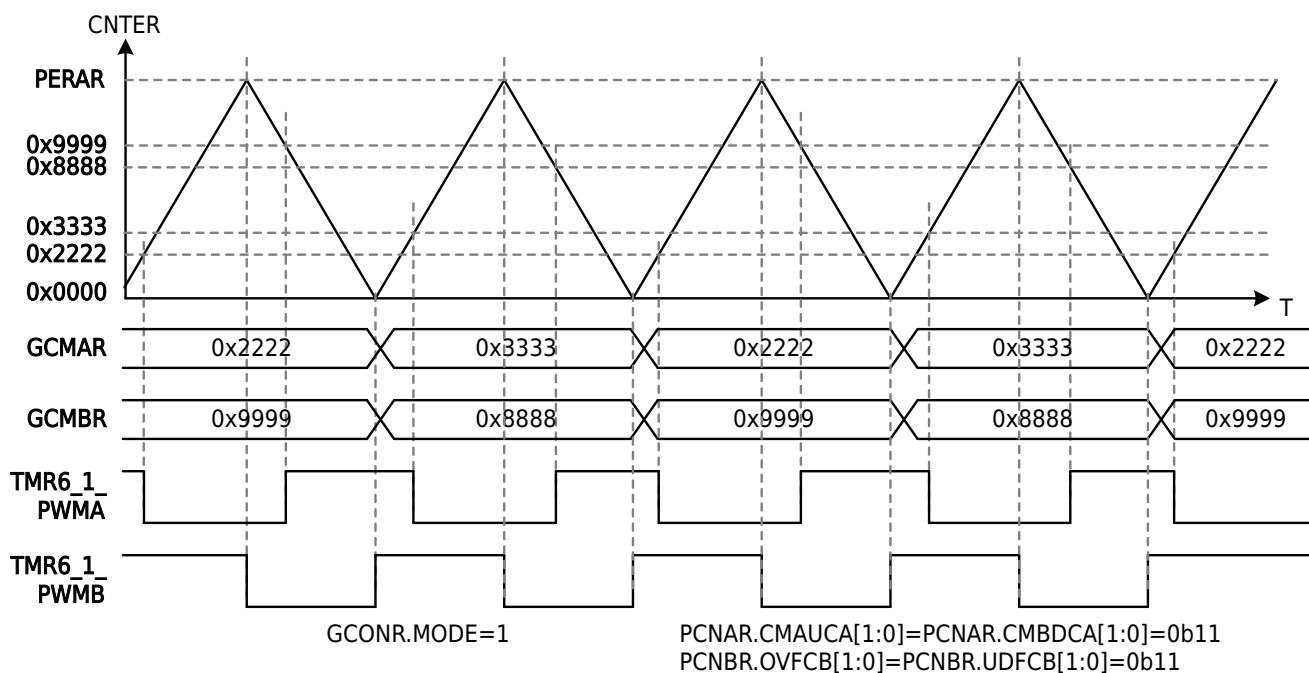


Figure 21-21 Double-Edge Asymmetrical PWM Output

### 21.3.12.5 Multi-Phase PWM Output between Units

The TMR6\_<t>\_PWMA and TMR6\_<t>\_PWMB ports of each unit can output 2-phase PWM waves. By combining multiple units and combining software and hardware synchronization actions, multi-phase PWM wave output can be achieved. [Figure 21-22 : Single-Edge Aligned Independent PWM](#) shows the single-edge aligned independent PWM output by the combination of Unit 1 and Unit 2. [Figure 21-23 : Double-Edge Symmetrical Complementary PWM with Dead Time](#) shows the double-edge symmetrical complementary PWM output by the combination of Unit 1 and Unit 2.

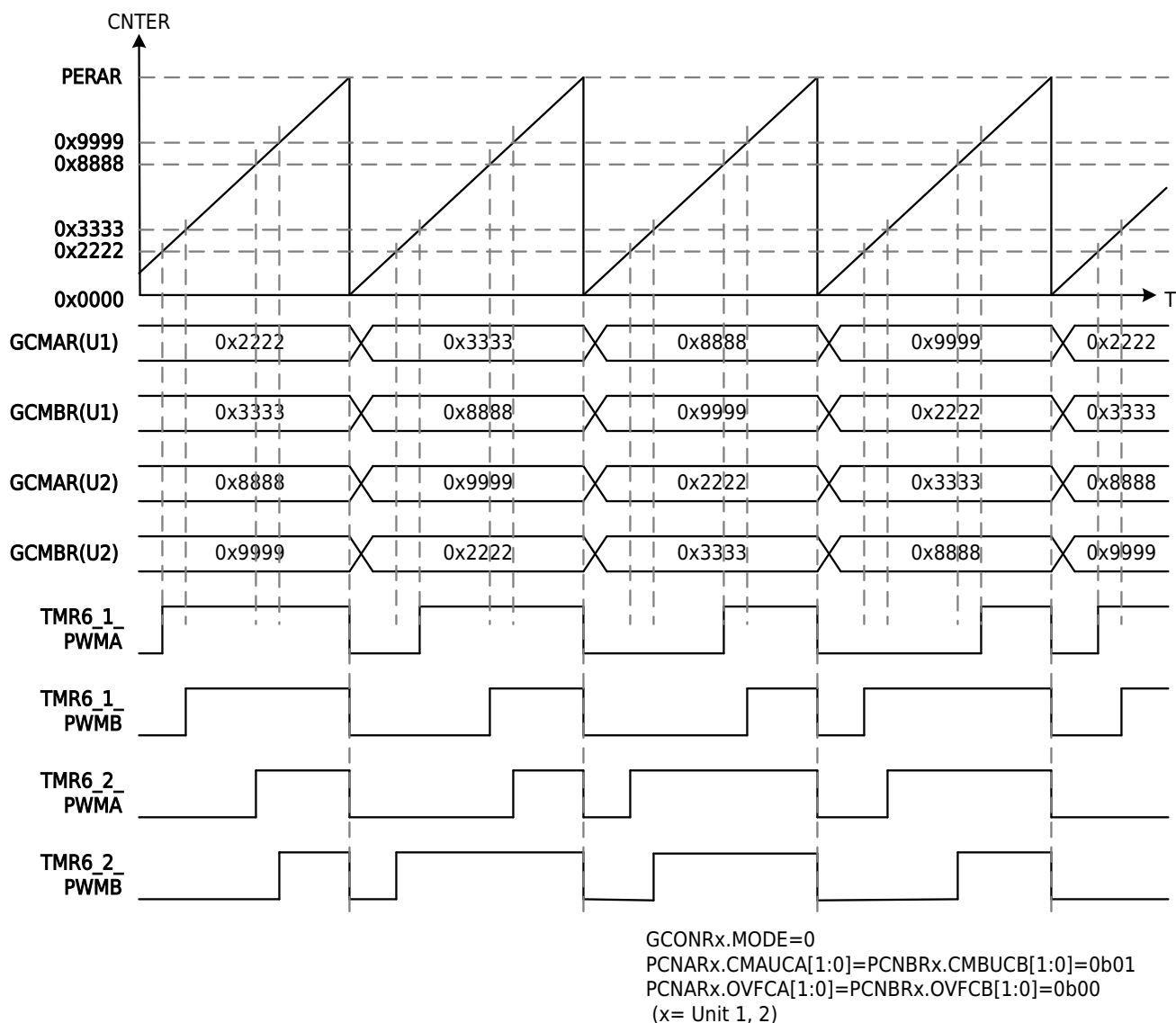


Figure 21-22 Single-Edge Aligned Independent PWM

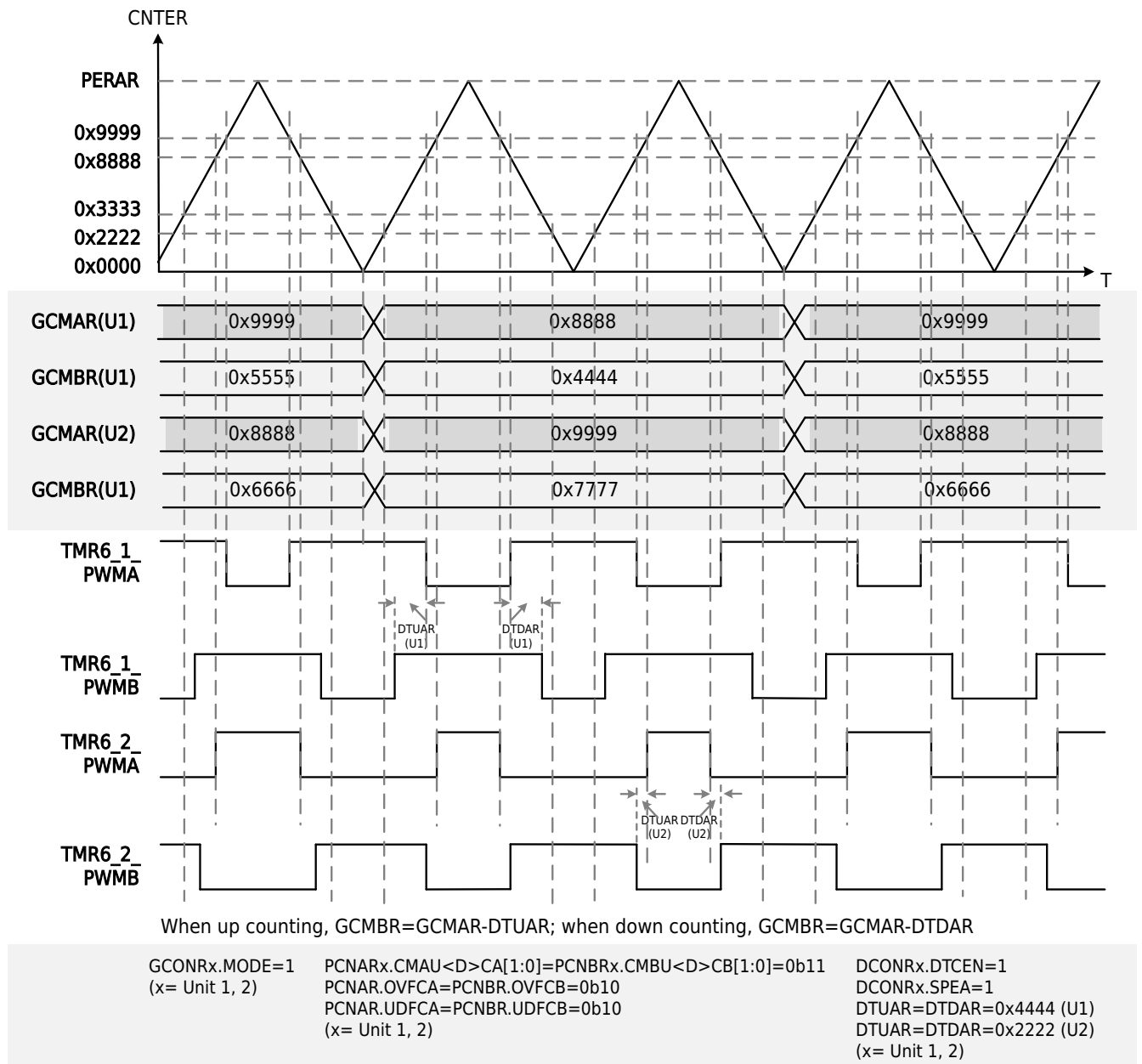


Figure 21-23 Double-Edge Symmetrical Complementary PWM with Dead Time

### 21.3.13 Quadrature Encoding Count

The TMR6\_<t>\_PWMA input is treated as the AIN input, the TMR6\_<t>\_PWMB input as the BIN input, and any input in TMR6\_TRIGA-D is treated as the ZIN input. With this configuration, Timer6 can implement quadrature encoding counting with three inputs.

For a single unit, the AIN and BIN inputs can independently perform position counting mode. For two units, the combination of AIN, BIN, and ZIN enables revolution counting mode, with one unit used for position counting and the other for revolution counting.

In revolution counting mode, the 8 units of Timer6 can be freely combined to act as either position counting or revolution counting units. However, when the Z phase masking function is enabled, the combinations are fixed as follows: Units 1 and 2 are combined, with Unit 1 as the position counting unit and Unit 2 as the revolution counting unit, each performing position and revolution counting respectively. Units 3 and 4 are combined, with Unit 3 as the position counting unit and Unit 4 as the revolution counting unit, each performing position and revolution counting respectively. Units 5 and

6 are combined, with Unit 5 as the position counting unit and Unit 6 as the revolution counting unit, each performing position and revolution counting respectively. Units 7 and 8 are combined, with Unit 7 as the position counting unit and Unit 8 as the revolution counting unit, each performing position and revolution counting respectively.

The counting conditions for AIN and BIN are determined by setting the quadrature relationship of TMR6\_<t>\_PWMA and TMR6\_<t>\_PWMB in the Hardware Count Up Event Selection Register (HCUPR) and Hardware Count Down Event Selection Register (HCDOR). The input action for ZIN is controlled by setting the Hardware Clear Event Selection Register (HCLRR) of the position unit to clear the position timer of the position counting unit, and the Hardware Count Up Event Selection Register (HCUPR) of the revolution unit to count the revolution timer of the revolution counting unit.

### 21.3.13.1 Position Counting Mode

The quadrature encoder position counting mode enables the implementation of basic counting, phase difference counting, and direction counting based on the input signals from AIN and BIN.

#### Basic Counting

Basic counting uses the input clock from either the AIN or BIN port to count.

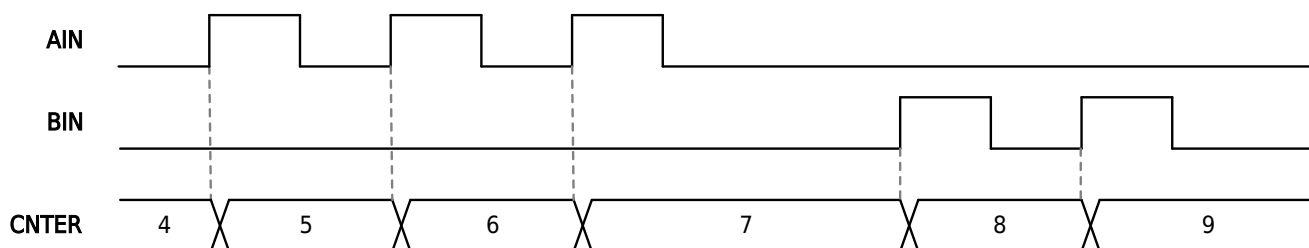
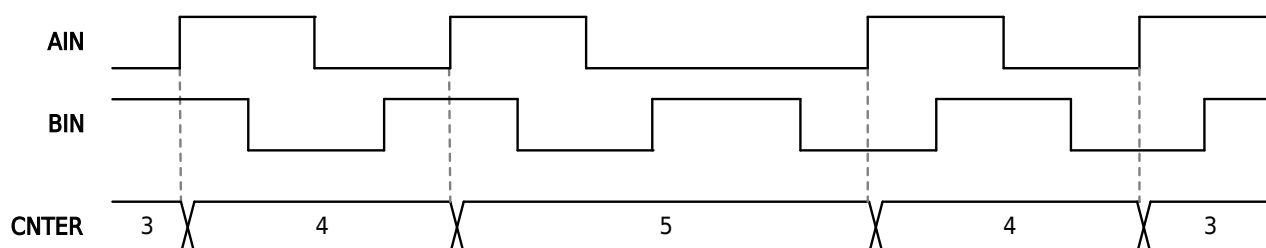


Figure 21-24 Position Mode-Basic Counting

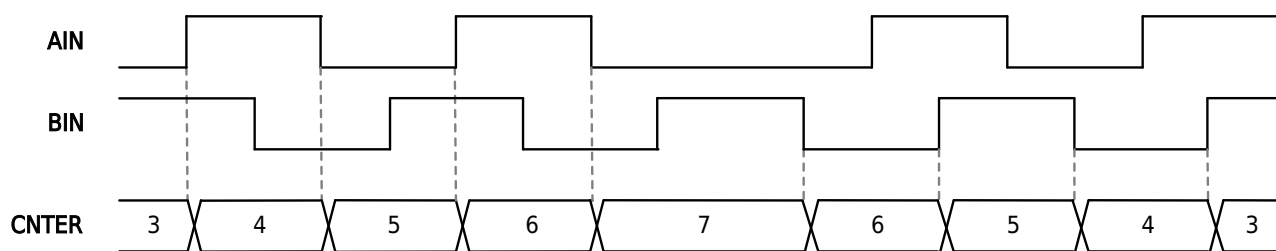
#### Phase Difference Counting

Phase difference counting operates based on the phase relationship between AIN and BIN. Depending on the configuration, the system can perform 1x counting, 2x counting, or 4x counting.



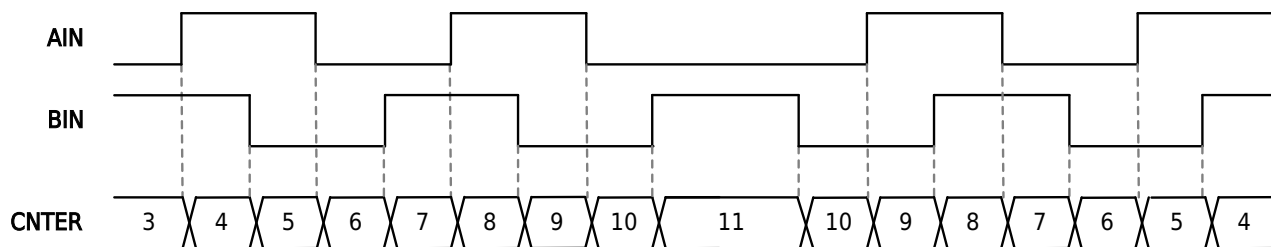
HCUPR.HCUP6=0b1 && HCDOR.HCDO4=0b1

Figure 21-25 Position Counting Mode-Phase Difference Counting (1x Counting)



HCUPR.HCUP6=HCUPR.HCUP5=0b1 && HCDOR.HCDO2=HCDOR.HCDO1=0b1

Figure 21-26 Position Counting Mode-Phase Difference Counting (2x Counting)

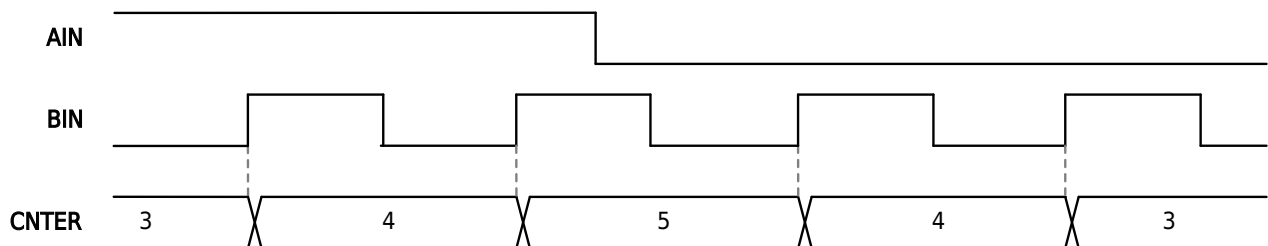


HCUPR.HCUP6=HCUPR.HCUP5=HCUPR.HCUP3=HCUPR.HCUP0=0b1  
HCDOR.HCDO7=HCDOR.HCDO4=HCDOR.HCDO2=HCDOR.HCDO1=0b1

Figure 21-27 Position Counting Mode-Phase Difference Counting (4x Counting)

### Direction Counting

Direction counting sets the AIN input as the direction control, and uses the BIN input as the counting clock.



HCUPR.HCUP2=0b1 && HCDOR.HCDO0=0b1

Figure 21-28 Position Counting Mode-Direction Counting

### 21.3.13.2 Revolution Counting Mode

The quadrature encoder revolution counting mode adds ZIN input events to the basic AIN/BIN counting mechanism to support features such as revolution count tracking. In this mode, depending on the revolution timer's counting method, the following functionalities can be realized: Z-phase counting, position overflow counting, and mixed counting.

#### Z-phase Counting

Z-phase counting refers to a counting operation in which the revolution counting unit counting based on the ZIN input signal, while the position counting unit is reset to zero.

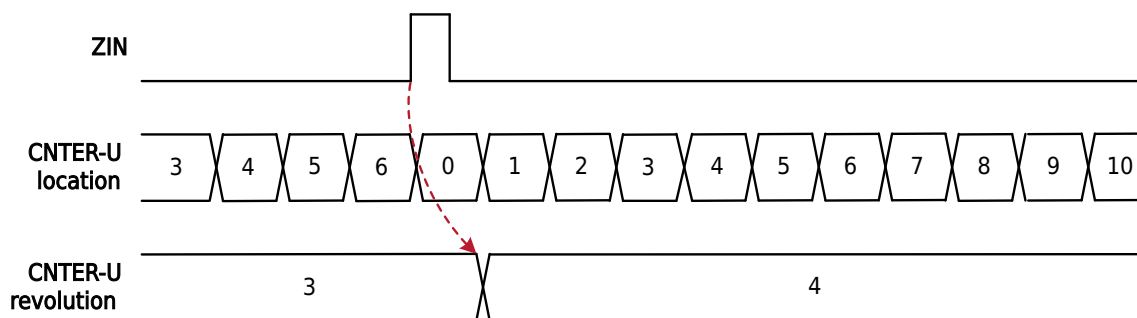


Figure 21-29 Revolution Counting Mode-Z-phase Counting

### Position Overflow Counting

Position overflow counting refers to the condition where the position counting unit generates an overflow or underflow event, which then triggers the revolution counting unit to count once (In this counting mode, the ZIN input does not trigger counting in the revolution counting unit or clear in the position counting unit).

The overflow event from the position counting unit is routed via the internal trigger event interface to the revolution counting unit. This enables position overflow counting. Select a bit from Bit8-Bit11 in the Hardware Count Up Event Selection Register (HCUPR) or Hardware Count Down Event Selection Register (HCDOR) of the revolution counting unit. Set the event number in the Hardware Trigger Event Selection Register (TMR6\_TRGSELx, x=0-3) to the overflow or underflow event of the position counting unit. For details on the register settings, refer to the chapter "Automatic Operation System (AOS) : Timer6 Hardware Trigger Event Selection Register (TMR6\_TRGSELx) (x=0-3)". For event-to-interrupt mapping, refer to the "Interrupt Controller (INTC)" chapter.

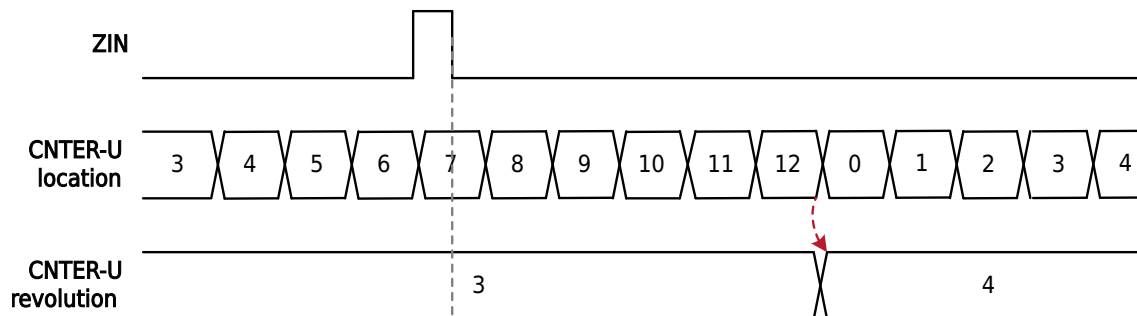


Figure 21-30 Revolution Counting Mode-Position Overflow Counting

### Mixed Counting

Mixed counting refers to a combined counting method that integrates both Z-phase counting and position overflow counting. Its implementation method is also a combination of the above two counting methods.

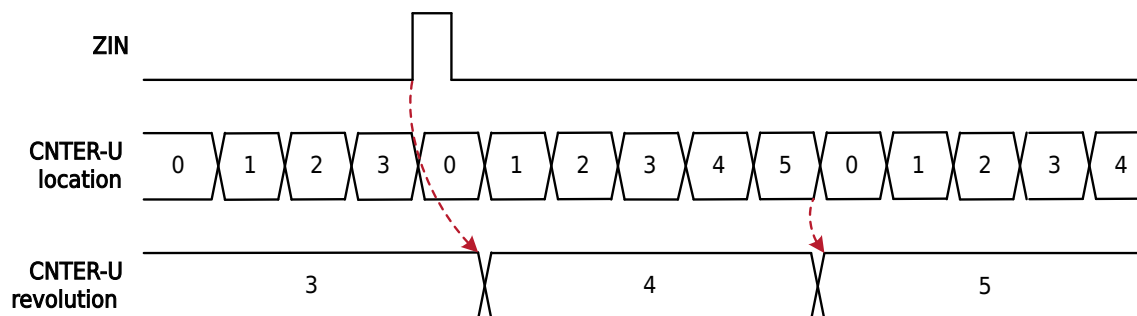


Figure 21-31 Revolution Counting Mode-Mixed Counting

### 21.3.13.3 Z-Phase Action Mask

In the Z-phase counting function or mixed counting function of the revolution counting mode, you can set a few periods after the overflow point or underflow point of the position timer (GCONR.ZMSKVA setting) to mask the active input of ZIN and not count the revolution counting unit and clear the position counting unit.

When GCONR.ZMSKPOS bit of the position counting unit is 1, the Z-phase mask function of the position counting unit is enabled, and the number of periods of Z-phase mask is set by GCONR.ZMSKVAL; when GCONR.ZMSKREV bit of the revolution counting unit is 1, the Z-phase mask function of the revolution counting unit is enabled.

Figure below shows the mixed counting in the revolution counting mode. When there is a ZIN phase input within 4 counting periods after the position counting unit overflows, the action of the ZIN phase input is invalid, that is, the revolution counting unit does not count and the position counting unit is not cleared; the subsequent ZIN phase input acts normally.

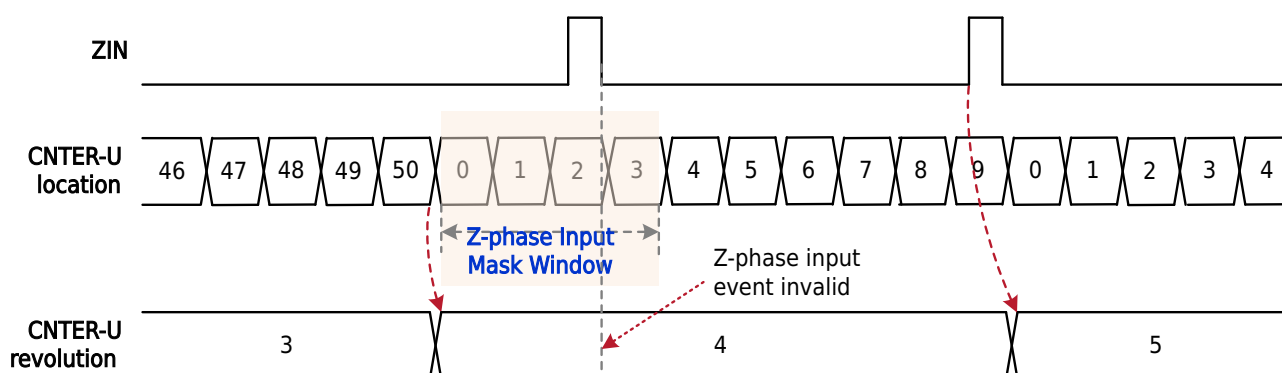


Figure 21-32 Revolution Counting Mode-Mixed Count Z-Phase Masking Action Instances 1

The figure below illustrates an example of mixed counting in revolution counting mode, where a change in counting direction occurs in the third period after the position counting unit overflows. As a result, the configured masking period of four periods becomes inactive (the Z-phase masking is actively applied for only three periods) before the counter begins down-counting. After a count underflow occurs in the position counting unit, the Z-phase masking function is re-enabled, lasting for four periods before becoming inactive. During the Z-phase masking period, ZIN input is ignored, that is, the revolution counting unit does not count, and the position counting unit is not cleared. Any subsequent ZIN input will be processed normally.

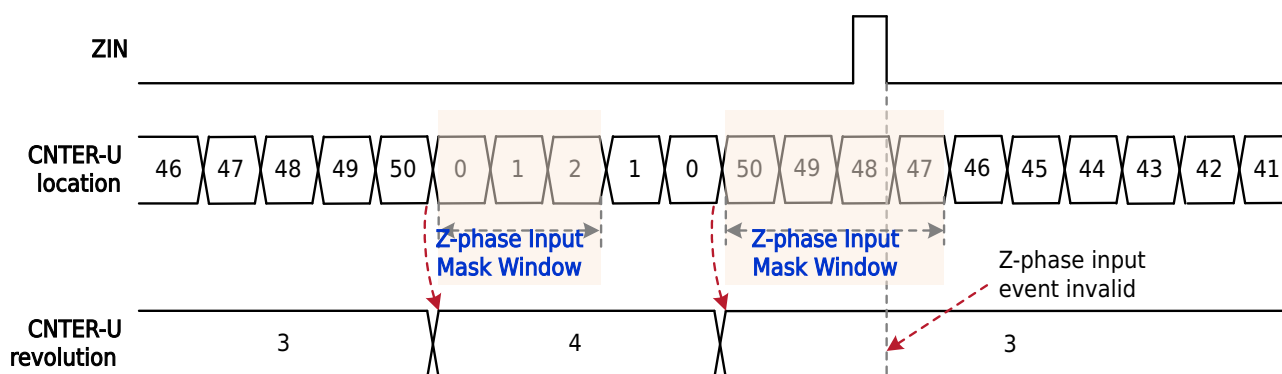


Figure 21-33 Revolution Counting Mode-Mixed Count Z-Phase Masking Action Instances 2

### 21.3.14 Periodic Interval Response

Timer6 features two Special Compare Reference Value Registers (SCMAR and SCMBR), which, upon compare match, can respectively generate Special Compare Match Interrupt A and Special Compare Match Interrupt B signals to the INTTC, thereby triggering corresponding interrupts. Simultaneously, Special Compare Match Event A and Event B signals can be output to coordinate with other modules—commonly used for triggering operations such as ADC conversions.

These interrupts and event request signals can be configured to become active only once every few periods, thus realizing periodic interval response. This functionality is enabled by setting the VPERR.PCNTE and VPERR.SPPERIA/B bits. Set the VPERR.PCNTS bit to specify how many period the request signal is valid. In other period, even if the count value is equal to the value of the special comparison reference value register SCMAR or SCMBR, no valid request signal will be output.

When this feature is enabled, period match interrupts and events in all waveform modes will only be triggered during active periods of special comparison match interrupts and events output (i.e., when STFLR.VPERNUM=0 in the diagram below). The diagram below illustrates the operation of active request signals at periodic intervals.

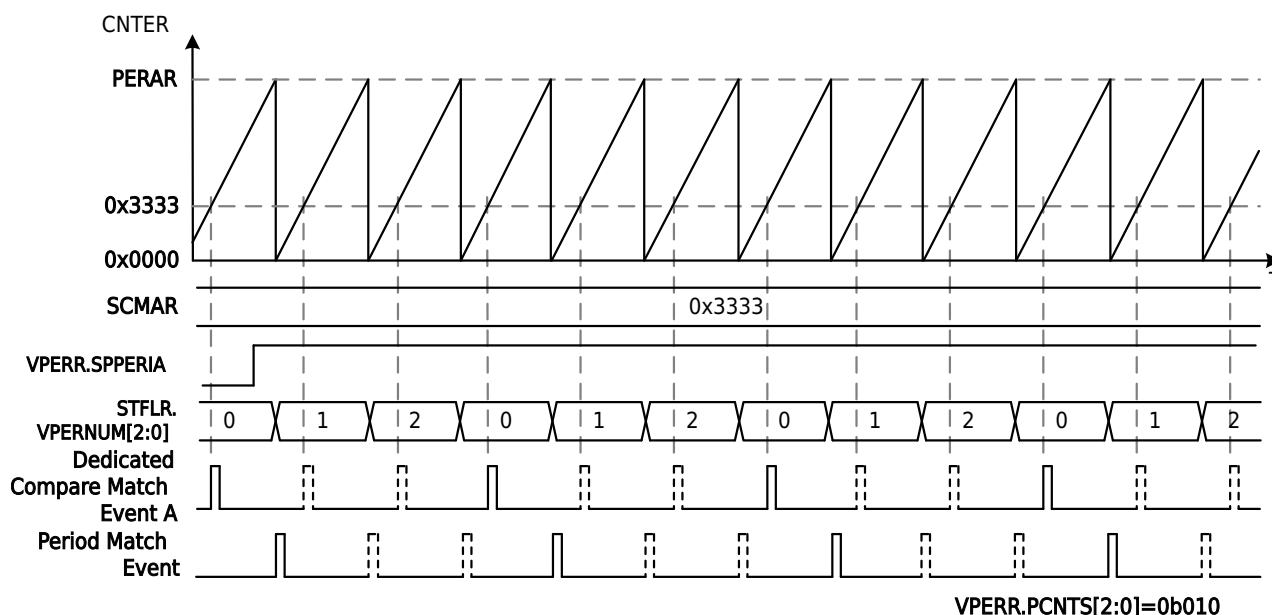


Figure 21-34 Periodic Interval Active Request Signal Behavior

### 21.3.15 EMB Control

Timer6 provides protection control for port output states, allowing the port states to be forced into a predefined safe state when an abnormal condition occurs. All units share 4 common port output control interfaces. Each unit can select the EMB event connection interfaces via the Port Control Register (PCNA<B>.EMBSA<B>), which connects to a group of EMB events from the EMB module. The abnormal conditions enabled on the interface can be configured from the EMB side (refer to the EMB chapter). When an abnormal condition is detected on these interfaces, control of the general-purpose PWM output can be performed accordingly.

During normal operation, if an EMB event is detected from the EMB module, the port output state can be switched to a predefined state. When an EMB abnormal event occurs, the PWM output port can be forced into one of the following output states: high-impedance, output low-level, or output high-level, depending on the configuration of PCNA<B>.EMBCA<B>. For example, if PCNAR.EMBCA=0b01, and an



EMB event occurs during normal output on the TMR6\_<t>\_PWMA port, the output on this port will switch to a high-impedance state.

After the EMB event is invalid (i.e., the abnormal event connected from the EMB module to Timer6 disappears and the signal returns to a normal level), the PWM port output can automatically resume normal operation. At this point, the normal output of PWM can be selected through the Port Control Register (PCNA<B>R.EMBRA<B>=0b00; this is referred to as One Shot release) or restore the normal output of the PWM after counting to the next overflow point (PCNA<B>R.EMBRA<B>=0b01, 0b10, or 0b11, this is referred to as Cycle By Cycle release).

### 21.3.16 PTO Function

By setting TMR6\_<t>\_PWMA and TMR6\_<t>\_PWMB as AB-phase outputs and using any of the TMR6\_TRI-GA-D as the Z-phase output, Timer6 can implement the PTO ABZ three-phase output.

The PTO algorithm period is set via TMR6\_PERAR. The next period's TMR6\_PTNUMR and TMR6\_PTPERR write-in time points are configured by setting GCMAR/GCMBR. With ICONR.INTENA/INTENB set to 1, the counting mode is set to sawtooth wave mode (GCONR.MODE=0), and the counting direction is set to up (GCMONR.DIR=1). TMR6\_PTCTLR.PTOENA controls the entry of Timer6 into the PTO mode. The three enable switches, TMR6\_PTCTLR.AOUTENA, TMR6\_PTCTLR.BOUTENA, and TMR6\_PTCTLR.ZOUTENA, control the output of the ABZ phases.

#### 21.3.16.1 AB-phase Output

The output phase difference of the AB phase is related to the number of pulses in the algorithm period TMR6\_PTNUMR.Bit31.

#### Forward Rotation

When TMR6\_PTNUMR[31] is 0, it indicates forward rotation. The AB-phase outputs follow the period of PTAOPR[3:0] and PTBOPR[3:0], in the sequence of Bit0 -> Bit1 -> Bit2 -> Bit3 -> Bit0 -> Bit1...

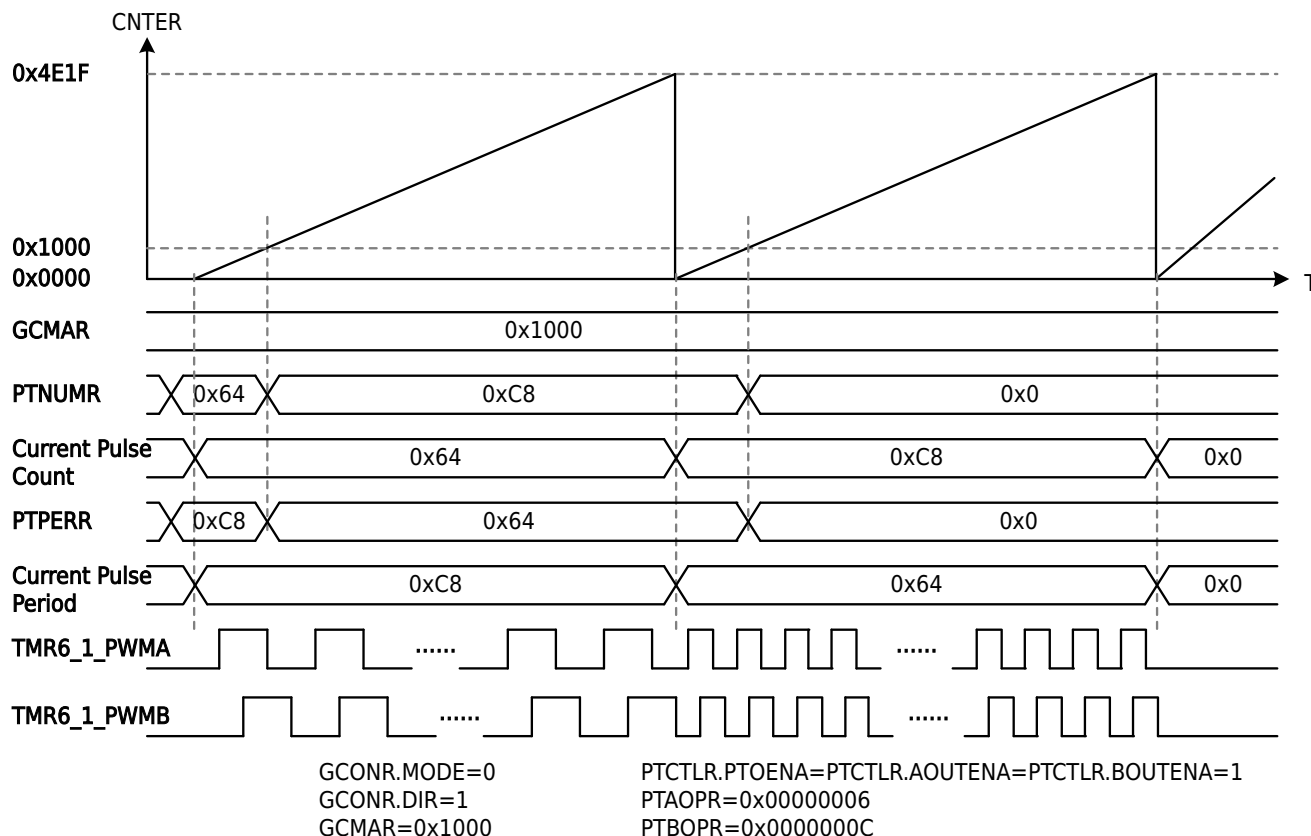


Figure 21-35 Forward Rotation AB Output Timing

### Reversal Rotation

When TMR6\_PTNUMR[31] is 1, it indicates reverse rotation. The AB-phase outputs follow the period of PTAOPR[3:0] and PTBOPR[3:0], in the sequence of Bit3 -> Bit2 -> Bit1 -> Bit0 -> Bit3 -> Bit2...

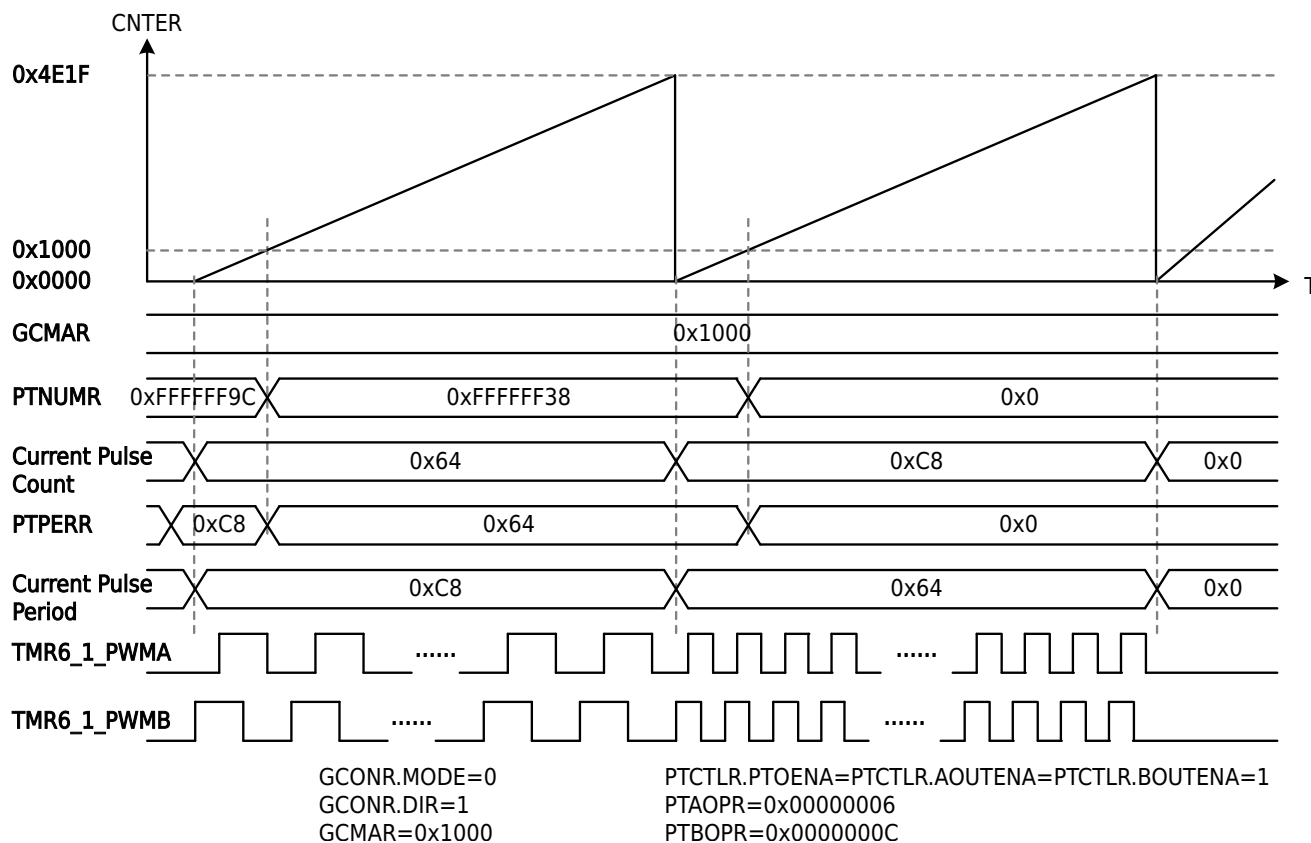


Figure 21-36 Reverse Rotation AB Output Timing

## Forward to Reverse Rotation

At the end of the current algorithm period, if the rotation direction reverses in the next period, when switching from forward to a reverse rotation, the AB-phase outputs follow the period of PTAOPR[3:0] and PTBOPR[3:0] in the sequence of Bit0 -> Bit1 -> Bit2 -> Bit3 -> Bit3 -> Bit2 -> Bit1 -> Bit0... Similarly, when transitioning from reverse to forward rotation, the AB-phase outputs follow the sequence of PTAOPR[3:0] and PTBOPR[3:0] in Bit3 -> Bit2 -> Bit1 -> Bit0 -> Bit0 -> Bit1 -> Bit2 -> Bit3...

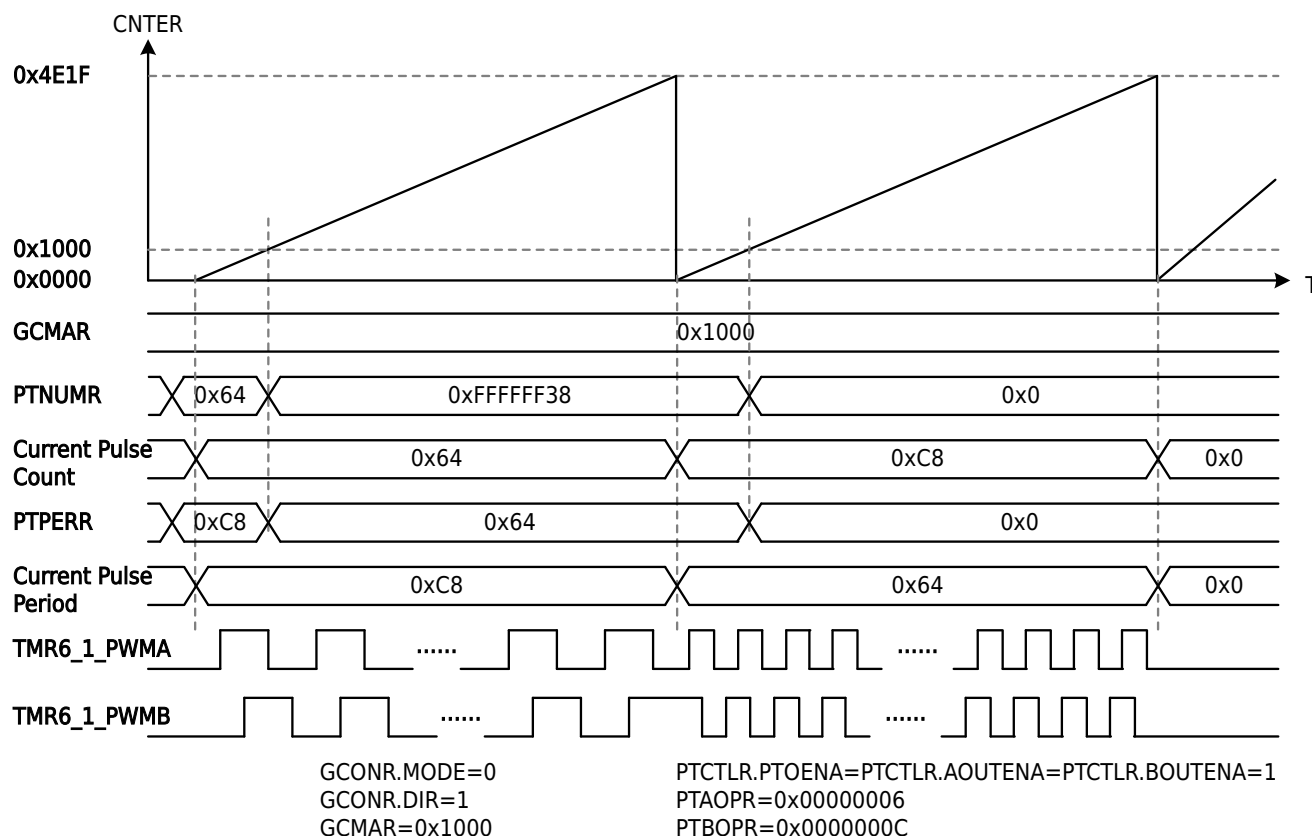
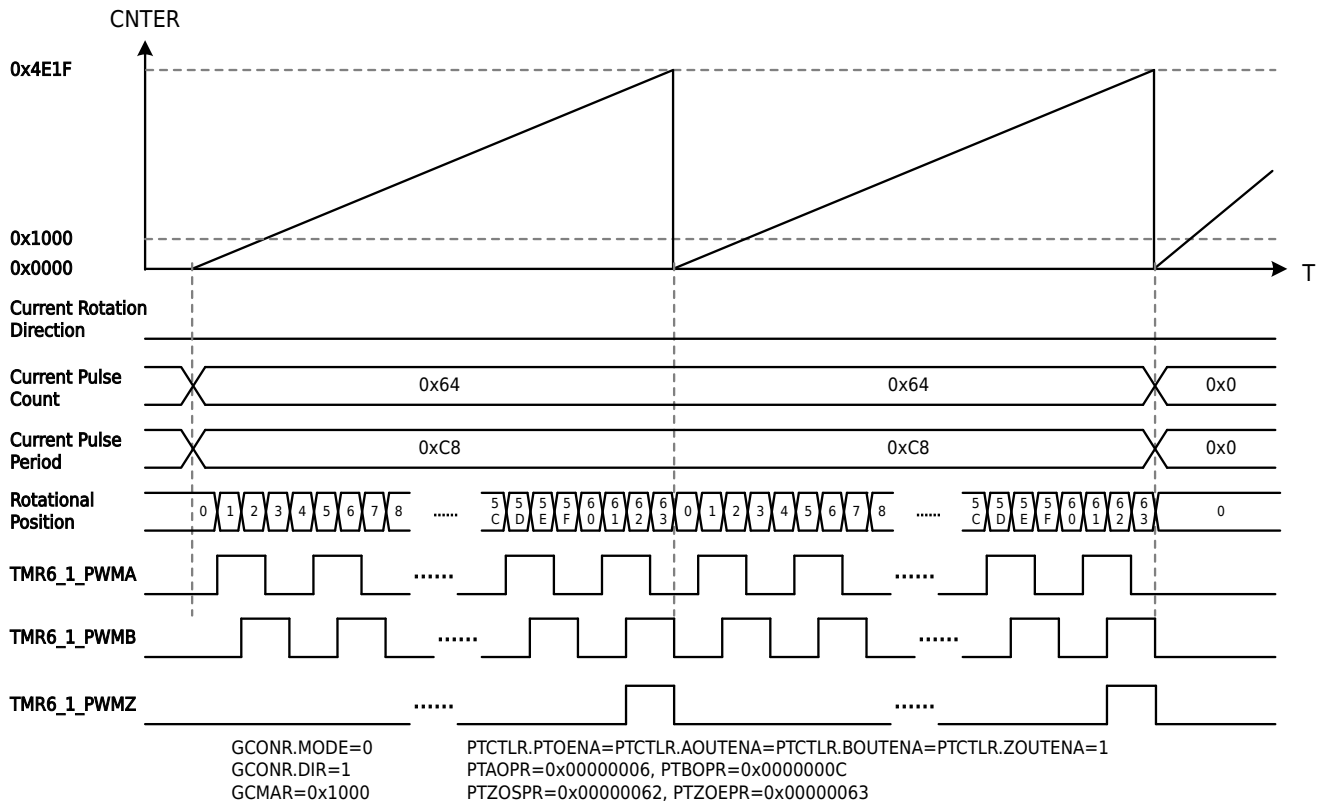


Figure 21-37 Forward to Reverse Rotation

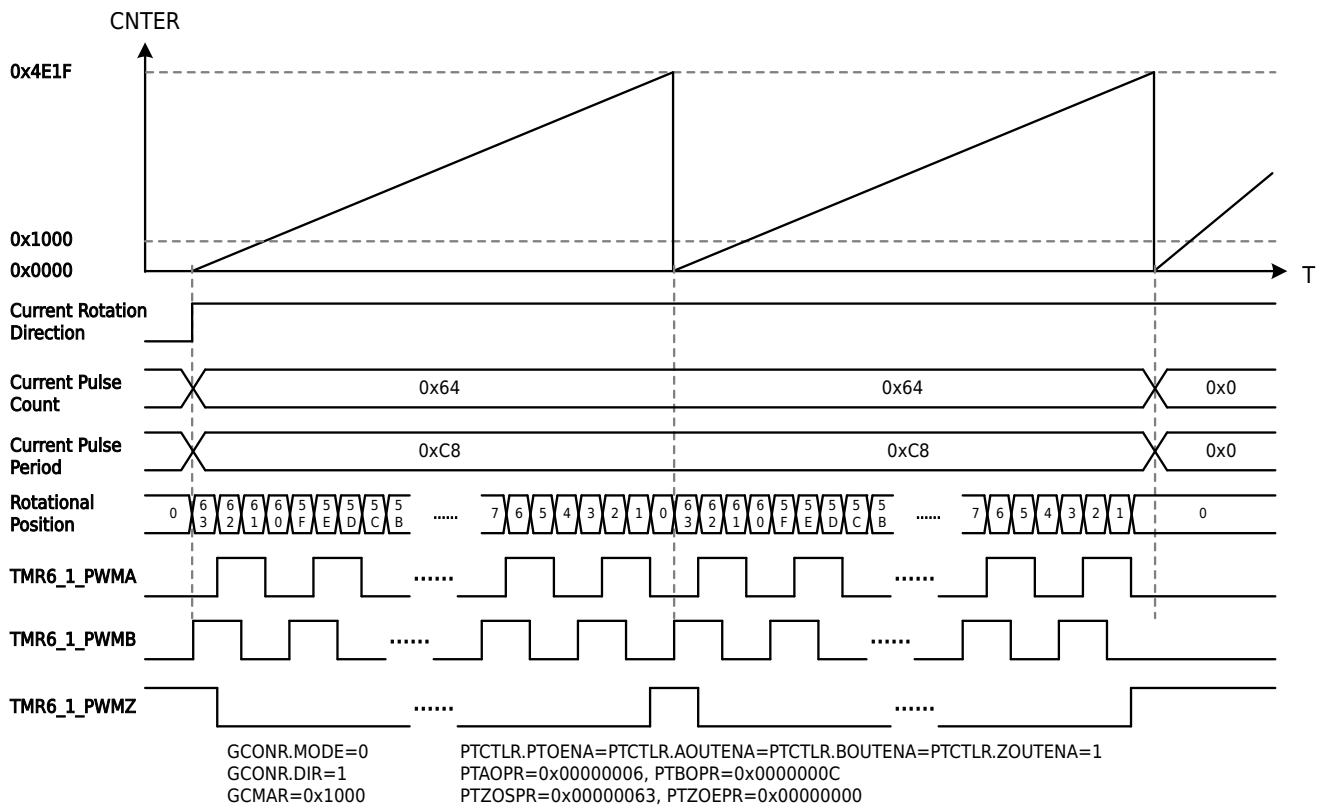
### 21.3.16.2 Z-Phase Output

#### Z-Phase Output (Non-Compressed)

The TMR6\_PTZPER register sets the number of pulses per full rotation. When the position counter reaches TMR6\_PTZPER-1, it automatically resets to zero on the next clock period. The Z-phase output is position-dependent. During forward rotation, when the position counter reaches TMR6\_PTZOSPR, the Z output goes high; when it reaches TMR6\_PTZOEPR, the Z output goes low, as shown in the figure below. In theory, TMR6\_PTZOSPR and TMR6\_PTZOEPR must be less than TMR6\_PTZPERR; otherwise, the Z-phase output will be disabled.

Figure 21-38 Forward Rotation Z Output ( $PTZOSPR < PTZOEPR$ )

During reverse rotation, when the position counter reaches TMR6\_PTZOEPR, the Z output goes high; when it reaches TMR6\_PTZOSPR, the Z output goes low, as illustrated below.

Figure 21-39 Reverse Rotation Z Output ( $PTZOSPR > PTZOEPR$ )

If the current algorithm period is forward rotation, and the next period is reverse rotation, and the last pulse position of the current algorithm period equals PTZOSPR, the Z-phase output behaves as shown below.

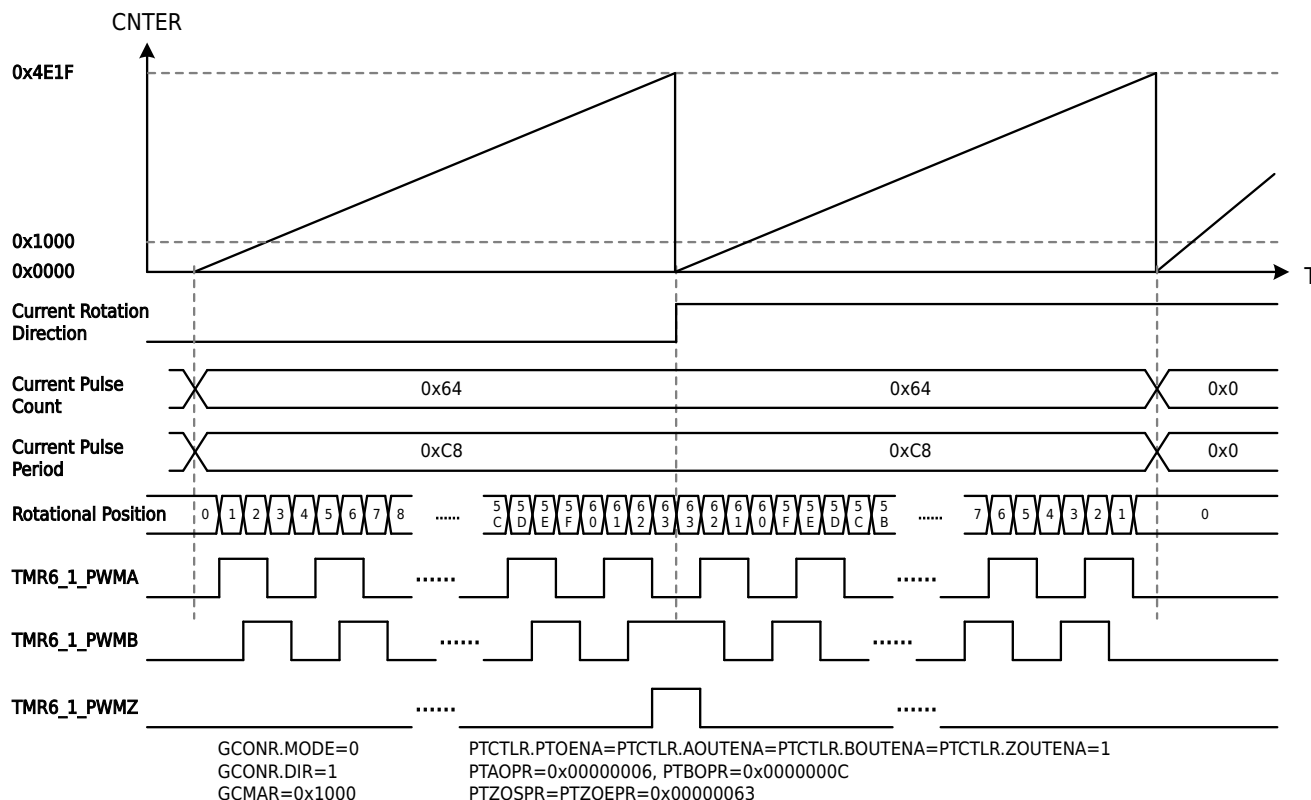


Figure 21-40 Forward-to-Reverse Rotation Z Output (PTZOSPR=PTZOEPR)

### Z-Phase Output (Compressed)

Z-phase output supports CLK-level compression. Once the Z output position is specified, the TMR6\_PTZCMPBR.ZCMPB register can be configured to define the compression range. Enable CLK compression via the TMR6\_PTZCMPBR.ZCMPBENA register.

Since the TMR6\_PTPER value is dynamic, the configuration of TMR6\_PTZCMPBR.ZCMPB must satisfy the following condition:

When PTZOSPR equals PTZOEPR, the following constraint must be met:  $\text{TMR6\_PTZCMPBR.ZCMPB} < (\text{TMR6\_PTPER}_{\min}/2)$ , otherwise the compression function will not take effect. When  $\text{PTZOSPR} \neq \text{PTZOEPR}$ , this constraint does not apply.

The compression function is illustrated as follows:

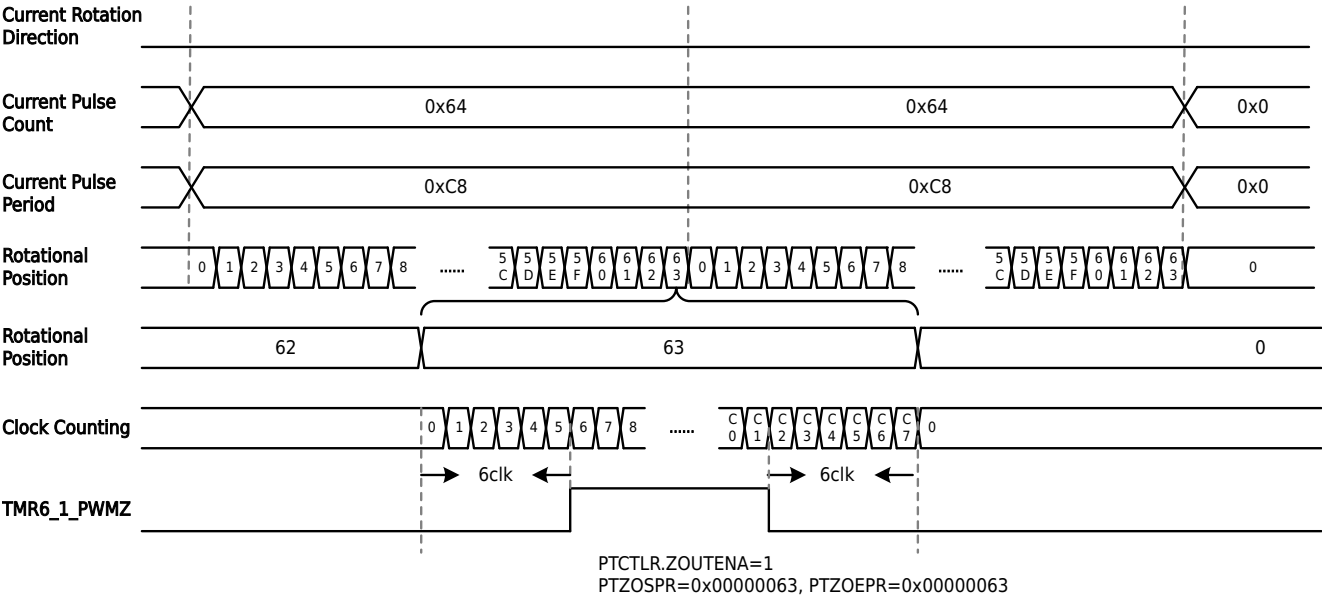


Figure 21-41 Forward Rotation Z Output Compressed

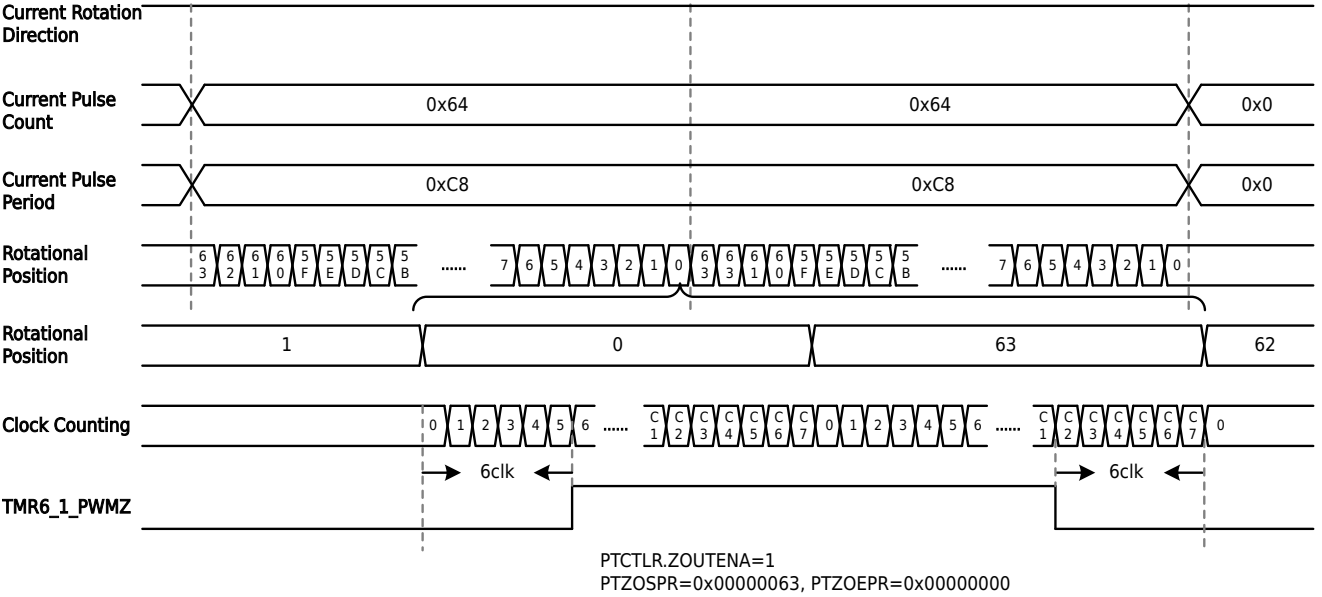


Figure 21-42 Reverse Z-phase Output Compressed

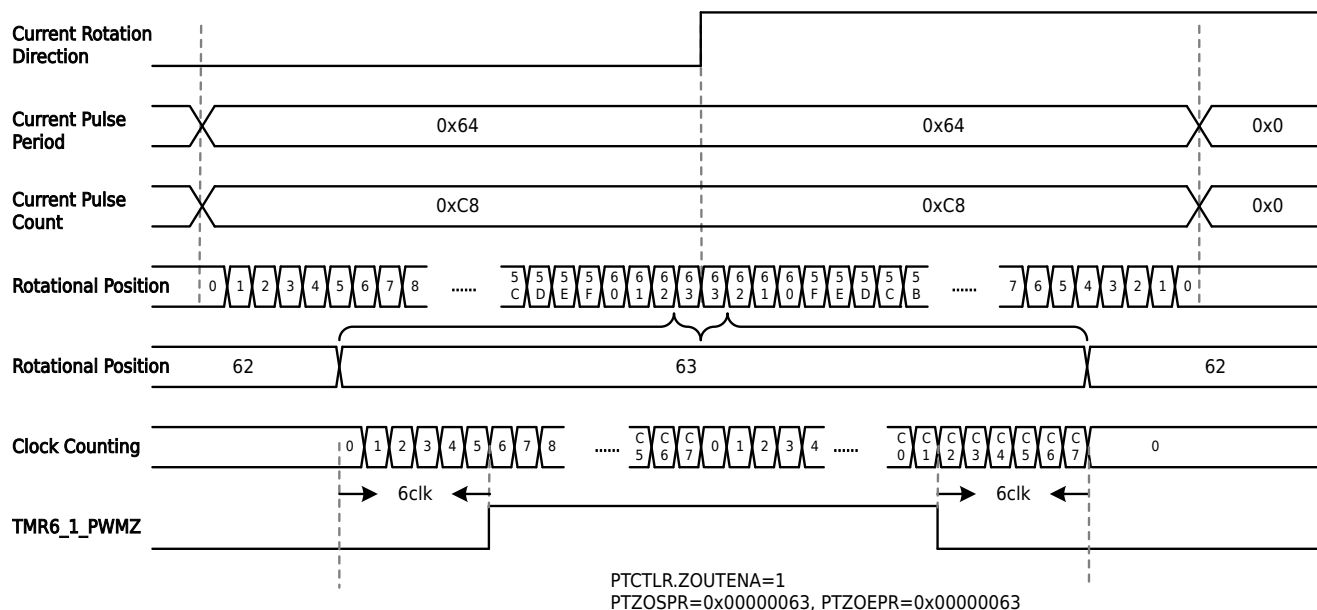


Figure 21-43 Forward and Reverse Z-phase Output Compressed

### 21.3.16.3 Remainder Processing

Since pulse period=algorithm period/number of pulses, the calculation of the pulse period in software may produce a remainder. If this remainder is discarded, the AB-phase signals will maintain their last output state until the end of the algorithm period. As illustrated below, for example: Algorithm period (0x 41EF) /Number of pulses (0x 12C) =Pulse period (0x 42), with a remainder of 0x C8. If the remainder is not compensated, the AB-phase signals will remain unchanged during the last 200 clock periods of the algorithm period.

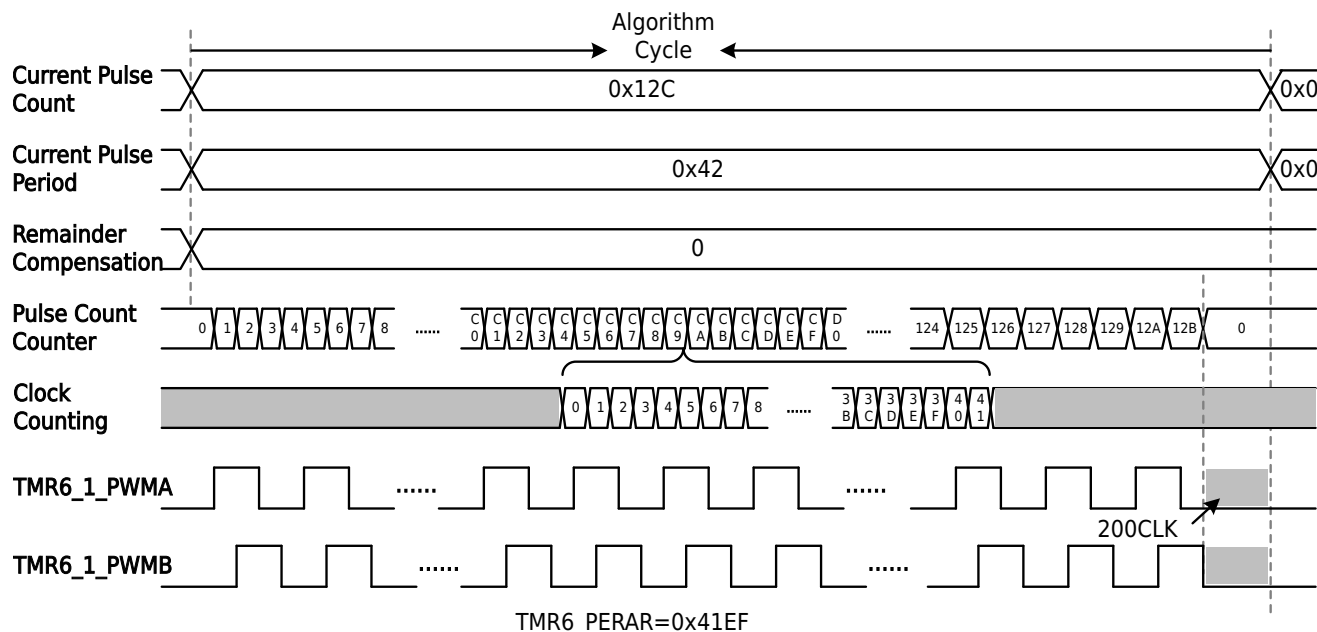


Figure 21-44 AB Output (Without Remainder Compensation)

To address this, the TMR6\_PTRMDR register can be used to pass the remainder to hardware. In this case, the first 200 pulse periods are set to 0x 42, and the last 100 pulse periods are set to 0x 41. This actively distributes the remainder across the first 200 pulses, each pulse compensates for one clock cycle, and there will be no 200 clock cycle gap period after the AB output stops at the end of the algorithm cycle as shown in the above figure.

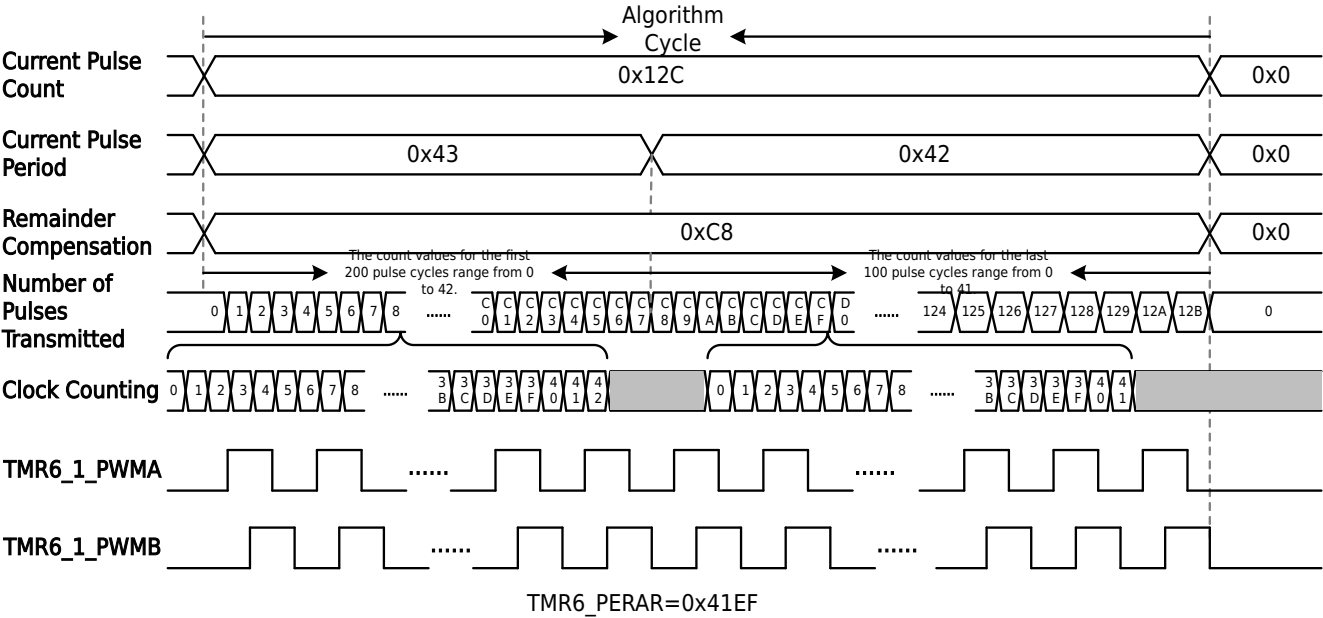


Figure 21-45 AB Output (With Remainder Compensation)

21.3.16.4 Position Memory

PTPCPR and PTPCTR are read-only registers that store position information, ABZ port levels, and rotation direction. Before performing a reset or any operation that may result in the loss of PTO information, the contents of PTPCPR and PTPCTR should be read and saved. After the operation is completed, the saved data can be written back to the corresponding PTPSPR and PTPSTR registers to restore the position and ABZ port level information to their pre-reset state. Once this restoration is complete, the PTO function can be re-enabled using the previous configuration. By setting TMR6\_GCONR.START to 1, the ABZ output and position tracking will resume based on the restored data.



### 21.3.17 Function Summary Table

The following table summarizes the main features of Timer6 in Sawtooth and Triangle A/B wave modes.

**Table 21-3 Function Compare Table in Different Modes**

| PWM Output Function      |                      |                         |                          | Sawtooth Wave | Triangle Wave | Related Main Registers                                         |
|--------------------------|----------------------|-------------------------|--------------------------|---------------|---------------|----------------------------------------------------------------|
| Independent PWM Output   | Port State Control   | On Start                |                          | Support       | Support       | PCNA<B>R.STACA<B>                                              |
|                          |                      | On Stop                 |                          | Support       | Support       | PCNA<B>R.STPCA<B>                                              |
|                          |                      | On Overflow             |                          | Support       | Support       | PCNA<B>R.OVFCA<B>                                              |
|                          |                      | On Underflow            |                          | Support       | Support       | PCNA<B>R.UDFCA<B>                                              |
|                          |                      | Up Counting             |                          | Support       | Support       | PCNA<B>R.CMA<B>UCA<B>                                          |
|                          |                      | Down Counting           |                          | Support       | Support       | PCNA<B>R.CMA<B>DCA<B>                                          |
|                          | Buffer Transfer      | Period Reference Value  | Single Buffer            | Support       | Support       | PERBR->PERAR                                                   |
|                          |                      |                         | Double Buffer            | Support       | Support       | PERCR->PERBR<br>PERBR->PERAR                                   |
|                          |                      | Compare Reference Value | Single Buffer            | Support       | Support       | GCMDB->GCMDBR<br>GCMCR->GCMAR                                  |
|                          |                      |                         | Double Buffer            | Support       | Support       | GCMFR->GCMDBR<br>GCMER->GCMCR<br>GCMDB->GCMDBR<br>GCMCR->GCMAR |
|                          | EMB                  |                         |                          | Support       | Support       | PCNA<B>R.EMBCA<B>                                              |
| Complementary PWM Output | Port State Control   | On Start                |                          | Support       | Support       | PCNA<B>R.STACA<B>                                              |
|                          |                      | On Stop                 |                          | Support       | Support       | PCNA<B>R.STPCA<B>                                              |
|                          |                      | On Overflow             |                          | Support       | Support       | PCNA<B>R.OVFCA<B>                                              |
|                          |                      | On Underflow            |                          | Support       | Support       | PCNA<B>R.UDFCA<B>                                              |
|                          |                      | Up Counting             |                          | Support       | Support       | PCNA<B>R.CMA<B>UCA<B>                                          |
|                          |                      | Down Counting           |                          | Support       | Support       | PCNA<B>R.CMA<B>DCA<B>                                          |
|                          | Buffer Transfer      | Period Reference Value  | Single Buffer            | Support       | Support       | PERBR->PERAR                                                   |
|                          |                      |                         | Double Buffer            | Support       | Support       | PERCR->PERBR<br>PERBR->PERAR                                   |
|                          |                      | Compare Reference Value | Single Buffer            | Support       | Support       | GCMDB->GCMDBR<br>GCMCR->GCMAR                                  |
|                          |                      |                         | Double Buffer            | Support       | Support       | GCMFR->GCMDBR<br>GCMER->GCMCR<br>GCMDB->GCMDBR<br>GCMCR->GCMAR |
|                          |                      | Dead Zone Reference     | Single Buffer            | Support       | Support       | DTUBR->DTUAR<br>DTDBR->DTDAR                                   |
|                          |                      |                         | Non-Dead-Time PWM Output |               |               | Support                                                        |
|                          | Dead-Time PWM Output |                         | Software-based Method    | Support       | Support       | GCMAR≠GCMDBR                                                   |
|                          |                      |                         | Hardware-based Method    | Not support   | Support       | GCMDBR=GCMAR-DTUAR<br>GCMDBR=GCMAR-DTDAR                       |
|                          | EMB                  |                         |                          | Support       | Support       | PCNA<B>R.EMBCA<B>                                              |

## 21.3.18 Interrupt and Events

### 21.3.18.1 Interrupt

Timer6 contains 6 general count compare match interrupts (including 2 capture input interrupts), 2 special count compare match interrupts, 2 count period match interrupts, and 1 dead time error interrupt.

#### Count Compare Match Interrupt

There are 6 general compare reference value registers (GCMAR-GCMFR), which can be compared with the count value to generate a comparison match. When the count compare matches, the STFLR.CMAF-STFLR.CMFF bits will be set to 1 respectively. At this time, if the corresponding bit in ICONR.INTENA-INTENF is set to 1 to enable interrupt, the corresponding interrupt request (TMR6\_<t>\_GCMA-F) will also be triggered.

When the capture input active condition selected by the hardware capture event selection register (HCPAR, HCPBR) occurs, the capture input action occurs. At this time, if the ICONR.INTENA or ICONR.INTENB bit is set to 1 to enable interrupt, the corresponding interrupt request (TMR6\_<t>\_GCMA-B) is triggered.

Two special compare reference value registers (SCMAR-SCMBR) can also be compared with the count value to generate a comparison match. When the count compare matches, the STFLR.CMSPAF-CMSPBF bits will be set to 1 respectively. At this time, if the corresponding bit in ICONR.INTENSAU<D> or ICONR.INTENSBUS<D> is set to 1 to enable interrupt, the corresponding interrupt request (TMR6\_<t>\_SCMA-B) will also be triggered.

#### Count Period Match Interrupt

When the sawtooth wave counts up to the overflow point, the sawtooth wave counts down to the underflow point, the triangle wave counts to the valley point, or the triangle wave counts to the peak point, the STFLR.OVFF or STFLR.UFFF bit will be set to 1. At this time, if the ICONR.INTENOVF or ICONR.INTENUDF bit is set to enable interrupt, the count period match interrupt (TMR6\_<t>\_GOVF and TMR6\_<t>\_GUDF) can be triggered at the corresponding time point.

#### Dead Time Error Interrupt

When the value of the dead time reference value register (DTU<D>AR) is loaded into the general compare reference value register (GCMBR), if it exceeds the period limit, a dead time error will occur, and the STFLR.DTEF bit will be set to 1. At this time, if the ICONR.INTENDTE bit is set to enable interrupts, the dead time error interrupt (TMR6\_<t>\_GDTE) will be triggered at this moment.

### 21.3.18.2 Events

During the clock counting process, if a period match event (overflow or underflow point of sawtooth wave, counting peak or valley point of triangle wave), general count comparison match event, or special count comparison match event occurs, a corresponding event output signal will be generated to select and trigger other modules, such as ADC, DMA, etc.

The following figure is an action example of general comparison match interrupt A-F&&event A-F (TMR6\_1\_GCMA-F), special comparison match interrupt A-B&&event A-B (TMR6\_1\_SCMA-B), and period match interrupt&&event (TMR6\_1\_GOVF) of unit 1.

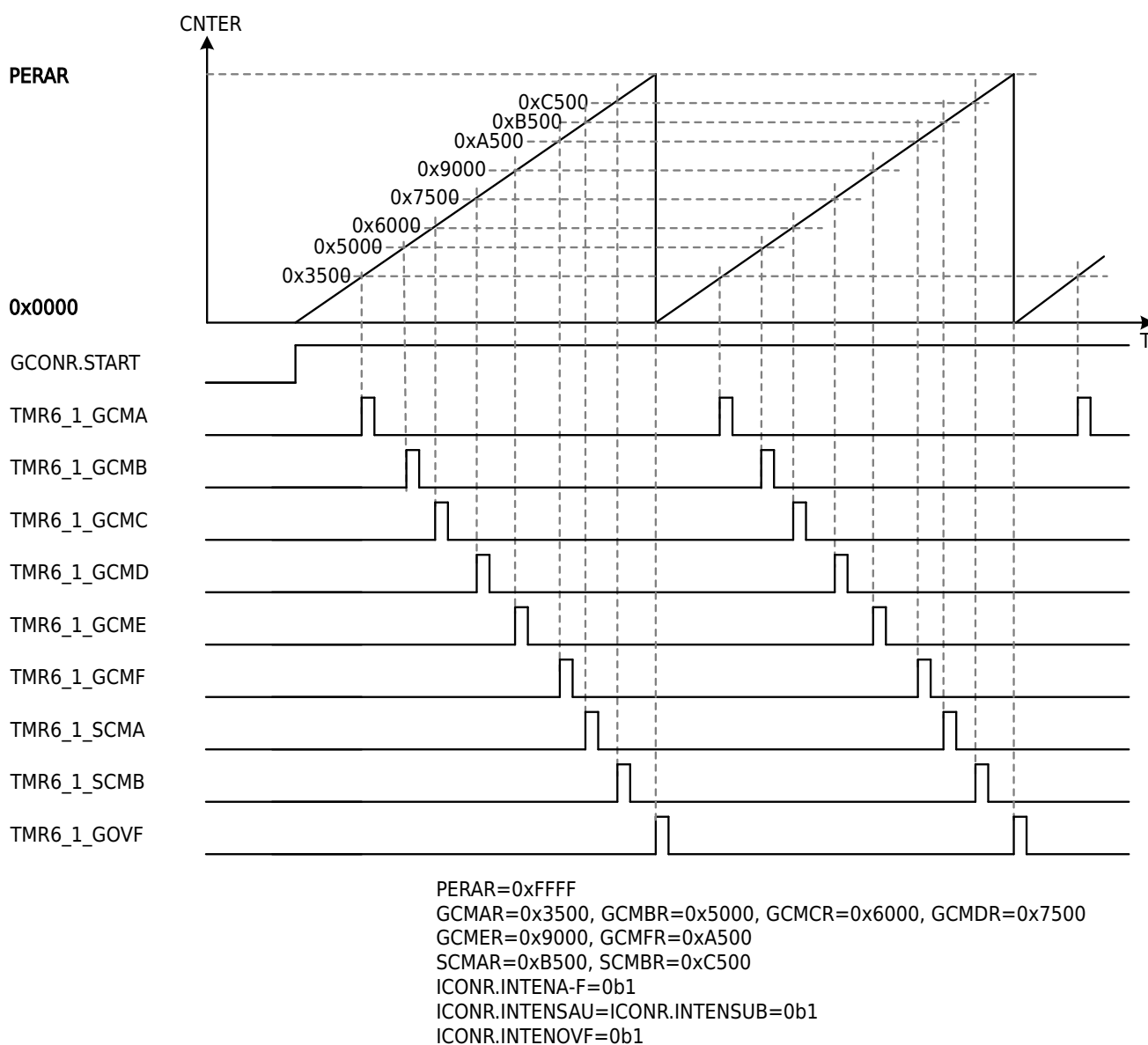


Figure 21-46 Interrupt & Event Output Example in Sawtooth Wave Mode

## 21.4 Operation Example

The following describes the basic settings of Timer6 related registers in several typical application cases for user reference.

### 21.4.1 Basic Counting and Interrupt Action

- Step1.** Set the general period reference value (PERAR);
- Step2.** Set the required compare reference value, including the general compare reference value (GCMAR-GCMFR), the special compare reference value (SCMAR-SCMBR), etc.;
- Step3.** Set the required interrupt enable bit, including the count overflow interrupt (ICONR.INTENOVF), the count underflow interrupt (ICONR.INTENUDF), the count match interrupt (ICONR.INTENA-F, ICONR.INTENSAU, ICONR.INTENSAD, ICONR.INTENSBU, ICONR.INTENSBD), etc.;
- Step4.** Set the internal count clock division (GCONR.CKDIV);
- Step5.** Set the waveform mode (GCONR.MODE);
- Step6.** Set the counting direction (only required when the sawtooth wave mode GCONR.MODE=0);
- Step7.** Start the counter (GCONR.START=1).

### 21.4.2 Compare Output and Interrupt Action

- Step1.** Set the general period reference value (PERAR);
- Step2.** Set the compare reference of each channel, including the general compare reference value A (GCMAR) and the general compare reference value B (GCMBR);
- Step3.** Set the required interrupt enable bits, including count overflow interrupt (ICONR.INTENOVF), count underflow interrupt (ICONR.INTENUDF), count match interrupt (ICONR.INTENA-B), etc.;
- Step4.** Set the port output status for each channel in different counting states (refer to Bit17-Bit0 of PCNAR or PCNBR for related control);
- Step5.** Set internal count clock frequency division (GCONR.CKDIV);
- Step6.** Set the waveform mode (GCONR.MODE);
- Step7.** Set the counting direction (only required in sawtooth wave mode when GCONR.MODE=0);
- Step8.** Set the comparison output mode of each channel (PCNAR.CAPMDA=0, PCNBR.CAPMDB=0);
- Step9.** Set each channel output enable (PCNAR.OUTENA=1, PCNBR.OUTENB=1);
- Step10.** Start the counter (GCONR.START=1).

### 21.4.3 Capture Input and Interrupt Action

- Step1.** Set the general period reference value (PERAR);
- Step2.** Set the required interrupt enable bits, including the counter overflow interrupt (ICONR.INTENOVF), counter underflow interrupt (ICONR.INTENUDF), and capture input interrupts (ICONR.INTENA-B), etc.;
- Step3.** Set the capture input external conditions for each channel (refer to all active control bits in HCPAR or HCPBR. These active control bits are independent of each other, and multiple conditions can be selected simultaneously to serve as capture input conditions for a single channel);
- Step4.** Set the internal counter clock division (GCONR.CKDIV);
- Step5.** Select the waveform mode (GCONR.MODE);
- Step6.** Set the counting direction (only required when in sawtooth wave mode GCONR.MODE=0);
- Step7.** Set the capture input mode (PCNAR.CAPMDA=1 and PCNBR.CAPMDB=1);
- Step8.** Start the counter (GCONR.START=1);

- Step9.** Wait for the capture input condition to occur, then read the capture input value from the corresponding channel (GCMAR or GCMBR), or wait for the corresponding interrupt to be triggered.

#### 21.4.4 Buffer Transmission Action (Period Reference Value)

- Step1.** Set the required general period reference value (PERAR, PERBR, PERCR);
- Step2.** Set the single or double buffer transmission mode (BCONR.BSEP);
- Step3.** Set the buffer transmission time point (BCONR.BTRUP, BCONR.BTRDP, these two control bits are independent of each other and can be selected at the same time as the buffer transmission time point) (this setting is only valid in triangle wave mode and invalid in sawtooth wave mode);
- Step4.** Set the internal count clock frequency division (GCONR.CKDIV);
- Step5.** Set the waveform mode (GCONR.MODE);
- Step6.** Set the counting direction (only required in sawtooth wave mode when GCONR.MODE=0);
- Step7.** Set the buffer function as valid (BCONR.BENP=1).
- Step8.** Start the counter (GCONR.START=1);
- Step9.** Wait for the corresponding buffer transmission time point, and the buffer action occurs (PERBR->PERAR (when BCONR.BSEP=0), PERCR->PERBR->PERAR (when BCONR.BSEP=1) ).

#### 21.4.5 Buffer Transmission Action (General Compare Reference Value)

- Step1.** Set the required general compare reference value (GCMAR, GCMCR, GCMER, GCMBR, GCMDR, GCMFR);
- Step2.** Set the single or double buffer transmission mode for each channel (BCONR.BSEA, BCONR.BSEB);
- Step3.** Set the buffer transmission time point of each channel (BCONR.BTRUA, BCONR.BTRDA, BCONR.BTRUB, BCONR.BTRDB, the two control bits of each channel are independent of each other and can be selected at the same time as the buffer transmission time point) (this setting is only valid in triangle wave mode and invalid in sawtooth wave mode);
- Step4.** Set the internal count clock division (GCONR.CKDIV);
- Step5.** Set the waveform mode (GCONR.MODE);
- Step6.** Set the counting direction (only required in sawtooth wave mode when GCONR.MODE=0);
- Step7.** Set the buffer function of each channel to be valid (BCONR.BENA=1, BCONR.BENB=1);
- Step8.** Start the counter (GCONR.START=1);
- Step9.** Wait for the corresponding buffer transmission time point set for each channel, and the buffer action occurs (GCMCR->GCMAR (when BCONR.BSEA=0), GCMER->GCMCR->GCMAR (when BCONR.BSEA=1), GCMDR->GCMBR (when BCONR.BSEB=0), GCMFR->GCMDR->GCMBR (when BCONR.BSEB=1) ).

#### 21.4.6 Buffer Transmission Action (Special Compare Reference Value)

- Step1.** Set the required special compare reference value (SCMAR, SCMCR, SCMER, SCMBR, SCMDR, SCMFR);
- Step2.** Set the single or double buffer transmission mode for each channel (BCONR.BSESPA, BCONR.BSESPB);
- Step3.** Set the buffer transmission time point of each channel (BCONR.BTRUSPA, BCONR.BTRDSPA, BCONR.BTRUSPB, BCONR.BTRDSPB, the two control bits of each channel are independent of each other and can be selected at the same time as the buffer transmission time point) (this setting is only valid in triangle wave mode and invalid in sawtooth wave mode);

- Step4.** Set the internal count clock division (GCONR.CKDIV);
- Step5.** Set the waveform mode (GCONR.MODE);
- Step6.** Set the counting direction (only required in sawtooth wave mode when GCONR.MODE=0).
- Step7.** Set the buffer function of each channel to be valid (BCONR.BENSPA=1, BCONR.BENSPB=1);
- Step8.** Start the counter (GCONR.START=1);
- Step9.** Wait for the corresponding buffer transmission time point set for each channel, and the buffer action occurs (SCMCR->SCMAR (when BCONR.BESPA=0), SCMER->SCMCR->SCMAR (when BCONR.BESPA=1), SCMDR->SCMBR (when BCONR.BESPB=0), SCMFR->SCMDR->SCMBR (when BCONR.BESPB=1) ).

#### 21.4.7 Buffer Transmission Action (Dead Zone Reference Value)

- Step1.** Set the required dead time reference value(DTUAR, DTUBR, DTDAR, DTDBR);
- Step2.** Set the buffer transmission time point (DCONR.DTBTRU, DCONR.DTBTRD, these two control bits are independent of each other and can be selected at the same time as the buffer transmission time point) (this setting is only valid in triangle wave mode and invalid in sawtooth wave mode);
- Step3.** Set the internal count clock division (GCONR.CKDIV);
- Step4.** Set the waveform mode (GCONR.MODE);
- Step5.** Set the counting direction (only required in sawtooth wave mode when GCONR.MODE=0);
- Step6.** Set the buffer function to be enabled (DCONR.DTBENU=1, DCONR.DTBEND=1);
- Step7.** Set the hardware dead zone function to be enabled (DCONR.DTCEN=1);
- Step8.** Start the counter (GCONR.START=1);
- Step9.** Wait for the corresponding buffer transmission time point and the buffer action to occur (DTUBR->DTUAR, DTDBR->DTDAR).

#### 21.4.8 Synchronization Start Action (Software Mode)

- Step1.** Refer to steps 1 to 6 in the "Basic Counting and Interrupt Action" section to set each unit that needs to be started synchronously;
- Step2.** Synchronously start the counter (set the corresponding bit of the SSTAR register to 1, each unit corresponds to a register bit).

#### 21.4.9 Synchronization Start Action (Hardware Mode)

- Step1.** Set the general oeriod reference value (PERAR);
- Step2.** Set the required compare reference value, including the general compare reference value (GCMAR-GCMFR), the special compare reference value (SCMAR-SCMBR), etc.;
- Step3.** Set the required interrupt enable bit, including the count overflow interrupt (ICONR.INTENOVF), the count underflow interrupt (ICONR.INTENUF), the count match interrupt (ICONR.INTENA-F, ICONR.INTENSAU, ICONR.INTENSAD, ICONR.INTENSB, ICONR.INTENSBD), etc.;
- Step4.** Set the hardware start condition (select through HSTAR.HSTAx, x=8-31);
- Step5.** Set the hardware start enable (HSTAR.STAS=1);
- Step6.** Repeat the above steps 1-5 to set each unit that needs to be started synchronously (the setting of step 4 must be consistent in each unit that needs to be started synchronously);
- Step7.** Wait for the set trigger event to occur and confirm that the counters of each unit are started synchronously.



### 21.4.10 Quadrature Encoding Count Action (2-phase)

- Step1.** Set the General Period Reference Value (PERAR);
- Step2.** Set the required compare reference value, including the general compare reference value (GCMAR-GCMFR), the special compare reference value (SCMAR-SCMBR), etc.;
- Step3.** Set the required interrupt enable bit, including the count overflow interrupt (ICONR.INTENOVF), the count underflow interrupt (ICONR.INTENUDF), the count match interrupt (ICONR.INTENA-F, ICONR.INTENSAU, ICONR.INTENSAD, ICONR.INTENSBU, ICONR.INTENSBD), etc.;
- Step4.** Set the required hardware up-counting condition (selected by HCUPR.HCUPx, x=0-7);
- Step5.** Set the required hardware down-counting condition (selected by HCDOR.HCDOx, x=0-7);
- Step6.** Start the counter (GCONR.START=1);
- Step7.** Wait for the set quadrature encoding counting event to occur and confirm that the counter is counting normally.

### 21.4.11 Quadrature Encoding Count Action (3-phase)

- Step1.** Refer to steps 1 to 5 in the "Quadrature Encoding Count Action (2-phase)" section to set the position counting unit;
- Step2.** Set the hardware clearing condition of the position counting unit (selected by HCLRR.HCLRx, x=16-31);
- Step3.** Set the hardware clearing enable of the position counting unit (HCLRR.CLES=1);
- Step4.** Set the general period reference value (PERAR) of the revolution counting unit;
- Step5.** Set the compare reference value of the revolution counting unit, including the general compare reference value (GCMAR-GCMFR), the special compare reference value (SCMAR-SCMBR), etc.;
- Step6.** Set the interrupt enable bits required by the revolution counting unit, including count overflow interrupt (ICONR.INTENOVF), count underflow interrupt (ICONR.INTENUDF), count match interrupt (ICONR.INTENA-F, ICONR.INTENSAU, ICONR.INTENSAD, ICONR.INTENSBU, ICONR.INTENSBD), etc.;
- Step7.** Configure hardware count-up condition 1 for the revolution counter unit (ZIN-phase input) (Select via HCUPR.HCUPx, x=16-31. This setting event must be consistent with the event configured for the position counter unit in Step 2);
- Step8.** Configure hardware count-up condition 2 (The overflow event input of the position counting unit) (Select internal hardware trigger event 0 via HCUPR.HCUP8);
- Step9.** Set the hardware count-down condition (The underflow event input of the position counting unit) (Select internal hardware trigger event 1 via HCDOR.HCDO9);
- Step10.** Set the trigger source number in TMR6\_TRGSEL0 to the count overflow event of the position counting unit (for the register content, see the "Register-Timer6 Hardware Trigger Event Selection Register (TMR6\_TRGSELx) (x=0-3) " in the "Automatic Operation System (AOS) " chapter, and for the overflow event number, see the "Interrupt Controller (INTC) " chapter);
- Step11.** Set the trigger source number in TMR6\_TRGSEL1 to the count underflow event of the position counting unit (for the register content, see the "Register-Timer6 Hardware Trigger Event Selection Register (TMR6\_TRGSELx) (x=0-3) " in the "Automatic Operation System (AOS) " chapter, and for the overflow event number, see the "Interrupt Controller (INTC) " chapter);
- Step12.** Start the revolution counting unit counter (GCONR.START=1);
- Step13.** Start the position counting unit counter (GCONR.START=1);
- Step14.** Wait for the set AIN, BIN, ZIN phase counting events to occur and confirm that the counter is counting normally.

### 21.4.12 Single Channel PWM Output

Refer to the settings of steps 1 to 10 in the "[compare Output and Interrupt Action](#)" section (the output status of the two PWM channels TMR6\_<t>\_PWMA and TMR6\_<t>\_PWMB inside each unit can be set independently to form two independent single-channel PWM outputs).

### 21.4.13 Complementary PWM Output (Software Dead Zone)

- Step1.** Set the general period reference value (PERAR);
- Step2.** Set the general compare reference value A (GCMAR) and general compare reference value B (GCMBR);
- Step3.** Set the required interrupt enable bits, including count overflow interrupt (ICONR.INTENOVF), count underflow interrupt (ICONR.INTENUF), count match interrupt (ICONR.INTENA-B), etc.;
- Step4.** Set the port output states for different counter conditions (refer to Bit17-Bit0 of PCNAR and PCNBR for control settings. In conjunction with the values set in GCMAR and GCMBR, ensure that a complementary dead time is formed between the two PWM outputs);
- Step5.** Set the internal count clock division (GCONR.CKDIV);
- Step6.** Set the waveform mode to triangle wave mode (GCONR.MODE=1);
- Step7.** Set the comparison output mode (PCNAR.CAPMDA=0, PCNBR.CAPMDB=0);
- Step8.** Enable the outputs (PCNAR.OUTENA=1, PCNBR.OUTENB=1);
- Step9.** Start the counter (GCONR.START=1).

### 21.4.14 Complementary PWM Output (Hardware Dead Zone)

- Step1.** Set the general period reference value (PERAR);
- Step2.** Set the general compare reference value A (GCMAR), and the dead time reference value (DTUAR, DTDAR);
- Step3.** Set the required interrupt enable bits, including the counter overflow interrupt (ICONR.INTENOVF), counter underflow interrupt (ICONR.INTENUF), count match interrupts (ICONR.INTENA-B), and the dead-time error interrupt (ICONR.INTENDTE), etc.;
- Step4.** Set the port output states for different counter conditions (refer to Bit17-Bit0 of PCNAR and PCNBR for related control settings. In conjunction with the configured values of GCMAR, DTUAR, and DTDAR, ensure that a complementary dead time is formed between the two PWM outputs);
- Step5.** Set the internal count clock division (GCONR.CKDIV);
- Step6.** Set the waveform mode to triangle wave mode (GCONR.MODE=1);
- Step7.** Set the comparison output mode of each channel (PCNAR.CAPMDA=0, PCNBR.CAPMDB=0);
- Step8.** Set each channel output enable (PCNAR.OUTENA=1, PCNBR.OUTENB=1);
- Step9.** Set the hardware dead zone function to be enabled (DCONR.DTCEN=1);
- Step10.** Start the counter (GCONR.START=1).

### 21.4.15 EMB Monitoring and Interrupt Action

- Step1.** Refer to steps 1-8 in the "[Complementary PWM Output \(Software Dead Zone\)](#)" section or steps 1-9 in the "[Complementary PWM Output \(Hardware Dead Zone\)](#)" section to set the complementary PWM output action;
- Step2.** Set the PWM port state when an EMB event occurs (PCNAR.EMBCA, PCNBR.EMBCB) (select the appropriate protection state according to the system application);
- Step3.** Set the time point at which the PWM port resumes normal output after the EMB event becomes inactive (PCNAR.EMBRA, PCNBR.EMBRB);



- Step4.** Set the EMB event source input from the EMB module (PCNAR.EMBSA, PCNBR.EMBSB);
- Step5.** Set the relevant EMB module registers (including the EMB interrupt permission registers (EMB\_INTEN0-3), EMB control registers 1/2 (EMB\_CTL1/2\_0-3), and the EMB control PWM output release mode selection registers (EMB\_RLSSEL0-3), etc.);
- Step6.** Start the counter (GCONR.START=1), and the EMB module monitors the system status in real time.

## 21.5 Register Description

### 21.5.1 Register List

- Timer6\_1 base address: 0x40018000
- Timer6\_2 base address: 0x40018400
- Timer6\_3 base Address: 0x40018800
- Timer6\_4 base address: 0x40018C00
- Timer6\_5 base address: 0x40019000
- Timer6\_6 base address: 0x40019400
- Timer6\_7 base address: 0x40019800
- Timer6\_8 base address: 0x40019C00



#### Note

Timer6 is the name of the timer, and Timer6\_<t> (where <t>: 1-8) refers to the timer unit.

Table 21-4 Advanced Control Timer (Timer6) Register Offset

| Offset | Register   | Description                                | Width |
|--------|------------|--------------------------------------------|-------|
| 0x0000 | TMR6_CNTER | General Count Value Register               | 32    |
| 0x0004 | TMR6_UPDAR | General Refresh Value Register             | 32    |
| 0x0040 | TMR6_PERAR | General Period Reference Value Register A  | 32    |
| 0x0044 | TMR6_PERBR | General Period Reference Value Register B  | 32    |
| 0x0048 | TMR6_PERCR | General Period Reference Value Register C  | 32    |
| 0x0080 | TMR6_GCMAR | General Compare Reference Value Register A | 32    |
| 0x0084 | TMR6_GCMBR | General Compare Reference Value Register B | 32    |
| 0x0088 | TMR6_GCMCR | General Compare Reference Value Register C | 32    |
| 0x008C | TMR6_GCMDR | General Compare Reference Value Register D | 32    |
| 0x0090 | TMR6_GCMER | General Compare Reference Value Register E | 32    |
| 0x0094 | TMR6_GCMFR | General Compare Reference Value Register F | 32    |
| 0x00C0 | TMR6_SCMAR | Special Compare Reference Value Register A | 32    |
| 0x00C4 | TMR6_SCMBR | Special Compare Reference Value Register B | 32    |
| 0x00C8 | TMR6_SCMCR | Special Compare Reference Value Register C | 32    |

| Offset | Register                    | Description                                  | Width |
|--------|-----------------------------|----------------------------------------------|-------|
| 0x00CC | <a href="#">TMR6_SCMDR</a>  | Special Compare Reference Value Register D   | 32    |
| 0x00D0 | <a href="#">TMR6_SCMER</a>  | Special Compare Reference Value Register E   | 32    |
| 0x00D4 | <a href="#">TMR6_SCMFR</a>  | Special Compare Reference Value Register F   | 32    |
| 0x0100 | <a href="#">TMR6_DTUAR</a>  | Dead Time Reference Value Register UA        | 32    |
| 0x0104 | <a href="#">TMR6_DTDAR</a>  | Dead Time Reference Value Register DA        | 32    |
| 0x0108 | <a href="#">TMR6_DTUBR</a>  | Dead Time Reference Value Register UB        | 32    |
| 0x010C | <a href="#">TMR6_DTDBR</a>  | Dead Time Reference Value Register DB        | 32    |
| 0x0140 | <a href="#">TMR6_GCONR</a>  | General Control Register                     | 32    |
| 0x0144 | <a href="#">TMR6_ICONR</a>  | Interrupt Control Register                   | 32    |
| 0x0148 | <a href="#">TMR6_BCONR</a>  | Buffer Control Register                      | 32    |
| 0x014C | <a href="#">TMR6_DCONR</a>  | Dead Zone Control Register                   | 32    |
| 0x0154 | <a href="#">TMR6_PCNAR</a>  | Port Control Register A                      | 32    |
| 0x0158 | <a href="#">TMR6_PCNBR</a>  | Port Control Register B                      | 32    |
| 0x015C | <a href="#">TMR6_FCNGR</a>  | Filter Control Register                      | 32    |
| 0x0160 | <a href="#">TMR6_VPERR</a>  | Valid Period Register                        | 32    |
| 0x0164 | <a href="#">TMR6_STFLR</a>  | Status Flag Register                         | 32    |
| 0x0180 | <a href="#">TMR6_HSTAR</a>  | Hardware Start Event Selection Register      | 32    |
| 0x0184 | <a href="#">TMR6_HSTPR</a>  | Hardware Stop Event Selection Register       | 32    |
| 0x0188 | <a href="#">TMR6_HCLRR</a>  | Hardware Clear Event Selection Register      | 32    |
| 0x018C | <a href="#">TMR6_HUPDR</a>  | Hardware Refresh Event Selection Register    | 32    |
| 0x0190 | <a href="#">TMR6_HCPAR</a>  | Hardware Capture Event Selection Register A  | 32    |
| 0x0194 | <a href="#">TMR6_HCPBR</a>  | Hardware Capture Event Selection Register B  | 32    |
| 0x0198 | <a href="#">TMR6_HCUPR</a>  | Hardware Count Up Event Selection Register   | 32    |
| 0x019C | <a href="#">TMR6_HCDOR</a>  | Hardware Count Down Event Selection Register | 32    |
| 0x0200 | <a href="#">TMR6_PTNUMR</a> | PTO Pulse Number Register                    | 32    |
| 0x0204 | <a href="#">TMR6_PTPERR</a> | PTO Pulse Period Register                    | 16    |

| Offset | Register                      | Description                                           | Width |
|--------|-------------------------------|-------------------------------------------------------|-------|
| 0x0208 | <a href="#">TMR6_PTRMDR</a>   | PTO Algorithm Period Remainder Register               | 32    |
| 0x020C | <a href="#">TMR6_PTCTLR</a>   | PTO Mode Control Register                             | 16    |
| 0x0220 | <a href="#">TMR6_PTAOPR</a>   | PTO A Phase Output Position Register                  | 16    |
| 0x0228 | <a href="#">TMR6_PTBOPR</a>   | PTO B Phase Output Position Register                  | 16    |
| 0x0240 | <a href="#">TMR6_PTZOSPR</a>  | PTO Z Phase Output Start Position Register            | 32    |
| 0x0248 | <a href="#">TMR6_PTZOEPR</a>  | PTO Z Phase Output End Position Register              | 32    |
| 0x0264 | <a href="#">TMR6_PTZCMPBR</a> | PTO Z Phase Output Pulse Width Compression B Register | 32    |
| 0x0268 | <a href="#">TMR6_PTZPERR</a>  | PTO Full Circle Pulse Count Register                  | 32    |
| 0x0280 | <a href="#">TMR6_PTPSPR</a>   | PTO Pulse Initial Position Register                   | 32    |
| 0x0284 | <a href="#">TMR6_PTPSTR</a>   | PTO Pulse Initial Port Register                       | 32    |
| 0x0290 | <a href="#">TMR6_PTPCPR</a>   | PTO Pulse Current Position Register                   | 32    |
| 0x0294 | <a href="#">TMR6_PTPCTR</a>   | PTO Pulse Current Port Register                       | 32    |

Timer6 common register base address: 0x40018300

Table 21-5 Timer6 Common Register Offset

| Offset | Register                   | Description                                          | Width |
|--------|----------------------------|------------------------------------------------------|-------|
| 0x00EC | <a href="#">TMR6_FCNT</a>  | Common Filter Control Register                       | 32    |
| 0x00F0 | <a href="#">TMR6_SSTAR</a> | Common Software Synchronous Start Control Register   | 32    |
| 0x00F4 | <a href="#">TMR6_SSTPR</a> | Common Software Synchronous Stop Control Register    | 32    |
| 0x00F8 | <a href="#">TMR6_SCLRR</a> | Common Software Synchronous Clear Control Register   | 32    |
| 0x00FC | <a href="#">TMR6_SUPDR</a> | Common Software Synchronous Refresh Control Register | 32    |

**Note**

- The bit widths of the general count value register, general refresh value register and reference registers of unit 1 to unit 4 are all 32 bits, and the reset values are different from those of unit 5 to unit 8. Among them, the reset value of the general count value register and the general refresh value register is 0x00000000; the reset value of each reference value register is 0xFFFFFFFF.
- Trigger Selection Registers (TMR6\_TRGSEL0-3) : These are 4 independent registers that are shared by all 8 Timer6 units..
- Software Synchronous Registers (TMR6\_SSTAR, TMR6\_SSTPR, TMR6\_SCLLR, TMR6\_SUPDR) : These are 4 independent registers that are shared by all 8 Timer6 units.
- Filter Control Register (TMR6\_FCNTNTR) : This is a single independent register shared by all 8 Timer6 units.

### 21.5.2 General Count Value Register (TMR6\_CNTER)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |     |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0x0000 | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15  | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |     |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | CNT |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |     |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|        |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RW  |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit   | Name     | Function    | Description                                                                                                                                                                                                                                                                                                                                                                                            | Access |
|-------|----------|-------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:16 | Reserved | Reserved    | -                                                                                                                                                                                                                                                                                                                                                                                                      | -      |
| 15:0  | CNT      | Count value | Current timer count value<br><br><div>  <b>Note</b><br/>           In the sawtooth up-counting mode, the initial value of CNT is set to 0. In the sawtooth down-counting mode, the initial value of CNT is set to the period value. In the triangular wave mode, the initial value of CNT is set to 1.         </div> | RW     |

### 21.5.3 General Refresh Value Register (TMR6\_UPDAR)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |      |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0x0004 | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15   | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |      |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | UPDA |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |      |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|        |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RW   |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit   | Name     | Function      | Description                                        | Access |
|-------|----------|---------------|----------------------------------------------------|--------|
| 31:16 | Reserved | Reserved      | -                                                  | -      |
| 15:0  | UPDA     | Refresh value | Set the refresh value to be updated into the timer | RW     |

### 21.5.4 General Period Reference Value Register (TMR6\_PERmR) (m=A-C)

Offset: 0x0040 (TMR6\_PERAR), 0x0044 (TMR6\_PERBR), 0x0048 (TMR6\_PERCR)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |     |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| -      | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15  | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x0000FFFF   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |     |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | PER |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RW  |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit   | Name     | Function              | Description                                                                      | Access |
|-------|----------|-----------------------|----------------------------------------------------------------------------------|--------|
| 31:16 | Reserved | Reserved              | -                                                                                | -      |
| 15:0  | PER      | Counting Period Value | Set the counting period value for each period and the corresponding buffer value | RW     |

### 21.5.5 General Compare Reference Value Register (TMR6\_GCMmR) (m=A-F)

Offset: 0x0080(TMR6\_GCMAR), 0x0084 (TMR6\_GCMBR), 0x0088 (TMR6\_GCMCR),  
0x008C(TMR6\_GCMDR), 0x0090(TMR6\_GCMER), 0x0094(TMR6\_GCMFR)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |     |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| -      | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15  | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x0000FFFF   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |     |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | GCM |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RW  |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit   | Name     | Function                      | Description                                                                                          | Access |
|-------|----------|-------------------------------|------------------------------------------------------------------------------------------------------|--------|
| 31:16 | Reserved | Reserved                      | -                                                                                                    | -      |
| 15:0  | GCM      | Count compare reference value | General compare reference setting, the matching signal is enable when it is equal to the count value | RW     |

### 21.5.6 Special Compare Reference Value Register (TMR6\_SCMmR) (m=A-F)

Offset : 0x00C0(TMR6\_SCMA), 0x00C4 (TMR6\_SCMB), 0x00C8 (TMR6\_SCMC),  
0x00CC(TMR6\_SCMD), 0x00D0(TMR6\_SCME), 0x00D4(TMR6\_SCMF)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |     |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| -      | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15  | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x0000FFFF   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |     |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | SCM |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RW  |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit   | Name     | Function                | Description                                       | Access |
|-------|----------|-------------------------|---------------------------------------------------|--------|
| 31:16 | Reserved | Reserved                | -                                                 | -      |
| 15:0  | SCM      | Special Reference value | Setting compare reference value and buffer values | RW     |

### 21.5.7 Dead Time Reference Value Register (TMR6\_DTmnR) (m=U/D, n=A/B)

Offset: 0x0100 (TMR6\_DTUAR), 0x0104(TMR6\_DTDAR), 0x0108(TMR6\_DTUBR), 0x010C(TMR6\_DTDDBR)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| -      | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x0000FFFF   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | DT |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit   | Name     | Function        | Description                              | Access |
|-------|----------|-----------------|------------------------------------------|--------|
| 31:16 | Reserved | Reserved        | -                                        | -      |
| 15:0  | DT       | Dead time value | Dead time setting value and buffer value | RW     |



## 21.5.8 General Control Register (TMR6\_GCONR)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |         |         |         |          |    |    |    |    |    |    |       |       |       |   |   |          |      |     |       |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|---------|---------|---------|----------|----|----|----|----|----|----|-------|-------|-------|---|---|----------|------|-----|-------|---|
| 0x0140 | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19      | 18      | 17      | 16       | 15 | 14 | 13 | 12 | 11 | 10 | 9     | 8     | 7     | 6 | 5 | 4        | 3    | 2   | 1     | 0 |
| Reset  | 0x00000002   |    |    |    |    |    |    |    |    |    |    |    |         |         |         |          |    |    |    |    |    |    |       |       |       |   |   |          |      |     |       |   |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    | ZMSKVAL | ZMSKPOS | ZMSKREV | Reserved |    |    |    |    |    |    | FODIR | OVSTP | CKDIV |   |   | Reserved | MODE | DIR | START |   |
| Access |              |    |    |    |    |    |    |    |    |    |    |    | RW      | RW      | RW      |          |    |    |    |    |    |    | RW    | RW    | RW    |   |   |          | RW   | RW  | RW    |   |

| Bit   | Name     | Function                                | Description                                                                                                                                                                                                                                                                                                                                                                                                                                          | Access |
|-------|----------|-----------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:20 | Reserved | Reserved                                | -                                                                                                                                                                                                                                                                                                                                                                                                                                                    | -      |
| 19:18 | ZMSKVAL  | Z Phase Input Mask Cycle Count          | <ul style="list-style-type: none"> <li>0b00: Z phase input mask function is disabled.</li> <li>0b01: Z phase input is masked during the 4 counting periods after position count overflow or underflow.</li> <li>0b10: Z phase input is masked during the 8 counting periods after position count overflow or underflow.</li> <li>0b11: Z phase input is masked during the 16 counting periods after position count overflow or underflow.</li> </ul> | RW     |
| 17    | ZMSKPOS  | Z Phase Input Position Timer Selection  | <ul style="list-style-type: none"> <li>0b0: When Z phase input is active, this unit functions as the position timer, and the position timer clear function operates normally during the mask period.</li> <li>0b1: When Z phase input is active, this unit functions as the position timer, and the position timer clear function is masked during the mask period.</li> </ul>                                                                       | RW     |
| 16    | ZMSKREV  | Z Phase Input Revolving Timer Selection | <ul style="list-style-type: none"> <li>0b0: When Z phase input is active, this unit functions as the revolving timer, and the revolving timer count function operates normally during the mask period.</li> <li>0b1: When Z phase input is active, this unit functions as the revolving timer, and the revolving timer count function is masked during the mask period.</li> </ul>                                                                   | RW     |
| 15:10 | Reserved | Reserved                                | -                                                                                                                                                                                                                                                                                                                                                                                                                                                    | -      |

| Bit | Name            | Function                      | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | Access |
|-----|-----------------|-------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 9   | FODIR           | Forced Count Direction Change | <ul style="list-style-type: none"> <li>0b0: In sawtooth mode, at the overflow or underflow point, the count direction is updated according to DIR; in triangle wave mode, the direction automatically switches at the overflow or underflow point, making the DIR setting inactive.</li> <li>0b1: Immediately updates the count direction to DIR.</li> </ul> <div>  <b>Note</b><br/>This function is enable when the counter is stopped.         </div> | RW     |
| 8   | OVSTP           | Count Overflow Stop Control   | <ul style="list-style-type: none"> <li>0b0: The counter continues counting after an overflow or underflow.</li> <li>0b1: The counter stops counting after an overflow or underflow.</li> </ul>                                                                                                                                                                                                                                                                                                                                           | RW     |
| 7:4 | CKDIV           | Count clock selection         | <ul style="list-style-type: none"> <li>0b0000: PCLK0</li> <li>0b0001: PCLK0/2</li> <li>0b0010: PCLK0/4</li> <li>0b0011: PCLK0/8</li> <li>0b0100: PCLK0/16</li> <li>0b0101: PCLK0/32</li> <li>0b0110: PCLK0/64</li> <li>0b0111: PCLK0/128</li> <li>0b1000: PCLK0/256</li> <li>0b1001: PCLK0/512</li> <li>0b1010: PCLK0/1024</li> <li>Others: Setting prohibited</li> </ul>                                                                                                                                                                | RW     |
| 3   | <i>Reserved</i> | <i>Reserved</i>               | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | -      |
| 2   | MODE            | Counting mode                 | <ul style="list-style-type: none"> <li>0b0: Sawtooth wave mode.</li> <li>0b1: Triangle wave mode.</li> </ul> <div>  <b>Note</b><br/>In PTO mode, set to sawtooth wave mode.         </div>                                                                                                                                                                                                                                                            | RW     |
| 1   | DIR             | Count direction               | <ul style="list-style-type: none"> <li>0b0: Decremental counting.</li> <li>0b1: Incremental counting.</li> </ul> <div>  <b>Note</b><br/>In PTO mode, set to incremental counting.         </div>                                                                                                                                                                                                                                                      | RW     |
| 0   | START           | Timer Start                   | <ul style="list-style-type: none"> <li>0b0: Timer OFF.</li> <li>0b1: Timer ON.</li> </ul> <div>  <b>Note</b><br/>This bit automatically becomes 0 when a software stop condition or hardware stop condition is triggered.         </div>                                                                                                                                                                                                              | RW     |

## 21.5.9 Interrupt Control Register (TMR6\_ICONR)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |          |          |          |          |          |    |    |    |    |    |   |   |          |          |          |        |        |        |        |        |        |    |    |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----------|----------|----------|----------|----------|----|----|----|----|----|---|---|----------|----------|----------|--------|--------|--------|--------|--------|--------|----|----|
| 0x0144 | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19       | 18       | 17       | 16       | 15       | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7        | 6        | 5        | 4      | 3      | 2      | 1      | 0      |        |    |    |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |          |          |          |          |          |    |    |    |    |    |   |   |          |          |          |        |        |        |        |        |        |    |    |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    | INTENSBD | INTENSBU | INTENSAD | INTENSAU | Reserved |    |    |    |    |    |   |   | INTENDTE | INTENUDF | INTENOVF | INTENF | INTENE | INTEND | INTENC | INTENB | INTENA |    |    |
| Access |              |    |    |    |    |    |    |    |    |    |    |    | RW       | RW       | RW       | RW       |          |    |    |    |    |    |   |   | RW       | RW       | RW       | RW     | RW     | RW     | RW     | RW     | RW     | RW | RW |

| Bit   | Name     | Function                                | Description                                                                                                                                                                                    | Access |
|-------|----------|-----------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:20 | Reserved | Reserved                                | -                                                                                                                                                                                              | -      |
| 19    | INTENSBD | Special Down-Counter Interrupt Enable B | Interrupt condition: During down-counting, the SCMBR register and the counter value (CNTERR) are equal.<br><ul style="list-style-type: none"> <li>0b0: Disable</li> <li>0b1: Enable</li> </ul> | RW     |
| 18    | INTENSBU | Special Up-Counter Interrupt Enable B   | Interrupt condition: During up-counting, the SCMBR register and the counter value (CNTERR) are equal.<br><ul style="list-style-type: none"> <li>0b0: Disable</li> <li>0b1: Enable</li> </ul>   | RW     |
| 17    | INTENSAD | Special Down-Counter Interrupt Enable A | Interrupt condition: During down-counting, the SCMAR register and the counter value (CNTERR) are equal.<br><ul style="list-style-type: none"> <li>0b0: Disable</li> <li>0b1: Enable</li> </ul> | RW     |
| 16    | INTENSAU | Special Up-Counter Interrupt Enable A   | Interrupt condition: During up-counting, the SCMAR register and the counter value (CNTERR) are equal.<br><ul style="list-style-type: none"> <li>0b0: Disable</li> <li>0b1: Enable</li> </ul>   | RW     |
| 15:9  | Reserved | Reserved                                | -                                                                                                                                                                                              | -      |
| 8     | INTENDTE | Dead Time Error Interrupt Enable        | Interrupt condition: Dead time error.<br><ul style="list-style-type: none"> <li>0b0: Disable</li> <li>0b1: Enable</li> </ul>                                                                   | RW     |
| 7     | INTENUDF | Underflow Interrupt Enable              | Interrupt condition: Counter underflow.<br><ul style="list-style-type: none"> <li>0b0: Disable</li> <li>0b1: Enable</li> </ul>                                                                 | RW     |
| 6     | INTENOVF | Overflow Interrupt Enable               | Interrupt condition: Counter overflow.<br><ul style="list-style-type: none"> <li>0b0: Disable</li> <li>0b1: Enable</li> </ul>                                                                  | RW     |

| Bit | Name   | Function                       | Description                                                                                                                                                                                            | Access |
|-----|--------|--------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 5   | INTENF | Count Match Interrupt Enable F | Interrupt condition: the GCMFR register is equal to the counter value (CNTERR).<br><ul style="list-style-type: none"> <li>● 0b0: Disable</li> <li>● 0b1: Enable</li> </ul>                             | RW     |
| 4   | INTENE | Count Match Interrupt Enable E | Interrupt condition: the GCMEER register is equal to the counter value (CNTERR).<br><ul style="list-style-type: none"> <li>● 0b0: Disable</li> <li>● 0b1: Enable</li> </ul>                            | RW     |
| 3   | INTEND | Count Match Interrupt Enable D | Interrupt condition: the GCMDR register is equal to the counter value (CNTERR).<br><ul style="list-style-type: none"> <li>● 0b0: Disable</li> <li>● 0b1: Enable</li> </ul>                             | RW     |
| 2   | INTENC | Count Match Interrupt Enable C | Interrupt condition: the GCMCR register is equal to the counter value (CNTERR).<br><ul style="list-style-type: none"> <li>● 0b0: Disable</li> <li>● 0b1: Enable</li> </ul>                             | RW     |
| 1   | INTENB | Count Match Interrupt Enable B | Interrupt condition: the GCMBR register is equal to the counter value (CNTERR), or an input capture occurs.<br><ul style="list-style-type: none"> <li>● 0b0: Disable</li> <li>● 0b1: Enable</li> </ul> | RW     |
| 0   | INTENA | Count Match Interrupt Enable A | Interrupt condition: the GCMAR register is equal to the counter value (CNTERR), or an input capture occurs.<br><ul style="list-style-type: none"> <li>● 0b0: Disable</li> <li>● 0b1: Enable</li> </ul> | RW     |

## 21.5.10 Buffer Control Register (TMR6\_BCONR)

| Offset | Bit Position |    |    |    |    |    |    |    |         |         |        |        |         |         |        |        |          |    |    |    |    |       |       |      |      |       |       |      |      |       |       |      |      |    |
|--------|--------------|----|----|----|----|----|----|----|---------|---------|--------|--------|---------|---------|--------|--------|----------|----|----|----|----|-------|-------|------|------|-------|-------|------|------|-------|-------|------|------|----|
| 0x0148 | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23      | 22      | 21     | 20     | 19      | 18      | 17     | 16     | 15       | 14 | 13 | 12 | 11 | 10    | 9     | 8    | 7    | 6     | 5     | 4    | 3    | 2     | 1     | 0    |      |    |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |         |         |        |        |         |         |        |        |          |    |    |    |    |       |       |      |      |       |       |      |      |       |       |      |      |    |
| Name   | Reserved     |    |    |    |    |    |    |    | BTRDSPB | BTRUSPB | BSESPB | BENSPB | BTRDSPA | BTRUSPA | BSESPA | BENSPA | Reserved |    |    |    |    | BTRDP | BTRUP | BSEP | BENP | BTRDB | BTRUB | BSEB | BENB | BTRDA | BTRUA | BSEA | BENA |    |
| Access |              |    |    |    |    |    |    |    | RW      | RW      | RW     | RW     | RW      | RW      | RW     | RW     |          |    |    |    |    | RW    | RW    | RW   | RW   | RW    | RW    | RW   | RW   | RW    | RW    | RW   | RW   | RW |

| Bit   | Name     | Function                                                    | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | Access |
|-------|----------|-------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:24 | Reserved | Reserved                                                    | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | -      |
| 23    | BTRDSPB  | Special Compare Reference Buffer Transfer Timing Setting DB | <ul style="list-style-type: none"> <li>0b0: In triangle wave counting mode, the buffer value is not transferred when the counter reaches the valley point.</li> <li>0b1: In triangle wave counting mode, the buffer value is transferred when the counter reaches the valley point.</li> </ul> <div>  <b>Note</b><br/>           When in sawtooth wave mode, this bit control is not required.         </div> | RW     |
| 22    | BTRUSPB  | Special Compare Reference Buffer Transfer Timing Setting UB | <ul style="list-style-type: none"> <li>0b0: In triangle wave counting mode, the buffer value is not transferred when the counter reaches the peak point.</li> <li>0b1: In triangle wave counting mode, the buffer value is transferred when the counter reaches the peak point.</li> </ul> <div>  <b>Note</b><br/>           When in sawtooth wave mode, this bit control is not required.         </div>    | RW     |
| 21    | BSESPB   | Special Compare Reference Buffer Transfer Selection B       | <ul style="list-style-type: none"> <li>0b0: Single buffer transfer (SCMDR -&gt; SCMBR).</li> <li>0b1: Double buffer transfer (SCMFR -&gt; SCMDR -&gt; SCMBR).</li> </ul>                                                                                                                                                                                                                                                                                                                        | RW     |
| 20    | BENSPB   | Special Compare Reference Buffer Transfer B                 | <ul style="list-style-type: none"> <li>0b0: Buffer transfer disabled.</li> <li>0b1: Buffer transfer enabled.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                         | RW     |

| Bit   | Name     | Function                                                    | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | Access |
|-------|----------|-------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 19    | BTRDSPA  | Special Compare Reference Buffer Transfer Timing Setting DA | <ul style="list-style-type: none"> <li>0b0: In triangle wave counting mode, the buffer value is not transferred when the counter reaches the valley point.</li> <li>0b1: In triangle wave counting mode, the buffer value is transferred when the counter reaches the valley point.</li> </ul> <div>  <b>Note</b><br/>           When in sawtooth wave mode, this bit control is not required.         </div>   | RW     |
| 18    | BTRUSPA  | Special Compare Reference Buffer Transfer Timing Setting UA | <ul style="list-style-type: none"> <li>0b0: In triangle wave counting mode, the buffer value is not transferred when the counter reaches the peak point.</li> <li>0b1: In triangle wave counting mode, the buffer value is transferred when the counter reaches the peak point.</li> </ul> <div>  <b>Note</b><br/>           When in sawtooth wave mode, this bit control is not required.         </div>       | RW     |
| 17    | BSESPA   | Special Compare Reference Buffer Transfer Selection A       | <ul style="list-style-type: none"> <li>0b0: Single buffer transfer (SCMCR -&gt; SCMAR).</li> <li>0b1: Double buffer transfer (SCMER -&gt; SCMCR -&gt; SCMAR).</li> </ul>                                                                                                                                                                                                                                                                                                                         | RW     |
| 16    | BENSPA   | Special Compare Reference Buffer Transfer A                 | <ul style="list-style-type: none"> <li>0b0: Buffer transfer disabled.</li> <li>0b1: Buffer transfer enabled.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                          | RW     |
| 15:12 | Reserved | Reserved                                                    | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | -      |
| 11    | BTRDP    | Period Value Buffer Transfer Timing Setting D               | <ul style="list-style-type: none"> <li>0b0: In triangle wave counting mode, the buffer value is not transferred when the counter reaches the valley point.</li> <li>0b1: In triangle wave counting mode, the buffer value is transferred when the counter reaches the valley point.</li> </ul> <div>  <b>Note</b><br/>           When in sawtooth wave mode, this bit control is not required.         </div> | RW     |

| Bit | Name  | Function                                                | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | Access |
|-----|-------|---------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 10  | BTRUP | Period Value Buffer Transfer Timing Setting U           | <ul style="list-style-type: none"> <li>0b0: In triangle wave counting mode, the buffer value is not transferred when the counter reaches the peak point.</li> <li>0b1: In triangle wave counting mode, the buffer value is transferred when the counter reaches the peak point.</li> </ul> <div>  <b>Note</b><br/> When in sawtooth wave mode, this bit control is not required. </div>         | RW     |
| 9   | BSEP  | Period Value Buffer Transfer Selection                  | <ul style="list-style-type: none"> <li>0b0: Single buffer transfer (PERBR -&gt; PERAR).</li> <li>0b1: Double buffer transfer (PERCR -&gt; PERBR -&gt; PERAR)</li> </ul> <div>  <b>Note</b><br/> The transfer timing is independent of the counting mode and only occurs at the upward overflow, downward overflow points in sawtooth mode, or at the valley point in triangle wave mode. </div> | RW     |
| 8   | BENP  | Period Value Buffer Transfer                            | <ul style="list-style-type: none"> <li>0b0: Buffer transfer disabled.</li> <li>0b1: Buffer transfer enabled.</li> </ul>                                                                                                                                                                                                                                                                                                                                                          | RW     |
| 7   | BTRDB | General Compare Value Buffer Transfer Timing Setting DB | <ul style="list-style-type: none"> <li>0b0: In triangle wave counting mode, the buffer value is not transferred when the counter reaches the valley point.</li> <li>0b1: In triangle wave counting mode, the buffer value is transferred when the counter reaches the valley point.</li> </ul> <div>  <b>Note</b><br/> When in sawtooth wave mode, this bit control is not required. </div>   | RW     |
| 6   | BTRUB | General Compare Value Buffer Transfer Timing Setting UB | <ul style="list-style-type: none"> <li>0b0: In triangle wave counting mode, the buffer value is not transferred when the counter reaches the peak point.</li> <li>0b1: In triangle wave counting mode, the buffer value is transferred when the counter reaches the peak point.</li> </ul> <div>  <b>Note</b><br/> When in sawtooth wave mode, this bit control is not required. </div>       | RW     |

| Bit | Name  | Function                                                | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | Access |
|-----|-------|---------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 5   | BSEB  | General Compare Value Buffer Transfer Selection B       | <p>For comparison output function:</p> <ul style="list-style-type: none"> <li>● 0b0: Single buffer transfer (GCMDBR -&gt; GCMBR).</li> <li>● 0b1: Dual buffer transfer (GCMFBR -&gt; GCMDBR -&gt; GCMBR).</li> </ul> <p>For capture input function:</p> <ul style="list-style-type: none"> <li>● 0b0: Single buffer transfer (GCMBR -&gt; GCMDBR).</li> <li>● 0b1: Dual buffer transfer (GCMBR -&gt; GCMDBR -&gt; GCMFBR).</li> </ul>                                                              | RW     |
| 4   | BENB  | General Compare Value Buffer Transfer B                 | <ul style="list-style-type: none"> <li>● 0b0: Buffer transfer disabled.</li> <li>● 0b1: Buffer transfer enabled.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                        | RW     |
| 3   | BTRDA | General Compare Value Buffer Transfer Timing Setting DA | <ul style="list-style-type: none"> <li>● 0b0: In triangle wave counting mode, the buffer value is not transferred when the counter reaches the valley point.</li> <li>● 0b1: In triangle wave counting mode, the buffer value is transferred when the counter reaches the valley point.</li> </ul> <div>  <b>Note</b><br/>           When in sawtooth wave mode, this bit control is not required.         </div> | RW     |
| 2   | BTRUA | General Compare Value Buffer Transfer Timing Setting UA | <ul style="list-style-type: none"> <li>● 0b0: In triangle wave counting mode, the buffer value is not transferred when the counter reaches the peak point.</li> <li>● 0b1: In triangle wave counting mode, the buffer value is transferred when the counter reaches the peak point.</li> </ul> <div>  <b>Note</b><br/>           When in sawtooth wave mode, this bit control is not required.         </div>   | RW     |
| 1   | BSEA  | General Compare Value Buffer Transfer Selection A       | <p>For comparison output function:</p> <ul style="list-style-type: none"> <li>● 0b0: Single buffer transfer (GCMCR -&gt; GCMA).</li> <li>● 0b1: Dual buffer transfer (GCMA -&gt; GCMCR -&gt; GCMA).</li> </ul> <p>For capture input function:</p> <ul style="list-style-type: none"> <li>● 0b0: Single buffer transfer (GCMA -&gt; GCMCR).</li> <li>● 0b1: Dual buffer transfer (GCMA -&gt; GCMCR -&gt; GCMA).</li> </ul>                                                                          | RW     |
| 0   | BENA  | General Compare Value Buffer Transfer A                 | <ul style="list-style-type: none"> <li>● 0b0: Buffer transfer disabled.</li> <li>● 0b1: Buffer transfer enabled.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                        | RW     |



## 21.5.11 Dead Zone Control Register (TMR6\_DCONR)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |        |   |        |   |        |   |        |  |          |  |      |  |       |  |       |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|--------|---|--------|---|--------|---|--------|--|----------|--|------|--|-------|--|-------|
| 0x014C | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6      | 5 | 4      | 3 | 2      | 1 | 0      |  |          |  |      |  |       |  |       |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |        |   |        |   |        |   |        |  |          |  |      |  |       |  |       |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   | DTBTRD |   | DTBTRU |   | DTBEND |   | DTBENU |  | Reserved |  | SEPA |  | DTCEN |  |       |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   | RW     |   | RW     |   | RW     |   | RW     |  |          |  | RW   |  | RW    |  | DTCEN |

| Bit  | Name     | Function                                            | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | Access |
|------|----------|-----------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:8 | Reserved | Reserved                                            | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | -      |
| 7    | DTBTRD   | Dead Zone Time Value Buffer Transfer Time Setting D | <ul style="list-style-type: none"> <li>0b0: In triangular wave counting mode, the dead zone buffer value is not transferred when counting reaches the valley.</li> <li>0b1: In triangular wave counting mode, the dead zone buffer value is transferred when counting reaches the valley.</li> </ul> <div>  <b>Note</b><br/>           In sawtooth wave mode, this bit does not control..         </div> | RW     |
| 6    | DTBTRU   | Dead Zone Time Value Buffer Transfer Time Setting U | <ul style="list-style-type: none"> <li>0b0: In triangular wave counting mode, the dead zone buffer value is not transferred when counting reaches the peak.</li> <li>0b1: In triangular wave counting mode, the dead zone buffer value is transferred when counting reaches the peak.</li> </ul> <div>  <b>Note</b><br/>           In sawtooth wave mode, this bit does not control..         </div>    | RW     |
| 5    | DTBEND   | Dead Zone Time Value Buffer Transfer D              | <ul style="list-style-type: none"> <li>0b0: Buffer transfer disabled.</li> <li>0b1: Buffer transfer enabled (DTDBR -&gt; DTDAR).</li> </ul>                                                                                                                                                                                                                                                                                                                                                | RW     |
| 4    | DTBENU   | Dead Zone Time Value Buffer Transfer U              | <ul style="list-style-type: none"> <li>0b0: Buffer transfer disabled.</li> <li>0b1: Buffer transfer enabled (DTUBR -&gt; DTUAR).</li> </ul>                                                                                                                                                                                                                                                                                                                                                | RW     |
| 3:2  | Reserved | Reserved                                            | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | -      |
| 1    | SEPA     | Separation Setting                                  | <ul style="list-style-type: none"> <li>0b0: DTUAR and DTDAR are set separately.</li> <li>0b1: DTUAR and DTDAR values are automatically equal.</li> </ul>                                                                                                                                                                                                                                                                                                                                   | RW     |
| 0    | DTCEN    | Dead Zone Function                                  | <ul style="list-style-type: none"> <li>0b0: Dead zone function disabled.</li> <li>0b1: Dead zone function enabled.</li> </ul>                                                                                                                                                                                                                                                                                                                                                              | RW     |

## 21.5.12 Port Control Register A (TMR6\_PCNAR)

| Offset | Bit Position |          |        |          |       |       |       |          |       |        |        |        |        |       |       |       |
|--------|--------------|----------|--------|----------|-------|-------|-------|----------|-------|--------|--------|--------|--------|-------|-------|-------|
| 0x0154 | 31           | 30       | 29     | 28       | 27    | 26    | 25    | 24       | 23    | 22     | 21     | 20     | 19     | 18    | 17    | 16    |
| Reset  | 0x00000000   |          |        |          |       |       |       |          |       |        |        |        |        |       |       |       |
| Name   | CAPMDA       | Reserved | OUTENA | Reserved | EMBSA | EMBRA | EMBCA | Reserved | FORCA | CMBDCA | CMBUCA | CMADCA | CMAUCA | UDFCA | OVFCA | STPCA |
| Access | RW           |          | RW     |          | RW    | RW    | RW    |          | RW    | RW     | RW     | RW     | RW     | RW    | RW    | RW    |

| Bit   | Name     | Function                         | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | Access |
|-------|----------|----------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31    | CAPMDA   | Function Mode Selection A        | <ul style="list-style-type: none"> <li>0b0: Compare Output Function</li> <li>0b1: Capture Input Function</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | RW     |
| 30:29 | Reserved | Reserved                         | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | -      |
| 28    | OUTENA   | Output Enable A                  | <ul style="list-style-type: none"> <li>0b0: When Timer6 is enabled, the TMR6_&lt;t&gt;_PWMA port output is disabled.</li> <li>0b1: When Timer6 is enabled, the TMR6_&lt;t&gt;_PWMA port output is enabled.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | RW     |
| 27:26 | Reserved | Reserved                         | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | -      |
| 25:24 | EMBSA    | EMB Event Port State Selection A | <ul style="list-style-type: none"> <li>0b00: Selects that the EMB Group 0 control event is active.</li> <li>0b01: Selects that the EMB Group 1 control event is active.</li> <li>0b10: Selects that the EMB Group 2 control event is active.</li> <li>0b11: Selects that the EMB Group 3 control event is active.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                              | RW     |
| 23:22 | EMBRA    | EMB Release Mode Selection A     | <ul style="list-style-type: none"> <li>0b00: When the EMB control event selected by EMBSA is disabled, the TMR6_&lt;t&gt;_PWMA port is immediately released (One Shot).</li> <li>0b01: When the EMB control event selected by EMBSA is disabled, the TMR6_&lt;t&gt;_PWMA port is released when the counter reaches the overflow condition (Cycle By Cycle 1).</li> <li>0b10: When the EMB control event selected by EMBSA is disabled, the TMR6_&lt;t&gt;_PWMA port is released when the counter reaches the underflow condition (Cycle By Cycle 2).</li> <li>0b11: When the EMB control event selected by EMBSA is disabled, the TMR6_&lt;t&gt;_PWMA port is released when the counter reaches either the overflow or underflow condition (Cycle By Cycle 3).</li> </ul> | RW     |

| Bit   | Name     | Function                                                   | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | Access |
|-------|----------|------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 21:20 | EMBCA    | EMB Event Port State Setting A                             | <ul style="list-style-type: none"> <li>0b00: When the EMB control event selected by EMBSA occurs, the TMR6_&lt;t&gt;_PWMA port will output normally.</li> <li>0b01: When the EMB control event selected by EMBSA occurs, the TMR6_&lt;t&gt;_PWMA port will output in high impedance state.</li> <li>0b10: When the EMB control event selected by EMBSA occurs, the TMR6_&lt;t&gt;_PWMA port will output a low level.</li> <li>0b11: When the EMB control event selected by EMBSA occurs, the TMR6_&lt;t&gt;_PWMA port will output a high level.</li> </ul>                                                                                                                                                                   | RW     |
| 19:18 | Reserved | Reserved                                                   | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | -      |
| 17:16 | FORCA    | Forced Port State Setting A                                | <ul style="list-style-type: none"> <li>0b0x: Setting prohibited</li> <li>0b10: At the start of the next period, the TMR6_&lt;t&gt;_PWMA port output is set to a low level.</li> <li>0b11: At the start of the next period, the TMR6_&lt;t&gt;_PWMA port output is set to a high level.</li> </ul> <div>  <b>Note</b><br/>           The next period refers to the hardware counting mode or the sawtooth wave counting to the overflow point or underflow point, and the triangle wave counting to the valley point.<br/><br/>           This register bit can be used to control the PWM output duty period to 0% or 100%.         </div> | RW     |
| 15:14 | CMBDCA   | Down-counting & Compare Value B Match Port State Setting A | <ul style="list-style-type: none"> <li>0b00: During down-counting, when the timer count value equals GCMBR, the TMR6_&lt;t&gt;_PWMA port output is set to low level.</li> <li>0b01: During down-counting, when the timer count value equals GCMBR, the TMR6_&lt;t&gt;_PWMA port output is set to high level.</li> <li>0b10: During down-counting, when the timer count value equals GCMBR, the TMR6_&lt;t&gt;_PWMA port output holds the previous state.</li> <li>0b11: During down-counting, when the timer count value equals GCMBR, the TMR6_&lt;t&gt;_PWMA port output is set to inverted level.</li> </ul>                                                                                                              | RW     |

| Bit   | Name   | Function                                                   | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | Access |
|-------|--------|------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 13:12 | CMBUCA | Up-counting & Compare Value B Match Port State Setting A   | <ul style="list-style-type: none"> <li>● 0b00: During up-counting, when the timer count value equals GCMBR, the TMR6_&lt;t&gt;_PWMA port output is set to low level.</li> <li>● 0b01: During up-counting, when the timer count value equals GCMBR, the TMR6_&lt;t&gt;_PWMA port output is set to high level.</li> <li>● 0b10: During up-counting, when the timer count value equals GCMBR, the TMR6_&lt;t&gt;_PWMA port output holds the previous state.</li> <li>● 0b11: During up-counting, when the timer count value equals GCMBR, the TMR6_&lt;t&gt;_PWMA port output is set to inverted level.</li> </ul>         | RW     |
| 11:10 | CMADCA | Down-counting & Compare Value A Match Port State Setting A | <ul style="list-style-type: none"> <li>● 0b00: During down-counting, when the timer count value equals GCMAR, the TMR6_&lt;t&gt;_PWMA port output is set to low level.</li> <li>● 0b01: During down-counting, when the timer count value equals GCMAR, the TMR6_&lt;t&gt;_PWMA port output is set to high level.</li> <li>● 0b10: During down-counting, when the timer count value equals GCMAR, the TMR6_&lt;t&gt;_PWMA port output holds the previous state.</li> <li>● 0b11: During down-counting, when the timer count value equals GCMAR, the TMR6_&lt;t&gt;_PWMA port output is set to inverted level.</li> </ul> | RW     |
| 9:8   | CMAUCA | Up-counting & Compare Value A Match Port State Setting A   | <ul style="list-style-type: none"> <li>● 0b00: During up-counting, when the timer count value equals GCMAR, the TMR6_&lt;t&gt;_PWMA port output is set to low level.</li> <li>● 0b01: During up-counting, when the timer count value equals GCMAR, the TMR6_&lt;t&gt;_PWMA port output is set to high level.</li> <li>● 0b10: During up-counting, when the timer count value equals GCMAR, the TMR6_&lt;t&gt;_PWMA port output holds the previous state.</li> <li>● 0b11: During up-counting, when the timer count value equals GCMAR, the TMR6_&lt;t&gt;_PWMA port output is set to inverted level.</li> </ul>         | RW     |

| Bit | Name  | Function                              | Description                                                                                                                                                                                                                                                                                                                                                                                                                                         | Access |
|-----|-------|---------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 7:6 | UDFCA | Count Under-flow Port State Setting A | <ul style="list-style-type: none"> <li>● 0b00: On count underflow, the TMR6_&lt;t&gt;_PWMA port output is set to low level.</li> <li>● 0b01: On count underflow, the TMR6_&lt;t&gt;_PWMA port output is set to high level.</li> <li>● 0b10: On count underflow, the TMR6_&lt;t&gt;_PWMA port output holds the previous state.</li> <li>● 0b11: On count underflow, the TMR6_&lt;t&gt;_PWMA port output is set to inverted level.</li> </ul>         | RW     |
| 5:4 | OVFCA | Count Overflow Port State Setting A   | <ul style="list-style-type: none"> <li>● 0b00: On count overflow, the TMR6_&lt;t&gt;_PWMA port output is set to low level.</li> <li>● 0b01: On count overflow, the TMR6_&lt;t&gt;_PWMA port output is set to high level.</li> <li>● 0b10: On count overflow, the TMR6_&lt;t&gt;_PWMA port output holds the previous state.</li> <li>● 0b11: On count overflow, the TMR6_&lt;t&gt;_PWMA port output is set to inverted level.</li> </ul>             | RW     |
| 3:2 | STPCA | Count Stop Port State Setting A       | <ul style="list-style-type: none"> <li>● 0b00: When counting stops, the TMR6_&lt;t&gt;_PWMA port output is set to low level.</li> <li>● 0b01: When counting stops, the TMR6_&lt;t&gt;_PWMA port output is set to high level.</li> <li>● 0b10: When counting stops, the TMR6_&lt;t&gt;_PWMA port output holds the previous state.</li> <li>● 0b11: When counting stops, the TMR6_&lt;t&gt;_PWMA port output holds the previous state.</li> </ul>     | RW     |
| 1:0 | STACA | Count Start Port State Setting A      | <ul style="list-style-type: none"> <li>● 0b00: When counting starts, the TMR6_&lt;t&gt;_PWMA port output is set to low level.</li> <li>● 0b01: When counting starts, the TMR6_&lt;t&gt;_PWMA port output is set to high level.</li> <li>● 0b10: When counting starts, the TMR6_&lt;t&gt;_PWMA port output holds the previous state.</li> <li>● 0b11: When counting starts, the TMR6_&lt;t&gt;_PWMA port output holds the previous state.</li> </ul> | RW     |

## 21.5.13 Port Control Register B (TMR6\_PCNBR)

| Offset | Bit Position |          |        |          |       |       |       |          |       |        |        |        |        |       |       |       |
|--------|--------------|----------|--------|----------|-------|-------|-------|----------|-------|--------|--------|--------|--------|-------|-------|-------|
| 0x0158 | 31           | 30       | 29     | 28       | 27    | 26    | 25    | 24       | 23    | 22     | 21     | 20     | 19     | 18    | 17    | 16    |
| Reset  | 0x00000000   |          |        |          |       |       |       |          |       |        |        |        |        |       |       |       |
| Name   | CAPMDB       | Reserved | OUTENB | Reserved | EMBSB | EMBRB | EMBCB | Reserved | FORCB | CMBDCB | CMBUCB | CMADCB | CMAUCB | UDFCB | OVFCB | STPCB |
| Access | RW           |          | RW     |          | RW    | RW    | RW    |          | RW    | RW     | RW     | RW     | RW     | RW    | RW    | RW    |

| Bit   | Name     | Function                          | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | Access |
|-------|----------|-----------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31    | CAPMDB   | Function Mode Selection B         | <ul style="list-style-type: none"> <li>0b0: Compare output function.</li> <li>0b1: Capture input function.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | RW     |
| 30:29 | Reserved | Reserved                          | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | -      |
| 28    | OUTENB   | Output Enable B                   | <ul style="list-style-type: none"> <li>0b0: TMR6_&lt;t&gt;_PWMB output is disabled during Timer6 operation.</li> <li>0b1: TMR6_&lt;t&gt;_PWMB output is enabled during Timer6 operation.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | RW     |
| 27:26 | Reserved | Reserved                          | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | -      |
| 25:24 | EMBSB    | EMB Event Port Status Selection B | <ul style="list-style-type: none"> <li>0b00: Selects EMB Group0 control event as active.</li> <li>0b01: Selects that the EMB Group 1 control event is active.</li> <li>0b10: Selects that the EMB Group 2 control event is active.</li> <li>0b11: Selects that the EMB Group 3 control event is active.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                   | RW     |
| 23:22 | EMBRB    | EMB Release Mode Selection B      | <ul style="list-style-type: none"> <li>0b00: When the EMB control event selected by EMBSB is disabled, the TMR6_&lt;t&gt;_PWMB port is immediately released (One Shot).</li> <li>0b01: When the EMB control event selected by EMBSB is disabled, the TMR6_&lt;t&gt;_PWMB port is released when the counter overflows (Cycle By Cycle 1).</li> <li>0b10: When the EMB control event selected by EMBSB is disabled, the TMR6_&lt;t&gt;_PWMB port is released when the counter underflows (Cycle By Cycle 2).</li> <li>0b11: When the EMB control event selected by EMBSB is disabled, the TMR6_&lt;t&gt;_PWMB port is released when the counter overflows or underflows (Cycle By Cycle 3).</li> </ul> | RW     |

| Bit   | Name     | Function                                                               | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | Access |
|-------|----------|------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 21:20 | EMBCB    | EMB Event Port Status Setting B                                        | <ul style="list-style-type: none"> <li>0b00: When the EMB control event selected by EMBSB occurs, the TMR6_&lt;t&gt;_PWMB port outputs normally.</li> <li>0b01: When the EMB control event selected by EMBSB occurs, the TMR6_&lt;t&gt;_PWMB port outputs high impedance.</li> <li>0b10: When the EMB control event selected by EMBSB occurs, the TMR6_&lt;t&gt;_PWMB port outputs a low level.</li> <li>0b11: When the EMB control event selected by EMBSB occurs, the TMR6_&lt;t&gt;_PWMB port outputs a high level.</li> </ul>                                                                                                                                                                                                    | RW     |
| 19:18 | Reserved | Reserved                                                               | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | -      |
| 17:16 | FORCB    | Forced Port State Setting B                                            | <ul style="list-style-type: none"> <li>0b0x: Setting prohibited</li> <li>0b10: At the beginning of the next period, the TMR6_&lt;t&gt;_PWMB port output is forced low level.</li> <li>0b11: At the beginning of the next period, the TMR6_&lt;t&gt;_PWMB port output is forced high level.</li> </ul> <div>  <b>Note</b><br/>           "Next period" refers to the moment when the hardware counter reaches an overflow or underflow point in sawtooth counting mode, or a valley point in triangle wave mode.<br/><br/>           This register bit can be used to implement PWM output control with a duty period of 0% or 100%.         </div> | RW     |
| 15:14 | CMBDCB   | Port Output State Setting B on Down-Count & Compare Match with Value B | <ul style="list-style-type: none"> <li>0b00: During down-counting, when the timer count equals GCMBR, the TMR6_&lt;t&gt;_PWMB port output is set low level.</li> <li>0b01: During down-counting, when the timer count equals GCMBR, the TMR6_&lt;t&gt;_PWMB port output is set high level.</li> <li>0b10: During down-counting, when the timer count equals GCMBR, the TMR6_&lt;t&gt;_PWMB port output maintains its previous state.</li> <li>0b11: During down-counting, when the timer count equals GCMBR, the TMR6_&lt;t&gt;_PWMB port output inverted level.</li> </ul>                                                                                                                                                          | RW     |

| Bit   | Name   | Function                                                               | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | Access |
|-------|--------|------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 13:12 | CMBUCB | Port Output State Setting B on Up-Count & Compare Match with Value B   | <ul style="list-style-type: none"> <li>0b00: During up-counting, when the timer count equals GCMBR, the TMR6_&lt;t&gt;_PWMB port output is set low level.</li> <li>0b01: During up-counting, when the timer count equals GCMBR, the TMR6_&lt;t&gt;_PWMB port output is set high level.</li> <li>0b10: During up-counting, when the timer count equals GCMBR, the TMR6_&lt;t&gt;_PWMB port output maintains its previous state.</li> <li>0b11: During up-counting, when the timer count equals GCMBR, the TMR6_&lt;t&gt;_PWMB port output inverted level.</li> </ul>         | RW     |
| 11:10 | CMADCB | Port Output State Setting B on Down-Count & Compare Match with Value A | <ul style="list-style-type: none"> <li>0b00: During down-counting, when the timer count equals GCMAR, the TMR6_&lt;t&gt;_PWMB port output is set low level.</li> <li>0b01: During down-counting, when the timer count equals GCMAR, the TMR6_&lt;t&gt;_PWMB port output is set high level.</li> <li>0b10: During down-counting, when the timer count equals GCMAR, the TMR6_&lt;t&gt;_PWMB port output maintains its previous state.</li> <li>0b11: During down-counting, when the timer count equals GCMAR, the TMR6_&lt;t&gt;_PWMB port output inverted level.</li> </ul> | RW     |
| 9:8   | CMAUCB | Port Output State Setting B on Up-Count & Compare Match with Value A   | <ul style="list-style-type: none"> <li>0b00: During up-counting, when the timer count equals GCMAR, the TMR6_&lt;t&gt;_PWMB port output is set low level.</li> <li>0b01: During up-counting, when the timer count equals GCMAR, the TMR6_&lt;t&gt;_PWMB port output is set high level.</li> <li>0b10: During up-counting, when the timer count equals GCMAR, the TMR6_&lt;t&gt;_PWMB port output maintains its previous state.</li> <li>0b11: During up-counting, when the timer count equals GCMAR, the TMR6_&lt;t&gt;_PWMB port output inverted level.</li> </ul>         | RW     |
| 7:6   | UDFCB  | Count Under-flow Port Output State Setting B                           | <ul style="list-style-type: none"> <li>0b00: When a count underflow occurs, the TMR6_&lt;t&gt;_PWMB port output is set low level.</li> <li>0b01: When a count underflow occurs, the TMR6_&lt;t&gt;_PWMB port output is set high level.</li> <li>0b10: When a count underflow occurs, the TMR6_&lt;t&gt;_PWMB port output retains its previous state.</li> <li>0b11: When a count underflow occurs, the TMR6_&lt;t&gt;_PWMB port output inverted level.</li> </ul>                                                                                                           | RW     |



| Bit | Name  | Function                                   | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                           | Access |
|-----|-------|--------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 5:4 | OVFCB | Count Overflow Port Output State Setting B | <ul style="list-style-type: none"> <li>● 0b00: When a count overflow occurs, the TMR6_&lt;t&gt;_PWMB port output is set low level.</li> <li>● 0b01: When a count overflow occurs, the TMR6_&lt;t&gt;_PWMB port output is set high level.</li> <li>● 0b10: When a count overflow occurs, the TMR6_&lt;t&gt;_PWMB port output retains its previous state.</li> <li>● 0b11: When a count overflow occurs, the TMR6_&lt;t&gt;_PWMB port output inverted level.</li> </ul> | RW     |
| 3:2 | STPCB | Counter Stop Port Output State Setting B   | <ul style="list-style-type: none"> <li>● 0b00: When the count stops, the TMR6_&lt;t&gt;_PWMB port output is set low level.</li> <li>● 0b01: When the count stops, the TMR6_&lt;t&gt;_PWMB port output is set high level.</li> <li>● 0b10: When the count stops, the TMR6_&lt;t&gt;_PWMB port output retains its previous state.</li> <li>● 0b11: When the count stops, the TMR6_&lt;t&gt;_PWMB port output retains its previous state.</li> </ul>                     | RW     |
| 1:0 | STACB | Count Start Port Status Setting B          | <ul style="list-style-type: none"> <li>● 0b00: When the count starts, the TMR6_&lt;t&gt;_PWMB port output is set low level.</li> <li>● 0b01: When the count starts, the TMR6_&lt;t&gt;_PWMB port output is set high level.</li> <li>● 0b10: When the count starts, the TMR6_&lt;t&gt;_PWMB port output retains its previous state.</li> <li>● 0b11: When the count starts, the TMR6_&lt;t&gt;_PWMB port output retains its previous state.</li> </ul>                 | RW     |

## 21.5.14 Filter Control Register (TMR6\_FCNGR)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |          |   |          |   |          |   |          |   |          |  |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|----------|---|----------|---|----------|---|----------|---|----------|--|
| 0x015C | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7        | 6 | 5        | 4 | 3        | 2 | 1        | 0 |          |  |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |          |   |          |   |          |   |          |   |          |  |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   | NOFICKGB |   | NOFIENGB |   | Reserved |   | NOFICKGA |   | NOFIENGA |  |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   | RW       |   | RW       |   |          |   | RW       |   | RW       |  |

| Bit  | Name     | Function                            | Description                                                                                                                                                                                                               | Access |
|------|----------|-------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:7 | Reserved | Reserved                            | -                                                                                                                                                                                                                         | -      |
| 6:5  | NOFICKGB | Filter Sampling Clock Selection GB  | <ul style="list-style-type: none"> <li>0b00: PCLK0</li> <li>0b01: PCLK0/4</li> <li>0b10: PCLK0/16</li> <li>0b11: PCLK0/64</li> </ul>                                                                                      | RW     |
| 4    | NOFIENGB | Capture Input Port Filter GB        | <ul style="list-style-type: none"> <li>0b0: The input port filter function for the TMR6_&lt;t&gt;PWMB pin is disabled.</li> <li>0b1: The input port filter function for the TMR6&lt;t&gt;_PWMB pin is enabled.</li> </ul> | RW     |
| 3    | Reserved | Reserved                            | -                                                                                                                                                                                                                         | -      |
| 2:1  | NOFICKGA | Filter Sampling Clock Selection GA  | <ul style="list-style-type: none"> <li>0b00: PCLK0</li> <li>0b01: PCLK0/4</li> <li>0b10: PCLK0/16</li> <li>0b11: PCLK0/64</li> </ul>                                                                                      | RW     |
| 0    | NOFIENGA | Capture Input Port Filter Enable GA | <ul style="list-style-type: none"> <li>0b0: The input port filter function for the TMR6_&lt;t&gt;PWMA pin is disabled.</li> <li>0b1: The input port filter function for the TMR6&lt;t&gt;_PWMA pin is enabled.</li> </ul> | RW     |

## 21.5.15 Valid Period Register (TMR6\_VPERR)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |       |       |          |    |    |    |    |    |    |    |         |         |          |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|-------|-------|----------|----|----|----|----|----|----|----|---------|---------|----------|---|---|---|---|---|---|---|
| 0x0160 | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19    | 18    | 17       | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9       | 8       | 7        | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |       |       |          |    |    |    |    |    |    |    |         |         |          |   |   |   |   |   |   |   |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    | PCNTS | PCNTE | Reserved |    |    |    |    |    |    |    | SPPERIB | SPPERIA | Reserved |   |   |   |   |   |   |   |
| Access |              |    |    |    |    |    |    |    |    |    |    |    | RW    | RW    |          |    |    |    |    |    |    |    | RW      | RW      |          |   |   |   |   |   |   |   |

| Bit   | Name     | Function                                  | Description                                                                                                                                                                                                                                                                                                                                                                                                        | Access |
|-------|----------|-------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:21 | Reserved | Reserved                                  | -                                                                                                                                                                                                                                                                                                                                                                                                                  | -      |
| 20:18 | PCNTS    | Active Period Selection                   | <ul style="list-style-type: none"> <li>0b000: Active period selection function is disabled</li> <li>0b001: Active once every 1 period</li> <li>0b010: Active once every 2 period</li> <li>0b011: Active once every 3 period</li> <li>0b100: Active once every 4 period</li> <li>0b101: Active once every 5 period</li> <li>0b110: Active once every 6 period</li> <li>0b111: Active once every 7 period</li> </ul> | RW     |
| 17:16 | PCNTE    | Active Cycle Counting Condition Selection | <ul style="list-style-type: none"> <li>0b00: Active period selection function is disabled</li> <li>0b01: Sawtooth wave count overflow/under-flow or triangle wave trough as count condition</li> <li>0b10: Sawtooth wave count overflow/under-flow or triangle wave peak as count condition</li> <li>0b11: Sawtooth wave count overflow/under-flow or triangle wave trough/peak as count condition</li> </ul>      | RW     |
| 15:10 | Reserved | Reserved                                  | -                                                                                                                                                                                                                                                                                                                                                                                                                  | -      |
| 9     | SPPERIB  | Special Signal Active Period Selection B  | <ul style="list-style-type: none"> <li>0b0: Active period selection function disabled</li> <li>0b1: Active period selection function enabled</li> </ul>                                                                                                                                                                                                                                                            | RW     |
| 8     | SPPERIA  | Special Signal Active Period Selection A  | <ul style="list-style-type: none"> <li>0b0: Active period selection function disabled</li> <li>0b1: Active period selection function enabled</li> </ul>                                                                                                                                                                                                                                                            | RW     |
| 7:0   | Reserved | Reserved                                  | -                                                                                                                                                                                                                                                                                                                                                                                                                  | -      |

## 21.5.16 Status Flag Register (TMR6\_STFLR)

| Offset | Bit Position |          |    |       |       |          |    |         |    |          |    |    |    |    |    |    |    |        |        |        |        |      |      |      |      |      |      |      |      |      |    |    |    |
|--------|--------------|----------|----|-------|-------|----------|----|---------|----|----------|----|----|----|----|----|----|----|--------|--------|--------|--------|------|------|------|------|------|------|------|------|------|----|----|----|
| 0x0164 | 31           | 30       | 29 | 28    | 27    | 26       | 25 | 24      | 23 | 22       | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14     | 13     | 12     | 11     | 10   | 9    | 8    | 7    | 6    | 5    | 4    | 3    | 2    | 1  | 0  |    |
| Reset  | 0x80000000   |          |    |       |       |          |    |         |    |          |    |    |    |    |    |    |    |        |        |        |        |      |      |      |      |      |      |      |      |      |    |    |    |
| Name   | DIRF         | Reserved |    | CMBF2 | CMAF2 | Reserved |    | VPERNUM |    | Reserved |    |    |    |    |    |    |    | CMSBDF | CMSBUF | CMSADF | CMSAUF | DTEF | UDFF | OVFF | CMFF | CMEF | CMDF | CMCF | CMBF | CMAF |    |    |    |
| Access | RO           |          |    | RW    | RW    |          |    |         | RO |          |    |    |    |    |    |    |    |        |        |        | RW     | RW   | RW   | RW   | RW   | RW   | RW   | RW   | RW   | RW   | RW | RW | RW |

| Bit   | Name     | Function                                        | Description                                                                                                                                                                                                                                                                | Access |
|-------|----------|-------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31    | DIRF     | Count direction                                 | <ul style="list-style-type: none"> <li>0b0: Decremental counting.</li> <li>0b1: Incremental counting.</li> </ul>                                                                                                                                                           | RO     |
| 30:28 | Reserved | Reserved                                        | -                                                                                                                                                                                                                                                                          | -      |
| 27    | CMBF2    | Recapture B                                     | <ul style="list-style-type: none"> <li>0b0: No capture completion occurs on TMR6_&lt;t&gt;PWMB after the count match B flag (CMBF) is set to 1</li> <li>0b1: TMR6&lt;t&gt;_PWMB capture completion occurs again after the count match B flag (CMBF) is set to 1</li> </ul> | RW     |
| 26    | CMAF2    | Recapture A                                     | <ul style="list-style-type: none"> <li>0b0: No capture completion occurs on TMR6_&lt;t&gt;PWMA after the count match A flag (CMAF) is set to 1</li> <li>0b1: TMR6&lt;t&gt;_PWMA capture completion occurs again after the count match A flag (CMAF) is set to 1</li> </ul> | RW     |
| 25:24 | Reserved | Reserved                                        | -                                                                                                                                                                                                                                                                          | -      |
| 23:21 | VPERNUM  | Number of periods                               | When the active period selection function is enabled, the number of periods after counting                                                                                                                                                                                 | RO     |
| 20:13 | Reserved | Reserved                                        | -                                                                                                                                                                                                                                                                          | -      |
| 12    | CMSBDF   | Down-counting Special Compare Reference Match B | <ul style="list-style-type: none"> <li>0b0: When counting down, the value of the SCMBR register does not match the counter value.</li> <li>0b1: When counting down, the value of the SCMBR register matches the counter value.</li> </ul>                                  | RW     |
| 11    | CMSBUF   | Up-counting Special Compare Reference Match B   | <ul style="list-style-type: none"> <li>0b0: When counting up, the value of the SCMBR register does not match the counter value.</li> <li>0b1: When counting up, the value of the SCMBR register matches the counter value.</li> </ul>                                      | RW     |
| 10    | CMSADF   | Down-counting Special Compare Reference Match A | <ul style="list-style-type: none"> <li>0b0: When counting down, the value of the SCMAR register does not match the counter value.</li> <li>0b1: When counting down, the value of the SCMAR register matches the counter value.</li> </ul>                                  | RW     |

| Bit | Name   | Function                                      | Description                                                                                                                                                                                                                                                                                                                         | Access |
|-----|--------|-----------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 9   | CMSAUF | Up-counting Special Compare Reference Match A | <ul style="list-style-type: none"> <li>0b0: When counting up, the value of the SCMAR register does not match the counter value.</li> <li>0b1: When counting up, the value of the SCMAR register matches the counter value.</li> </ul>                                                                                               | RW     |
| 8   | DTEF   | Dead Zone Time Error                          | <ul style="list-style-type: none"> <li>0b0: No dead zone time error occurred</li> <li>0b1: Dead zone time error occurred</li> </ul>                                                                                                                                                                                                 | RW     |
| 7   | UDFF   | Down Under-flow Match                         | <ul style="list-style-type: none"> <li>0b0: No sawtooth wave down underflow or triangular wave counting to the valley point occurred</li> <li>0b1: Sawtooth wave down underflow or triangular wave counting to the valley point occurred</li> </ul>                                                                                 | RW     |
| 6   | OVFF   | Up Overflow Match                             | <ul style="list-style-type: none"> <li>0b0: No sawtooth wave up overflow or triangular wave counting to the peak point occurred</li> <li>0b1: Sawtooth wave up overflow or triangular wave counting to the peak point occurred</li> </ul>                                                                                           | RW     |
| 5   | CMFF   | Count Match F                                 | <ul style="list-style-type: none"> <li>0b0: The value in the GCMFR register does not match the count value</li> <li>0b1: The value in the GCMFR register matches the count value</li> </ul>                                                                                                                                         | RW     |
| 4   | CMEF   | Count Match E                                 | <ul style="list-style-type: none"> <li>0b0: The value in the GCMER register does not match the count value</li> <li>0b1: The value in the GCMER register matches the count value</li> </ul>                                                                                                                                         | RW     |
| 3   | CMDF   | Count Match D                                 | <ul style="list-style-type: none"> <li>0b0: The value in the GCMDR register does not match the count value</li> <li>0b1: The value in the GCMDR register matches the count value</li> </ul>                                                                                                                                         | RW     |
| 2   | CMCF   | Count Match C                                 | <ul style="list-style-type: none"> <li>0b0: The value in the GCMCR register does not match the count value</li> <li>0b1: The value in the GCMCR register matches the count value</li> </ul>                                                                                                                                         | RW     |
| 1   | CMBF   | Counter Match B                               | <ul style="list-style-type: none"> <li>0b0: The value in the GCMBR register does not match the counter value, and no TMR6_&lt;t&gt;_PWMB capture completion event has occurred.</li> <li>0b1: The value in the GCMBR register matches the counter value, or a TMR6_&lt;t&gt;_PWMB capture completion event has occurred.</li> </ul> | RW     |
| 0   | CMAF   | Counter Match A                               | <ul style="list-style-type: none"> <li>0b0: The value in the GCMAR register does not match the counter value, and no TMR6_&lt;t&gt;_PWMA capture completion event has occurred.</li> <li>0b1: The value in the GCMAR register matches the counter value, or a TMR6_&lt;t&gt;_PWMA capture completion event has occurred.</li> </ul> | RW     |

## 21.5.17 Hardware Start Event Selection Register (TMR6\_HSTAR)

| Offset | Bit Position |    |    |    |    |    |    |    |        |        |        |        |        |        |        |        |          |    |    |    |    |    |        |        |       |       |      |          |   |   |   |       |       |       |       |    |    |
|--------|--------------|----|----|----|----|----|----|----|--------|--------|--------|--------|--------|--------|--------|--------|----------|----|----|----|----|----|--------|--------|-------|-------|------|----------|---|---|---|-------|-------|-------|-------|----|----|
| 0x0180 | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23     | 22     | 21     | 20     | 19     | 18     | 17     | 16     | 15       | 14 | 13 | 12 | 11 | 10 | 9      | 8      | 7     | 6     | 5    | 4        | 3 | 2 | 1 | 0     |       |       |       |    |    |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |        |        |        |        |        |        |        |        |          |    |    |    |    |    |        |        |       |       |      |          |   |   |   |       |       |       |       |    |    |
| Name   | Reserved     |    |    |    |    |    |    |    | HSTA23 | HSTA22 | HSTA21 | HSTA20 | HSTA19 | HSTA18 | HSTA17 | HSTA16 | Reserved |    |    |    |    |    | HSTA11 | HSTA10 | HSTA9 | HSTA8 | STAS | Reserved |   |   |   | HSTA3 | HSTA2 | HSTA1 | HSTA0 |    |    |
| Access |              |    |    |    |    |    |    |    | RW     | RW     | RW     | RW     | RW     | RW     | RW     | RW     |          |    |    |    |    |    |        | RW     | RW    | RW    | RW   | RW       |   |   |   |       |       | RW    | RW    | RW | RW |

| Bit   | Name     | Function                    | Description                                                                                                                                                                                                                                        | Access |
|-------|----------|-----------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:24 | Reserved | Reserved                    | -                                                                                                                                                                                                                                                  | -      |
| 23    | HSTA23   | Hardware Start Condition 23 | Condition: Falling edge detected on the TMR6_TRIGD pin <ul style="list-style-type: none"> <li>0b0: When the condition is matched, hardware start is disabled</li> <li>0b1: When the condition is matched, the hardware start is enabled</li> </ul> | RW     |
| 22    | HSTA22   | Hardware Start Condition 22 | Condition: Rising edge detected on the TMR6_TRIGD pin <ul style="list-style-type: none"> <li>0b0: When the condition is matched, hardware start is disabled</li> <li>0b1: When the condition is matched, the hardware start is enabled</li> </ul>  | RW     |
| 21    | HSTA21   | Hardware Start Condition 21 | Condition: Falling edge detected on the TMR6_TRIGC pin <ul style="list-style-type: none"> <li>0b0: When the condition is matched, hardware start is disabled</li> <li>0b1: When the condition is matched, the hardware start is enabled</li> </ul> | RW     |
| 20    | HSTA20   | Hardware Start Condition 20 | Condition: Rising edge detected on the TMR6_TRIGC pin <ul style="list-style-type: none"> <li>0b0: When the condition is matched, hardware start is disabled</li> <li>0b1: When the condition is matched, the hardware start is enabled</li> </ul>  | RW     |
| 19    | HSTA19   | Hardware Start Condition 19 | Condition: Falling edge detected on the TMR6_TRIGB pin <ul style="list-style-type: none"> <li>0b0: When the condition is matched, hardware start is disabled</li> <li>0b1: When the condition is matched, the hardware start is enabled</li> </ul> | RW     |

| Bit   | Name     | Function                    | Description                                                                                                                                                                                                                                           | Access |
|-------|----------|-----------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 18    | HSTA18   | Hardware Start Condition 18 | Condition: Rising edge detected on the TMR6_TRIGB pin<br><ul style="list-style-type: none"> <li>0b0: When the condition is matched, hardware start is disabled</li> <li>0b1: When the condition is matched, the hardware start is enabled</li> </ul>  | RW     |
| 17    | HSTA17   | Hardware Start Condition 17 | Condition: Falling edge detected on the TMR6_TRIGA pin<br><ul style="list-style-type: none"> <li>0b0: When the condition is matched, hardware start is disabled</li> <li>0b1: When the condition is matched, the hardware start is enabled</li> </ul> | RW     |
| 16    | HSTA16   | Hardware Start Condition 16 | Condition: Rising edge detected on the TMR6_TRIGA pin<br><ul style="list-style-type: none"> <li>0b0: When the condition is matched, hardware start is disabled</li> <li>0b1: When the condition is matched, the hardware start is enabled</li> </ul>  | RW     |
| 15:12 | Reserved | Reserved                    | -                                                                                                                                                                                                                                                     | -      |
| 11    | HSTA11   | Hardware Start Condition 11 | Condition: Internal hardware trigger event 3 enabled<br><ul style="list-style-type: none"> <li>0b0: When the condition is matched, hardware start is disabled</li> <li>0b1: When the condition is matched, the hardware start is enabled</li> </ul>   | RW     |
| 10    | HSTA10   | Hardware Start Condition 10 | Condition: Internal hardware trigger event 2 enabled<br><ul style="list-style-type: none"> <li>0b0: When the condition is matched, hardware start is disabled</li> <li>0b1: When the condition is matched, the hardware start is enabled</li> </ul>   | RW     |
| 9     | HSTA9    | Hardware Start Condition 9  | Condition: Internal hardware trigger event 1 enabled<br><ul style="list-style-type: none"> <li>0b0: When the condition is matched, hardware start is disabled</li> <li>0b1: When the condition is matched, the hardware start is enabled</li> </ul>   | RW     |
| 8     | HSTA8    | Hardware Start Condition 8  | Condition: Internal hardware trigger event 0 enabled<br><ul style="list-style-type: none"> <li>0b0: When the condition is matched, hardware start is disabled</li> <li>0b1: When the condition is matched, the hardware start is enabled</li> </ul>   | RW     |

| Bit | Name            | Function                   | Description                                                                                                                                                                                                                                                                                                             | Access |
|-----|-----------------|----------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 7   | STAS            | Hardware Start Enable      | <ul style="list-style-type: none"> <li>0b0: Hardware start is disabled</li> <li>0b1: Hardware start is enabled</li> </ul> <div>  <b>Note</b><br/>           When hardware start is enabled, SSTAR settings are ignored.         </div> | RW     |
| 6:4 | <i>Reserved</i> | <i>Reserved</i>            | -                                                                                                                                                                                                                                                                                                                       | -      |
| 3   | HSTA3           | Hardware Start Condition 3 | Condition: TMR6_<t>_PWMB port is sampled to the falling edge <ul style="list-style-type: none"> <li>0b0: When the condition is matched, hardware start is disabled</li> <li>0b1: When the condition is matched, the hardware start is enabled</li> </ul>                                                                | RW     |
| 2   | HSTA2           | Hardware Start Condition 2 | Condition: TMR6_<t>_PWMB port is sampled to the rising edge <ul style="list-style-type: none"> <li>0b0: When the condition is matched, hardware start is disabled</li> <li>0b1: When the condition is matched, the hardware start is enabled</li> </ul>                                                                 | RW     |
| 1   | HSTA1           | Hardware Start Condition 1 | Condition: TMR6_<t>_PWMA port is sampled to the falling edge <ul style="list-style-type: none"> <li>0b0: When the condition is matched, hardware start is disabled</li> <li>0b1: When the condition is matched, the hardware start is enabled</li> </ul>                                                                | RW     |
| 0   | HSTA0           | Hardware Start Condition 0 | Condition: TMR6_<t>_PWMA port is sampled to the rising edge <ul style="list-style-type: none"> <li>0b0: When the condition is matched, hardware start is disabled</li> <li>0b1: When the condition is matched, the hardware start is enabled</li> </ul>                                                                 | RW     |



## 21.5.18 Hardware Stop Event Selection Register (TMR6\_HSTPR)

| Offset | Bit Position |    |    |    |    |    |    |    |        |        |        |        |        |        |        |        |          |    |    |    |    |    |   |   |        |        |       |       |      |          |   |   |  |  |  |  |  |
|--------|--------------|----|----|----|----|----|----|----|--------|--------|--------|--------|--------|--------|--------|--------|----------|----|----|----|----|----|---|---|--------|--------|-------|-------|------|----------|---|---|--|--|--|--|--|
| 0x0184 | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23     | 22     | 21     | 20     | 19     | 18     | 17     | 16     | 15       | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7      | 6      | 5     | 4     | 3    | 2        | 1 | 0 |  |  |  |  |  |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |        |        |        |        |        |        |        |        |          |    |    |    |    |    |   |   |        |        |       |       |      |          |   |   |  |  |  |  |  |
| Name   | Reserved     |    |    |    |    |    |    |    | HSTP23 | HSTP22 | HSTP21 | HSTP20 | HSTP19 | HSTP18 | HSTP17 | HSTP16 | Reserved |    |    |    |    |    |   |   | HSTP11 | HSTP10 | HSTP9 | HSTP8 | STPS | Reserved |   |   |  |  |  |  |  |
| Access |              |    |    |    |    |    |    |    | RW     | RW     | RW     | RW     | RW     | RW     | RW     | RW     |          |    |    |    |    |    |   |   | RW     | RW     | RW    | RW    | RW   |          |   |   |  |  |  |  |  |

| Bit   | Name     | Function                   | Description                                                                                                                                                                                                                                            | Access |
|-------|----------|----------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:24 | Reserved | Reserved                   | -                                                                                                                                                                                                                                                      | -      |
| 23    | HSTP23   | Hardware Stop Condition 23 | Condition: TMR6_TRIGD port is sampled to the falling edge<br><ul style="list-style-type: none"> <li>0b0: When the condition is matched, hardware stop is disabled</li> <li>0b1: When the condition is matched, the hardware stop is enabled</li> </ul> | RW     |
| 22    | HSTP22   | Hardware Stop Condition 22 | Condition: TMR6_TRIGD port is sampled to the rising edge<br><ul style="list-style-type: none"> <li>0b0: When the condition is matched, hardware stop is disabled</li> <li>0b1: When the condition is matched, the hardware stop is enabled</li> </ul>  | RW     |
| 21    | HSTP21   | Hardware Stop Condition 21 | Condition: TMR6_TRIGC port is sampled to the falling edge<br><ul style="list-style-type: none"> <li>0b0: When the condition is matched, hardware stop is disabled</li> <li>0b1: When the condition is matched, the hardware stop is enabled</li> </ul> | RW     |
| 20    | HSTP20   | Hardware Stop Condition 20 | Condition: TMR6_TRIGC port is sampled to the rising edge<br><ul style="list-style-type: none"> <li>0b0: When the condition is matched, hardware stop is disabled</li> <li>0b1: When the condition is matched, the hardware stop is enabled</li> </ul>  | RW     |
| 19    | HSTP19   | Hardware Stop Condition 19 | Condition: TMR6_TRIGB port is sampled to the falling edge<br><ul style="list-style-type: none"> <li>0b0: When the condition is matched, hardware stop is disabled</li> <li>0b1: When the condition is matched, the hardware stop is enabled</li> </ul> | RW     |

| Bit   | Name            | Function                   | Description                                                                                                                                                                                                                                            | Access |
|-------|-----------------|----------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 18    | HSTP18          | Hardware Stop Condition 18 | Condition: TMR6_TRIGB port is sampled to the rising edge<br><ul style="list-style-type: none"> <li>0b0: When the condition is matched, hardware stop is disabled</li> <li>0b1: When the condition is matched, the hardware stop is enabled</li> </ul>  | RW     |
| 17    | HSTP17          | Hardware Stop Condition 17 | Condition: TMR6_TRIGA port is sampled to the falling edge<br><ul style="list-style-type: none"> <li>0b0: When the condition is matched, hardware stop is disabled</li> <li>0b1: When the condition is matched, the hardware stop is enabled</li> </ul> | RW     |
| 16    | HSTP16          | Hardware Stop Condition 16 | Condition: TMR6_TRIGA port is sampled to the rising edge<br><ul style="list-style-type: none"> <li>0b0: When the condition is matched, hardware stop is disabled</li> <li>0b1: When the condition is matched, the hardware stop is enabled</li> </ul>  | RW     |
| 15:12 | <i>Reserved</i> | <i>Reserved</i>            | -                                                                                                                                                                                                                                                      | -      |
| 11    | HSTP11          | Hardware Stop Condition 11 | Condition: Internal hardware trigger event 3 enabled<br><ul style="list-style-type: none"> <li>0b0: When the condition is matched, hardware stop is disabled</li> <li>0b1: When the condition is matched, the hardware stop is enabled</li> </ul>      | RW     |
| 10    | HSTP10          | Hardware Stop Condition 10 | Condition: Internal hardware trigger event 2 enabled<br><ul style="list-style-type: none"> <li>0b0: When the condition is matched, hardware stop is disabled</li> <li>0b1: When the condition is matched, the hardware stop is enabled</li> </ul>      | RW     |
| 9     | HSTP9           | Hardware Stop Condition 9  | Condition: Internal hardware trigger event 1 enabled<br><ul style="list-style-type: none"> <li>0b0: When the condition is matched, hardware stop is disabled</li> <li>0b1: When the condition is matched, the hardware stop is enabled</li> </ul>      | RW     |
| 8     | HSTP8           | Hardware Stop Condition 8  | Condition: Internal hardware trigger event 0 enabled<br><ul style="list-style-type: none"> <li>0b0: When the condition is matched, hardware stop is disabled</li> <li>0b1: When the condition is matched, the hardware stop is enabled</li> </ul>      | RW     |

| Bit | Name            | Function                  | Description                                                                                                                                                                                                                                                                                                               | Access |
|-----|-----------------|---------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 7   | STPS            | Hardware stop Enabled     | <ul style="list-style-type: none"> <li>0b0: Hardware stop is disabled</li> <li>0b1: Hardware stop is enabled</li> </ul> <div>  <b>Note</b><br/>           When hardware stop is enabled, the SSTPR setting has no effect.         </div> | RW     |
| 6:4 | <i>Reserved</i> | <i>Reserved</i>           | -                                                                                                                                                                                                                                                                                                                         | -      |
| 3   | HSTP3           | Hardware Stop Condition 3 | Condition: TMR6_<t>_PWMB port is sampled to the falling edge <ul style="list-style-type: none"> <li>0b0: When the condition is matched, hardware stop is disabled</li> <li>0b1: When the condition is matched, the hardware stop is enabled</li> </ul>                                                                    | RW     |
| 2   | HSTP2           | Hardware Stop Condition 2 | Condition: TMR6_<t>_PWMB port is sampled to the rising edge <ul style="list-style-type: none"> <li>0b0: When the condition is matched, hardware stop is disabled</li> <li>0b1: When the condition is matched, the hardware stop is enabled</li> </ul>                                                                     | RW     |
| 1   | HSTP1           | Hardware Stop Condition 1 | Condition: TMR6_<t>_PWMA port is sampled to the falling edge <ul style="list-style-type: none"> <li>0b0: When the condition is matched, hardware stop is disabled</li> <li>0b1: When the condition is matched, the hardware stop is enabled</li> </ul>                                                                    | RW     |
| 0   | HSTP0           | Hardware Stop Condition 0 | Condition: TMR6_<t>_PWMA port is sampled to the rising edge <ul style="list-style-type: none"> <li>0b0: When the condition is matched, hardware stop is disabled</li> <li>0b1: When the condition is matched, the hardware stop is enabled</li> </ul>                                                                     | RW     |

## 21.5.19 Hardware Clear Event Selection Register (TMR6\_HCLRR)

| Offset | Bit Position |    |    |    |    |    |    |    |        |        |        |        |        |        |        |        |
|--------|--------------|----|----|----|----|----|----|----|--------|--------|--------|--------|--------|--------|--------|--------|
| 0x0188 | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23     | 22     | 21     | 20     | 19     | 18     | 17     | 16     |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |        |        |        |        |        |        |        |        |
| Name   | Reserved     |    |    |    |    |    |    |    | HCLE23 | HCLE22 | HCLE21 | HCLE20 | HCLE19 | HCLE18 | HCLE17 | HCLE16 |
| Access |              |    |    |    |    |    |    |    | RW     | RW     | RW     | RW     | RW     | RW     | RW     | RW     |

| Bit   | Name     | Function                    | Description                                                                                                                                                                                                                                              | Access |
|-------|----------|-----------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:24 | Reserved | Reserved                    | -                                                                                                                                                                                                                                                        | -      |
| 23    | HCLE23   | Hardware Clear Condition 23 | Condition: TMR6_TRIGD port is sampled to the falling edge<br><ul style="list-style-type: none"> <li>0b0: When the condition is matched, hardware clear is disabled</li> <li>0b1: When the condition is matched, the hardware clear is enabled</li> </ul> | RW     |
| 22    | HCLE22   | Hardware Clear Condition 22 | Condition: TMR6_TRIGD port is sampled to the rising edge<br><ul style="list-style-type: none"> <li>0b0: When the condition is matched, hardware clear is disabled</li> <li>0b1: When the condition is matched, the hardware clear is enabled</li> </ul>  | RW     |
| 21    | HCLE21   | Hardware Clear Condition 21 | Condition: TMR6_TRIGC port is sampled to the falling edge<br><ul style="list-style-type: none"> <li>0b0: When the condition is matched, hardware clear is disabled</li> <li>0b1: When the condition is matched, the hardware clear is enabled</li> </ul> | RW     |
| 20    | HCLE20   | Hardware Clear Condition 20 | Condition: TMR6_TRIGC port is sampled to the rising edge<br><ul style="list-style-type: none"> <li>0b0: When the condition is matched, hardware clear is disabled</li> <li>0b1: When the condition is matched, the hardware clear is enabled</li> </ul>  | RW     |
| 19    | HCLE19   | Hardware Clear Condition 19 | Condition: TMR6_TRIGB port is sampled to the falling edge<br><ul style="list-style-type: none"> <li>0b0: When the condition is matched, hardware clear is disabled</li> <li>0b1: When the condition is matched, the hardware clear is enabled</li> </ul> | RW     |

| Bit   | Name     | Function                    | Description                                                                                                                                                                                                                                              | Access |
|-------|----------|-----------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 18    | HCLE18   | Hardware Clear Condition 18 | Condition: TMR6_TRIGB port is sampled to the rising edge<br><ul style="list-style-type: none"> <li>0b0: When the condition is matched, hardware clear is disabled</li> <li>0b1: When the condition is matched, the hardware clear is enabled</li> </ul>  | RW     |
| 17    | HCLE17   | Hardware Clear Condition 17 | Condition: TMR6_TRIGA port is sampled to the falling edge<br><ul style="list-style-type: none"> <li>0b0: When the condition is matched, hardware clear is disabled</li> <li>0b1: When the condition is matched, the hardware clear is enabled</li> </ul> | RW     |
| 16    | HCLE16   | Hardware Clear Condition 16 | Condition: TMR6_TRIGA port is sampled to the rising edge<br><ul style="list-style-type: none"> <li>0b0: When the condition is matched, hardware clear is disabled</li> <li>0b1: When the condition is matched, the hardware clear is enabled</li> </ul>  | RW     |
| 15:12 | Reserved | Reserved                    | -                                                                                                                                                                                                                                                        | -      |
| 11    | HCLE11   | Hardware Clear Condition 11 | Condition: Internal hardware trigger event 3 is enable<br><ul style="list-style-type: none"> <li>0b0: When the condition is matched, hardware clear is disabled</li> <li>0b1: When the condition is matched, the hardware clear is enabled</li> </ul>    | RW     |
| 10    | HCLE10   | Hardware Clear Condition 10 | Condition: Internal hardware trigger event 2 is enable<br><ul style="list-style-type: none"> <li>0b0: When the condition is matched, hardware clear is disabled</li> <li>0b1: When the condition is matched, the hardware clear is enabled</li> </ul>    | RW     |
| 9     | HCLE9    | Hardware Clear Condition 9  | Condition: Internal Hardware Trigger Event 1 Active<br><ul style="list-style-type: none"> <li>0b0: When the condition is matched, hardware clear is disabled</li> <li>0b1: When the condition is matched, the hardware clear is enabled</li> </ul>       | RW     |
| 8     | HCLE8    | Hardware Clear Condition 8  | Condition: Internal Hardware Trigger Event 0 Active<br><ul style="list-style-type: none"> <li>0b0: When the condition is matched, hardware clear is disabled</li> <li>0b1: When the condition is matched, the hardware clear is enabled</li> </ul>       | RW     |

| Bit | Name            | Function                   | Description                                                                                                                                                                                                                                                                                                                   | Access |
|-----|-----------------|----------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 7   | CLES            | Hardware clear enable      | <ul style="list-style-type: none"> <li>0b0: Hardware clear is disabled</li> <li>0b1: Hardware clear is enabled</li> </ul> <div>  <b>Note</b><br/>           When hardware clear is enabled, the setting of SCLRR is disabled.         </div> | RW     |
| 6:4 | <i>Reserved</i> | <i>Reserved</i>            | -                                                                                                                                                                                                                                                                                                                             | -      |
| 3   | HCLE3           | Hardware Clear Condition 3 | Condition: TMR6_<t>_PWMB port is sampled to the falling edge <ul style="list-style-type: none"> <li>0b0: When the condition is matched, hardware clear is disabled</li> <li>0b1: When the condition is matched, the hardware clear is enabled</li> </ul>                                                                      | RW     |
| 2   | HCLE2           | Hardware Clear Condition 2 | Condition: TMR6_<t>_PWMB port is sampled to the rising edge <ul style="list-style-type: none"> <li>0b0: When the condition is matched, hardware clear is disabled</li> <li>0b1: When the condition is matched, the hardware clear is enabled</li> </ul>                                                                       | RW     |
| 1   | HCLE1           | Hardware Clear Condition 1 | Condition: TMR6_<t>_PWMA port is sampled to the falling edge <ul style="list-style-type: none"> <li>0b0: When the condition is matched, hardware clear is disabled</li> <li>0b1: When the condition is matched, the hardware clear is enabled</li> </ul>                                                                      | RW     |
| 0   | HCLE0           | Hardware Clear Condition 0 | Condition: TMR6_<t>_PWMA port is sampled to the rising edge <ul style="list-style-type: none"> <li>0b0: When the condition is matched, hardware clear is disabled</li> <li>0b1: When the condition is matched, the hardware clear is enabled</li> </ul>                                                                       | RW     |

## 21.5.20 Hardware Refresh Event Selection Register (TMR6\_HUPDR)

| Offset | Bit Position |    |    |    |    |    |    |    |        |        |        |        |        |        |        |        |          |    |    |    |        |        |       |       |      |          |   |   |   |       |       |       |       |
|--------|--------------|----|----|----|----|----|----|----|--------|--------|--------|--------|--------|--------|--------|--------|----------|----|----|----|--------|--------|-------|-------|------|----------|---|---|---|-------|-------|-------|-------|
| 0x018C | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23     | 22     | 21     | 20     | 19     | 18     | 17     | 16     | 15       | 14 | 13 | 12 | 11     | 10     | 9     | 8     | 7    | 6        | 5 | 4 | 3 | 2     | 1     | 0     |       |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |        |        |        |        |        |        |        |        |          |    |    |    |        |        |       |       |      |          |   |   |   |       |       |       |       |
| Name   | Reserved     |    |    |    |    |    |    |    | HUPD23 | HUPD22 | HUPD21 | HUPD20 | HUPD19 | HUPD18 | HUPD17 | HUPD16 | Reserved |    |    |    | HUPD11 | HUPD10 | HUPD9 | HUPD8 | UPDS | Reserved |   |   |   | HUPD3 | HUPD2 | HUPD1 | HUPD0 |
| Access |              |    |    |    |    |    |    |    | RW     | RW     | RW     | RW     | RW     | RW     | RW     | RW     |          |    |    |    | RW     | RW     | RW    | RW    | RW   |          |   |   |   | RW    | RW    | RW    | RW    |

| Bit   | Name     | Function                       | Description                                                                                                                                                                                                                                  | Access |
|-------|----------|--------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:24 | Reserved | Reserved                       | -                                                                                                                                                                                                                                            | -      |
| 23    | HUPD23   | Hardware re-fresh condition 23 | Condition: TMR6_TRIGD port is sampled to the fall-ing edge <ul style="list-style-type: none"> <li>0b0: Hardware refresh is disabled when the condition is met</li> <li>0b1: Hardware refresh is enabled when the condition is met</li> </ul> | RW     |
| 22    | HUPD22   | Hardware Re-fresh Condition 22 | Condition: Rising edge detected on the TMR6_TRIGD port <ul style="list-style-type: none"> <li>0b0: Hardware refresh is disabled when the condition is met</li> <li>0b1: Hardware refresh is enabled when the condition is met</li> </ul>     | RW     |
| 21    | HUPD21   | Hardware re-fresh condition 21 | Condition: TMR6_TRIGC port is sampled to the fall-ing edge <ul style="list-style-type: none"> <li>0b0: Hardware refresh is disabled when the condition is met</li> <li>0b1: Hardware refresh is enabled when the condition is met</li> </ul> | RW     |
| 20    | HUPD20   | Hardware re-fresh condition 20 | Condition: Rising edge detected on the TMR6_TRIGC port <ul style="list-style-type: none"> <li>0b0: Hardware refresh is disabled when the condition is met</li> <li>0b1: Hardware refresh is enabled when the condition is met</li> </ul>     | RW     |
| 19    | HUPD19   | Hardware re-fresh condition 19 | Condition: TMR6_TRIGB port is sampled to the fall-ing edge <ul style="list-style-type: none"> <li>0b0: Hardware refresh is disabled when the condition is met</li> <li>0b1: Hardware refresh is enabled when the condition is met</li> </ul> | RW     |

| Bit   | Name            | Function                       | Description                                                                                                                                                                                                                                    | Access |
|-------|-----------------|--------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 18    | HUPD18          | Hardware re-fresh condition 18 | Condition: Rising edge detected on the TMR6_TRIGB port<br><ul style="list-style-type: none"> <li>0b0: Hardware refresh is disabled when the condition is met</li> <li>0b1: Hardware refresh is enabled when the condition is met</li> </ul>    | RW     |
| 17    | HUPD17          | Hardware re-fresh condition 17 | Condition: TMR6_TRIGA port is sampled to the falling edge<br><ul style="list-style-type: none"> <li>0b0: Hardware refresh is disabled when the condition is met</li> <li>0b1: Hardware refresh is enabled when the condition is met</li> </ul> | RW     |
| 16    | HUPD16          | Hardware re-fresh condition 16 | Condition: Rising edge detected on the TMR6_TRIGA port<br><ul style="list-style-type: none"> <li>0b0: Hardware refresh is disabled when the condition is met</li> <li>0b1: Hardware refresh is enabled when the condition is met</li> </ul>    | RW     |
| 15:12 | <i>Reserved</i> | <i>Reserved</i>                | -                                                                                                                                                                                                                                              | -      |
| 11    | HUPD11          | Hardware re-fresh condition 11 | Condition: Internal hardware trigger event 3 enabled<br><ul style="list-style-type: none"> <li>0b0: Hardware refresh is disabled when the condition is met</li> <li>0b1: Hardware refresh is enabled when the condition is met</li> </ul>      | RW     |
| 10    | HUPD10          | Hardware re-fresh condition 10 | Condition: Internal hardware trigger event 2 enabled<br><ul style="list-style-type: none"> <li>0b0: Hardware refresh is disabled when the condition is met</li> <li>0b1: Hardware refresh is enabled when the condition is met</li> </ul>      | RW     |
| 9     | HUPD9           | Hardware re-fresh condition 9  | Condition: Internal hardware trigger event 1 enabled<br><ul style="list-style-type: none"> <li>0b0: Hardware refresh is disabled when the condition is met</li> <li>0b1: Hardware refresh is enabled when the condition is met</li> </ul>      | RW     |
| 8     | HUPD8           | Hardware re-fresh condition 8  | Condition: Internal hardware trigger event 0 enabled<br><ul style="list-style-type: none"> <li>0b0: Hardware refresh is disabled when the condition is met</li> <li>0b1: Hardware refresh is enabled when the condition is met</li> </ul>      | RW     |



| Bit | Name            | Function                             | Description                                                                                                                                                                                                                                                                                                                        | Access |
|-----|-----------------|--------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 7   | UPDS            | Hardware re-<br>fresh enable         | <ul style="list-style-type: none"> <li>0b0: Hardware refresh is disabled</li> <li>0b1: Hardware refresh is enabled</li> </ul> <div>  <b>Note</b><br/>           When hardware refresh is enabled, the SUPDR setting has no effect.         </div> | RW     |
| 6:4 | <i>Reserved</i> | <i>Reserved</i>                      | -                                                                                                                                                                                                                                                                                                                                  | -      |
| 3   | HUPD3           | Hardware re-<br>fresh condition<br>3 | Condition: TMR6_<t>_PWMB port is sampled to the falling edge <ul style="list-style-type: none"> <li>0b0: Hardware refresh is disabled when the condition is met</li> <li>0b1: Hardware refresh is enabled when the condition is met</li> </ul>                                                                                     | RW     |
| 2   | HUPD2           | Hardware re-<br>fresh condition<br>2 | Condition: TMR6_<t>_PWMB port is sampled to the rising edge <ul style="list-style-type: none"> <li>0b0: Hardware refresh is disabled when the condition is met</li> <li>0b1: Hardware refresh is enabled when the condition is met</li> </ul>                                                                                      | RW     |
| 1   | HUPD1           | Hardware re-<br>fresh condition<br>1 | Condition: TMR6_<t>_PWMA port is sampled to the falling edge <ul style="list-style-type: none"> <li>0b0: Hardware refresh is disabled when the condition is met</li> <li>0b1: Hardware refresh is enabled when the condition is met</li> </ul>                                                                                     | RW     |
| 0   | HUPD0           | Hardware re-<br>fresh condition<br>0 | Condition: TMR6_<t>_PWMA port is sampled to the rising edge <ul style="list-style-type: none"> <li>0b0: Hardware refresh is disabled when the condition is met</li> <li>0b1: Hardware refresh is enabled when the condition is met</li> </ul>                                                                                      | RW     |

## 21.5.21 Hardware Capture Event Selection Register A (TMR6\_HCPAR)

| Offset | Bit Position |    |    |    |    |    |        |        |        |        |        |        |        |        |        |        |          |    |    |    |    |    |        |        |       |       |          |   |   |   |   |   |       |       |       |       |    |
|--------|--------------|----|----|----|----|----|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|----------|----|----|----|----|----|--------|--------|-------|-------|----------|---|---|---|---|---|-------|-------|-------|-------|----|
| 0x0190 | 31           | 30 | 29 | 28 | 27 | 26 | 25     | 24     | 23     | 22     | 21     | 20     | 19     | 18     | 17     | 16     | 15       | 14 | 13 | 12 | 11 | 10 | 9      | 8      | 7     | 6     | 5        | 4 | 3 | 2 | 1 | 0 |       |       |       |       |    |
| Reset  | 0x00000000   |    |    |    |    |    |        |        |        |        |        |        |        |        |        |        |          |    |    |    |    |    |        |        |       |       |          |   |   |   |   |   |       |       |       |       |    |
| Name   | Reserved     |    |    |    |    |    | HCPA25 | HCPA24 | HCPA23 | HCPA22 | HCPA21 | HCPA20 | HCPA19 | HCPA18 | HCPA17 | HCPA16 | Reserved |    |    |    |    |    | HCPA11 | HCPA10 | HCPA9 | HCPA8 | Reserved |   |   |   |   |   | HCPA3 | HCPA2 | HCPA1 | HCPA0 |    |
| Access |              |    |    |    |    |    | RW     | RW     | RW     | RW     | RW     | RW     | RW     | RW     | RW     | RW     |          |    |    |    |    |    |        | RW     | RW    | RW    | RW       |   |   |   |   |   |       | RW    | RW    | RW    | RW |

| Bit   | Name     | Function                        | Description                                                                                                                                                                                                                                                         | Access |
|-------|----------|---------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:26 | Reserved | Reserved                        | -                                                                                                                                                                                                                                                                   | -      |
| 25    | HCPA25   | Hardware capture A condition 25 | Condition: internal signal TMR6_<t>_XOR upsampled to falling edge <ul style="list-style-type: none"> <li>0b0: When the condition is met, hardware capture A is disabled.</li> <li>0b1: When the condition is met, hardware capture A is enabled.</li> </ul>         | RW     |
| 24    | HCPA24   | Hardware capture A condition 24 | Condition: The internal signal TMR6_<t>_XOR is sampled to the rising edge <ul style="list-style-type: none"> <li>0b0: When the condition is met, hardware capture A is disabled.</li> <li>0b1: When the condition is met, hardware capture A is enabled.</li> </ul> | RW     |
| 23    | HCPA23   | Hardware capture A condition 23 | Condition: Falling edge detected on the TMR6_TRIGD pin <ul style="list-style-type: none"> <li>0b0: When the condition is met, hardware capture A is disabled.</li> <li>0b1: When the condition is met, hardware capture A is enabled.</li> </ul>                    | RW     |
| 22    | HCPA22   | Hardware capture A condition 22 | Condition: Rising edge detected on the TMR6_TRIGD port <ul style="list-style-type: none"> <li>0b0: When the condition is met, hardware capture A is disabled.</li> <li>0b1: When the condition is met, hardware capture A is enabled.</li> </ul>                    | RW     |
| 21    | HCPA21   | Hardware capture A condition 21 | Condition: TMR6_TRIGC port is sampled to the falling edge <ul style="list-style-type: none"> <li>0b0: When the condition is met, hardware capture A is disabled.</li> <li>0b1: When the condition is met, hardware capture A is enabled.</li> </ul>                 | RW     |

| Bit   | Name            | Function                        | Description                                                                                                                                                                                                                                            | Access |
|-------|-----------------|---------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 20    | HCPA20          | Hardware capture A condition 20 | Condition: Rising edge detected on the TMR6_TRIGC port<br><ul style="list-style-type: none"> <li>0b0: When the condition is met, hardware capture A is disabled.</li> <li>0b1: When the condition is met, hardware capture A is enabled.</li> </ul>    | RW     |
| 19    | HCPA19          | Hardware capture A condition 19 | Condition: TMR6_TRIGB port is sampled to the falling edge<br><ul style="list-style-type: none"> <li>0b0: When the condition is met, hardware capture A is disabled.</li> <li>0b1: When the condition is met, hardware capture A is enabled.</li> </ul> | RW     |
| 18    | HCPA18          | Hardware capture A condition 18 | Condition: Rising edge detected on the TMR6_TRIGB port<br><ul style="list-style-type: none"> <li>0b0: When the condition is met, hardware capture A is disabled.</li> <li>0b1: When the condition is met, hardware capture A is enabled.</li> </ul>    | RW     |
| 17    | HCPA17          | Hardware capture A condition 17 | Condition: TMR6_TRIGA port is sampled to the falling edge<br><ul style="list-style-type: none"> <li>0b0: When the condition is met, hardware capture A is disabled.</li> <li>0b1: When the condition is met, hardware capture A is enabled.</li> </ul> | RW     |
| 16    | HCPA16          | Hardware capture A condition 16 | Condition: Rising edge detected on the TMR6_TRIGA port<br><ul style="list-style-type: none"> <li>0b0: When the condition is met, hardware capture A is disabled.</li> <li>0b1: When the condition is met, hardware capture A is enabled.</li> </ul>    | RW     |
| 15:12 | <i>Reserved</i> | <i>Reserved</i>                 | -                                                                                                                                                                                                                                                      | -      |
| 11    | HCPA11          | Hardware capture A condition 11 | Condition: Internal hardware trigger event 3 enabled<br><ul style="list-style-type: none"> <li>0b0: When the condition is met, hardware capture A is disabled.</li> <li>0b1: When the condition is met, hardware capture A is enabled.</li> </ul>      | RW     |
| 10    | HCPA10          | Hardware capture A condition 10 | Condition: Internal hardware trigger event 2 enabled<br><ul style="list-style-type: none"> <li>0b0: When the condition is met, hardware capture A is disabled.</li> <li>0b1: When the condition is met, hardware capture A is enabled.</li> </ul>      | RW     |

| Bit | Name            | Function                       | Description                                                                                                                                                                                                                                               | Access |
|-----|-----------------|--------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 9   | HCPA9           | Hardware capture A condition 9 | Condition: Internal hardware trigger event 1 enabled<br><ul style="list-style-type: none"> <li>0b0: When the condition is met, hardware capture A is disabled.</li> <li>0b1: When the condition is met, hardware capture A is enabled.</li> </ul>         | RW     |
| 8   | HCPA8           | Hardware capture A condition 8 | Condition: Internal hardware trigger event 0 enabled<br><ul style="list-style-type: none"> <li>0b0: When the condition is met, hardware capture A is disabled.</li> <li>0b1: When the condition is met, hardware capture A is enabled.</li> </ul>         | RW     |
| 7:4 | <i>Reserved</i> | <i>Reserved</i>                | -                                                                                                                                                                                                                                                         | -      |
| 3   | HCPA3           | Hardware capture A condition 3 | Condition: TMR6_<t>_PWMB port is sampled to the falling edge<br><ul style="list-style-type: none"> <li>0b0: When the condition is met, hardware capture A is disabled.</li> <li>0b1: When the condition is met, hardware capture A is enabled.</li> </ul> | RW     |
| 2   | HCPA2           | Hardware capture A condition 2 | Condition: TMR6_<t>_PWMB port is sampled to the rising edge<br><ul style="list-style-type: none"> <li>0b0: When the condition is met, hardware capture A is disabled.</li> <li>0b1: When the condition is met, hardware capture A is enabled.</li> </ul>  | RW     |
| 1   | HCPA1           | Hardware capture A condition 1 | Condition: TMR6_<t>_PWMA port is sampled to the falling edge<br><ul style="list-style-type: none"> <li>0b0: When the condition is met, hardware capture A is disabled.</li> <li>0b1: When the condition is met, hardware capture A is enabled.</li> </ul> | RW     |
| 0   | HCPA0           | Hardware capture A condition 0 | Condition: TMR6_<t>_PWMA port is sampled to the rising edge<br><ul style="list-style-type: none"> <li>0b0: When the condition is met, hardware capture A is disabled.</li> <li>0b1: When the condition is met, hardware capture A is enabled.</li> </ul>  | RW     |

**Note**

The internal signal TMR6\_<t>\_XOR is the signal obtained by performing an XOR logic operation on TMR6\_<t>\_PWMA and TMR6\_<t>\_PWMB.

## 21.5.22 Hardware Capture Event Selection Register B (TMR6\_HCPBR)

| Offset | Bit Position |    |    |    |    |    |        |        |        |        |        |        |        |        |        |        |          |    |    |    |    |    |        |        |       |       |          |   |   |   |   |   |       |       |       |       |    |
|--------|--------------|----|----|----|----|----|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|----------|----|----|----|----|----|--------|--------|-------|-------|----------|---|---|---|---|---|-------|-------|-------|-------|----|
| 0x0194 | 31           | 30 | 29 | 28 | 27 | 26 | 25     | 24     | 23     | 22     | 21     | 20     | 19     | 18     | 17     | 16     | 15       | 14 | 13 | 12 | 11 | 10 | 9      | 8      | 7     | 6     | 5        | 4 | 3 | 2 | 1 | 0 |       |       |       |       |    |
| Reset  | 0x00000000   |    |    |    |    |    |        |        |        |        |        |        |        |        |        |        |          |    |    |    |    |    |        |        |       |       |          |   |   |   |   |   |       |       |       |       |    |
| Name   | Reserved     |    |    |    |    |    | HCPB25 | HCPB24 | HCPB23 | HCPB22 | HCPB21 | HCPB20 | HCPB19 | HCPB18 | HCPB17 | HCPB16 | Reserved |    |    |    |    |    | HCPB11 | HCPB10 | HCPB9 | HCPB8 | Reserved |   |   |   |   |   | HCPB3 | HCPB2 | HCPB1 | HCPB0 |    |
| Access |              |    |    |    |    |    | RW     | RW     | RW     | RW     | RW     | RW     | RW     | RW     | RW     | RW     |          |    |    |    |    |    |        | RW     | RW    | RW    | RW       |   |   |   |   |   |       | RW    | RW    | RW    | RW |

| Bit   | Name     | Function                              | Description                                                                                                                                                                                                                                                             | Access |
|-------|----------|---------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:26 | Reserved | Reserved                              | -                                                                                                                                                                                                                                                                       | -      |
| 25    | HCPB25   | Hardware capture event B condition 25 | Condition: The internal signal TMR6_<t>_XOR is sampled on the falling edge. <ul style="list-style-type: none"> <li>0b0: When this condition is met, hardware capture B is disabled.</li> <li>0b1: When this condition is met, hardware capture B is enabled.</li> </ul> | RW     |
| 24    | HCPB24   | Hardware capture event B condition 24 | Condition: The internal signal TMR6_<t>_XOR is sampled to the rising edge <ul style="list-style-type: none"> <li>0b0: When this condition is met, hardware capture B is disabled.</li> <li>0b1: When this condition is met, hardware capture B is enabled.</li> </ul>   | RW     |
| 23    | HCPB23   | Hardware capture event B condition 23 | Condition: Falling edge detected on the TMR6_TRIGD pin <ul style="list-style-type: none"> <li>0b0: When this condition is met, hardware capture B is disabled.</li> <li>0b1: When this condition is met, hardware capture B is enabled.</li> </ul>                      | RW     |
| 22    | HCPB22   | Hardware capture event B condition 22 | Condition: Rising edge detected on the TMR6_TRIGD port <ul style="list-style-type: none"> <li>0b0: When this condition is met, hardware capture B is disabled.</li> <li>0b1: When this condition is met, hardware capture B is enabled.</li> </ul>                      | RW     |
| 21    | HCPB21   | Hardware capture event B condition 21 | Condition: TMR6_TRIGC port is sampled to the falling edge <ul style="list-style-type: none"> <li>0b0: When this condition is met, hardware capture B is disabled.</li> <li>0b1: When this condition is met, hardware capture B is enabled.</li> </ul>                   | RW     |

| Bit   | Name     | Function                              | Description                                                                                                                                                                                                                                           | Access |
|-------|----------|---------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 20    | HCPB20   | Hardware capture event B condition 20 | Condition: Rising edge detected on the TMR6_TRIGC port <ul style="list-style-type: none"> <li>0b0: When this condition is met, hardware capture B is disabled.</li> <li>0b1: When this condition is met, hardware capture B is enabled.</li> </ul>    | RW     |
| 19    | HCPB19   | Hardware capture event B condition 19 | Condition: TMR6_TRIGB port is sampled to the falling edge <ul style="list-style-type: none"> <li>0b0: When this condition is met, hardware capture B is disabled.</li> <li>0b1: When this condition is met, hardware capture B is enabled.</li> </ul> | RW     |
| 18    | HCPB18   | Hardware capture event B condition 18 | Condition: Rising edge detected on the TMR6_TRIGB port <ul style="list-style-type: none"> <li>0b0: When this condition is met, hardware capture B is disabled.</li> <li>0b1: When this condition is met, hardware capture B is enabled.</li> </ul>    | RW     |
| 17    | HCPB17   | Hardware capture event B condition 17 | Condition: TMR6_TRIGA port is sampled to the falling edge <ul style="list-style-type: none"> <li>0b0: When this condition is met, hardware capture B is disabled.</li> <li>0b1: When this condition is met, hardware capture B is enabled.</li> </ul> | RW     |
| 16    | HCPB16   | Hardware capture event B condition 16 | Condition: Rising edge detected on the TMR6_TRIGA port <ul style="list-style-type: none"> <li>0b0: When this condition is met, hardware capture B is disabled.</li> <li>0b1: When this condition is met, hardware capture B is enabled.</li> </ul>    | RW     |
| 15:12 | Reserved | Reserved                              | -                                                                                                                                                                                                                                                     | -      |
| 11    | HCPB11   | Hardware capture event B condition 11 | Condition: Internal hardware trigger event 3 enabled <ul style="list-style-type: none"> <li>0b0: When this condition is met, hardware capture B is disabled.</li> <li>0b1: When this condition is met, hardware capture B is enabled.</li> </ul>      | RW     |
| 10    | HCPB10   | Hardware capture event B condition 10 | Condition: Internal hardware trigger event 2 enabled <ul style="list-style-type: none"> <li>0b0: When this condition is met, hardware capture B is disabled.</li> <li>0b1: When this condition is met, hardware capture B is enabled.</li> </ul>      | RW     |

| Bit | Name            | Function                             | Description                                                                                                                                                                                                                                                     | Access |
|-----|-----------------|--------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 9   | HCPB9           | Hardware capture event B condition 9 | Condition: Internal hardware trigger event 1 enabled<br><ul style="list-style-type: none"> <li>● 0b0: When this condition is met, hardware capture B is disabled.</li> <li>● 0b1: When this condition is met, hardware capture B is enabled.</li> </ul>         | RW     |
| 8   | HCPB8           | Hardware capture event B condition 8 | Condition: Internal hardware trigger event 0 enabled<br><ul style="list-style-type: none"> <li>● 0b0: When this condition is met, hardware capture B is disabled.</li> <li>● 0b1: When this condition is met, hardware capture B is enabled.</li> </ul>         | RW     |
| 7:4 | <i>Reserved</i> | <i>Reserved</i>                      | -                                                                                                                                                                                                                                                               | -      |
| 3   | HCPB3           | Hardware capture event B condition 3 | Condition: TMR6_<t>_PWMB port is sampled to the falling edge<br><ul style="list-style-type: none"> <li>● 0b0: When this condition is met, hardware capture B is disabled.</li> <li>● 0b1: When this condition is met, hardware capture B is enabled.</li> </ul> | RW     |
| 2   | HCPB2           | Hardware capture event B condition 2 | Condition: TMR6_<t>_PWMB port is sampled to the rising edge<br><ul style="list-style-type: none"> <li>● 0b0: When this condition is met, hardware capture B is disabled.</li> <li>● 0b1: When this condition is met, hardware capture B is enabled.</li> </ul>  | RW     |
| 1   | HCPB1           | Hardware capture event B condition 1 | Condition: TMR6_<t>_PWMA port is sampled to the falling edge<br><ul style="list-style-type: none"> <li>● 0b0: When this condition is met, hardware capture B is disabled.</li> <li>● 0b1: When this condition is met, hardware capture B is enabled.</li> </ul> | RW     |
| 0   | HCPB0           | Hardware capture event B condition 0 | Condition: TMR6_<t>_PWMA port is sampled to the rising edge<br><ul style="list-style-type: none"> <li>● 0b0: When this condition is met, hardware capture B is disabled.</li> <li>● 0b1: When this condition is met, hardware capture B is enabled.</li> </ul>  | RW     |

**Note**

The internal signal TMR6\_<t>\_XOR is the signal obtained by performing an XOR logic operation on TMR6\_<t>\_PWMA and TMR6\_<t>\_PWMB.

## 21.5.23 Hardware Count Up Event Select Register (TMR6\_HCUPR)

| Offset | Bit Position |    |    |    |    |    |    |    |        |        |        |        |        |        |        |        |          |    |    |    |    |    |   |   |        |        |       |       |       |       |       |       |       |       |       |       |    |
|--------|--------------|----|----|----|----|----|----|----|--------|--------|--------|--------|--------|--------|--------|--------|----------|----|----|----|----|----|---|---|--------|--------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|----|
| 0x0198 | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23     | 22     | 21     | 20     | 19     | 18     | 17     | 16     | 15       | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7      | 6      | 5     | 4     | 3     | 2     | 1     | 0     |       |       |       |       |    |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |        |        |        |        |        |        |        |        |          |    |    |    |    |    |   |   |        |        |       |       |       |       |       |       |       |       |       |       |    |
| Name   | Reserved     |    |    |    |    |    |    |    | HCUP23 | HCUP22 | HCUP21 | HCUP20 | HCUP19 | HCUP18 | HCUP17 | HCUP16 | Reserved |    |    |    |    |    |   |   | HCUP11 | HCUP10 | HCUP9 | HCUP8 | HCUP7 | HCUP6 | HCUP5 | HCUP4 | HCUP3 | HCUP2 | HCUP1 | HCUP0 |    |
| Access |              |    |    |    |    |    |    |    | RW     | RW     | RW     | RW     | RW     | RW     | RW     | RW     |          |    |    |    |    |    |   |   | RW     | RW     | RW    | RW    | RW    | RW    | RW    | RW    | RW    | RW    | RW    | RW    | RW |

| Bit   | Name     | Function                       | Description                                                                                                                                                                                                                                       | Access |
|-------|----------|--------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:24 | Reserved | Reserved                       | -                                                                                                                                                                                                                                                 | -      |
| 23    | HCUP23   | Hardware Count Up Condition 23 | Condition: Falling edge detected on the TMR6_TRIGD port <ul style="list-style-type: none"> <li>0b0: When this condition is met, hardware count up is disabled.</li> <li>0b1: When this condition is met, hardware count up is enabled.</li> </ul> | RW     |
| 22    | HCUP22   | Hardware Count Up Condition 22 | Condition: Rising edge detected on the TMR6_TRIGD port <ul style="list-style-type: none"> <li>0b0: When this condition is met, hardware count up is disabled.</li> <li>0b1: When this condition is met, hardware count up is enabled.</li> </ul>  | RW     |
| 21    | HCUP21   | Hardware Count Up Condition 21 | Condition: Falling edge detected on the TMR6_TRIGC port <ul style="list-style-type: none"> <li>0b0: When this condition is met, hardware count up is disabled.</li> <li>0b1: When this condition is met, hardware count up is enabled.</li> </ul> | RW     |
| 20    | HCUP20   | Hardware Count Up Condition 20 | Condition: Rising edge detected on the TMR6_TRIGC port <ul style="list-style-type: none"> <li>0b0: When this condition is met, hardware count up is disabled.</li> <li>0b1: When this condition is met, hardware count up is enabled.</li> </ul>  | RW     |
| 19    | HCUP19   | Hardware Count Up Condition 19 | Condition: Falling edge detected on the TMR6_TRIGB port <ul style="list-style-type: none"> <li>0b0: When this condition is met, hardware count up is disabled.</li> <li>0b1: When this condition is met, hardware count up is enabled.</li> </ul> | RW     |



| Bit   | Name            | Function                       | Description                                                                                                                                                                                                                                           | Access |
|-------|-----------------|--------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 18    | HCUP18          | Hardware Count Up Condition 18 | Condition: Rising edge detected on the TMR6_TRIGB port<br><ul style="list-style-type: none"> <li>0b0: When this condition is met, hardware count up is disabled.</li> <li>0b1: When this condition is met, hardware count up is enabled.</li> </ul>   | RW     |
| 17    | HCUP17          | Hardware Count Up Condition 17 | Condition: Falling edge detected on the TMR6_TRI-GA port<br><ul style="list-style-type: none"> <li>0b0: When this condition is met, hardware count up is disabled.</li> <li>0b1: When this condition is met, hardware count up is enabled.</li> </ul> | RW     |
| 16    | HCUP16          | Hardware Count Up Condition 16 | Condition: Rising edge detected on the TMR6_TRI-GA port<br><ul style="list-style-type: none"> <li>0b0: When this condition is met, hardware count up is disabled.</li> <li>0b1: When this condition is met, hardware count up is enabled.</li> </ul>  | RW     |
| 15:12 | <i>Reserved</i> | <i>Reserved</i>                | -                                                                                                                                                                                                                                                     | -      |
| 11    | HCUP11          | Hardware Count Up Condition 11 | Condition: Internal hardware trigger event 3 enabled<br><ul style="list-style-type: none"> <li>0b0: When this condition is met, hardware count up is disabled.</li> <li>0b1: When this condition is met, hardware count up is enabled.</li> </ul>     | RW     |
| 10    | HCUP10          | Hardware Count Up Condition 10 | Condition: Internal hardware trigger event 2 enabled<br><ul style="list-style-type: none"> <li>0b0: When this condition is met, hardware count up is disabled.</li> <li>0b1: When this condition is met, hardware count up is enabled.</li> </ul>     | RW     |
| 9     | HCUP9           | Hardware Count Up Condition 9  | Condition: Internal hardware trigger event 1 enabled<br><ul style="list-style-type: none"> <li>0b0: When this condition is met, hardware count up is disabled.</li> <li>0b1: When this condition is met, hardware count up is enabled.</li> </ul>     | RW     |
| 8     | HCUP8           | Hardware Count Up Condition 8  | Condition: Internal hardware trigger event 0 enabled<br><ul style="list-style-type: none"> <li>0b0: When this condition is met, hardware count up is disabled.</li> <li>0b1: When this condition is met, hardware count up is enabled.</li> </ul>     | RW     |

| Bit | Name  | Function                      | Description                                                                                                                                                                                                                                                                                   | Access |
|-----|-------|-------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 7   | HCUP7 | Hardware Count Up Condition 7 | Condition: When the TMR6_<t>_PWMB port is high, the falling edge detected on the TMR6_<t>_PWMA port <ul style="list-style-type: none"> <li>0b0: When this condition is met, hardware count up is disabled.</li> <li>0b1: When this condition is met, hardware count up is enabled.</li> </ul> | RW     |
| 6   | HCUP6 | Hardware Count Up Condition 6 | Condition: When the TMR6_<t>_PWMB port is high, the rising edge detected on the TMR6_<t>_PWMA port <ul style="list-style-type: none"> <li>0b0: When this condition is met, hardware count up is disabled.</li> <li>0b1: When this condition is met, hardware count up is enabled.</li> </ul>  | RW     |
| 5   | HCUP5 | Hardware Count Up Condition 5 | Condition: When the TMR6_<t>_PWMB port is low, the falling edge detected on the TMR6_<t>_PWMA port <ul style="list-style-type: none"> <li>0b0: When this condition is met, hardware count up is disabled.</li> <li>0b1: When this condition is met, hardware count up is enabled.</li> </ul>  | RW     |
| 4   | HCUP4 | Hardware Count Up Condition 4 | Condition: When the TMR6_<t>_PWMB port is low, the rising edge detected on the TMR6_<t>_PWMA port <ul style="list-style-type: none"> <li>0b0: When this condition is met, hardware count up is disabled.</li> <li>0b1: When this condition is met, hardware count up is enabled.</li> </ul>   | RW     |
| 3   | HCUP3 | Hardware Count Up Condition 3 | Condition: When the TMR6_<t>_PWMA port is high, the falling edge detected on the TMR6_<t>_PWMB port <ul style="list-style-type: none"> <li>0b0: When this condition is met, hardware count up is disabled.</li> <li>0b1: When this condition is met, hardware count up is enabled.</li> </ul> | RW     |
| 2   | HCUP2 | Hardware Count Up Condition 2 | Condition: When the TMR6_<t>_PWMA port is high, the rising edge detected on the TMR6_<t>_PWMB port <ul style="list-style-type: none"> <li>0b0: When this condition is met, hardware count up is disabled.</li> <li>0b1: When this condition is met, hardware count up is enabled.</li> </ul>  | RW     |
| 1   | HCUP1 | Hardware Count Up Condition 1 | Condition: When the TMR6_<t>_PWMA port is low, the falling edge detected on the TMR6_<t>_PWMB port <ul style="list-style-type: none"> <li>0b0: When this condition is met, hardware count up is disabled.</li> <li>0b1: When this condition is met, hardware count up is enabled.</li> </ul>  | RW     |

| Bit | Name  | Function                      | Description                                                                                                                                                                                                                                                                                     | Access |
|-----|-------|-------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 0   | HCUP0 | Hardware Count Up Condition 0 | Condition: When the TMR6_<t>_PWMA port is low, the rising edge detected on the TMR6_<t>_PWMB port <ul style="list-style-type: none"> <li>● 0b0: When this condition is met, hardware count up is disabled.</li> <li>● 0b1: When this condition is met, hardware count up is enabled.</li> </ul> | RW     |

**Note**

In hardware count mode, the GCONR.CKDIV must be set to 0.

## 21.5.24 Hardware Count Down Event Select Register (TMR6\_HCDOR)

| Offset | Bit Position |    |    |    |    |    |    |    |        |        |        |        |        |        |        |        |          |    |    |    |    |    |   |   |        |        |       |       |       |       |       |       |       |       |       |       |
|--------|--------------|----|----|----|----|----|----|----|--------|--------|--------|--------|--------|--------|--------|--------|----------|----|----|----|----|----|---|---|--------|--------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| 0x019C | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23     | 22     | 21     | 20     | 19     | 18     | 17     | 16     | 15       | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7      | 6      | 5     | 4     | 3     | 2     | 1     | 0     |       |       |       |       |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |        |        |        |        |        |        |        |        |          |    |    |    |    |    |   |   |        |        |       |       |       |       |       |       |       |       |       |       |
| Name   | Reserved     |    |    |    |    |    |    |    | HCDO23 | HCDO22 | HCDO21 | HCDO20 | HCDO19 | HCDO18 | HCDO17 | HCDO16 | Reserved |    |    |    |    |    |   |   | HCDO11 | HCDO10 | HCDO9 | HCDO8 | HCDO7 | HCDO6 | HCDO5 | HCDO4 | HCDO3 | HCDO2 | HCDO1 | HCDO0 |
| Access |              |    |    |    |    |    |    |    | RW     | RW     | RW     | RW     | RW     | RW     | RW     | RW     |          |    |    |    |    |    |   |   | RW     | RW     | RW    | RW    | RW    | RW    | RW    | RW    | RW    | RW    | RW    | RW    |

| Bit   | Name     | Function                         | Description                                                                                                                                                                                                                                       | Access |
|-------|----------|----------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:24 | Reserved | Reserved                         | -                                                                                                                                                                                                                                                 | -      |
| 23    | HCDO23   | Hardware Count Down Condition 23 | Condition: Falling edge detected on the TMR6_TRIGD port <ul style="list-style-type: none"> <li>0b0: Hardware count down is disabled when the condition is met.</li> <li>0b1: Hardware count down is enabled when the condition is met.</li> </ul> | RW     |
| 22    | HCDO22   | Hardware Count Down Condition 22 | Condition: Rising edge detected on the TMR6_TRIGD port <ul style="list-style-type: none"> <li>0b0: Hardware count down is disabled when the condition is met.</li> <li>0b1: Hardware count down is enabled when the condition is met.</li> </ul>  | RW     |
| 21    | HCDO21   | Hardware Count Down Condition 21 | Condition: Falling edge detected on the TMR6_TRIGC port <ul style="list-style-type: none"> <li>0b0: Hardware count down is disabled when the condition is met.</li> <li>0b1: Hardware count down is enabled when the condition is met.</li> </ul> | RW     |
| 20    | HCDO20   | Hardware Count Down Condition 20 | Condition: Rising edge detected on the TMR6_TRIGC port <ul style="list-style-type: none"> <li>0b0: Hardware count down is disabled when the condition is met.</li> <li>0b1: Hardware count down is enabled when the condition is met.</li> </ul>  | RW     |
| 19    | HCDO19   | Hardware Count Down Condition 19 | Condition: Falling edge detected on the TMR6_TRIGB port <ul style="list-style-type: none"> <li>0b0: Hardware count down is disabled when the condition is met.</li> <li>0b1: Hardware count down is enabled when the condition is met.</li> </ul> | RW     |

| Bit   | Name            | Function                         | Description                                                                                                                                                                                                                                           | Access |
|-------|-----------------|----------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 18    | HCDO18          | Hardware Count Down Condition 18 | Condition: Rising edge detected on the TMR6_TRIGB port<br><ul style="list-style-type: none"> <li>0b0: Hardware count down is disabled when the condition is met.</li> <li>0b1: Hardware count down is enabled when the condition is met.</li> </ul>   | RW     |
| 17    | HCDO17          | Hardware Count Down Condition 17 | Condition: Falling edge detected on the TMR6_TRI-GA port<br><ul style="list-style-type: none"> <li>0b0: Hardware count down is disabled when the condition is met.</li> <li>0b1: Hardware count down is enabled when the condition is met.</li> </ul> | RW     |
| 16    | HCDO16          | Hardware Count Down Condition 16 | Condition: Rising edge detected on the TMR6_TRI-GA port<br><ul style="list-style-type: none"> <li>0b0: Hardware count down is disabled when the condition is met.</li> <li>0b1: Hardware count down is enabled when the condition is met.</li> </ul>  | RW     |
| 15:12 | <i>Reserved</i> | <i>Reserved</i>                  | -                                                                                                                                                                                                                                                     | -      |
| 11    | HCDO11          | Hardware Count Down Condition 11 | Condition: Internal hardware trigger event 3 enabled<br><ul style="list-style-type: none"> <li>0b0: Hardware count down is disabled when the condition is met.</li> <li>0b1: Hardware count down is enabled when the condition is met.</li> </ul>     | RW     |
| 10    | HCDO10          | Hardware Count Down Condition 10 | Condition: Internal hardware trigger event 2 enabled<br><ul style="list-style-type: none"> <li>0b0: Hardware count down is disabled when the condition is met.</li> <li>0b1: Hardware count down is enabled when the condition is met.</li> </ul>     | RW     |
| 9     | HCDO9           | Hardware Count Down Condition 9  | Condition: Internal hardware trigger event 1 enabled<br><ul style="list-style-type: none"> <li>0b0: Hardware count down is disabled when the condition is met.</li> <li>0b1: Hardware count down is enabled when the condition is met.</li> </ul>     | RW     |
| 8     | HCDO8           | Hardware Count Down Condition 8  | Condition: Internal hardware trigger event 0 enabled<br><ul style="list-style-type: none"> <li>0b0: Hardware count down is disabled when the condition is met.</li> <li>0b1: Hardware count down is enabled when the condition is met.</li> </ul>     | RW     |

| Bit | Name  | Function                        | Description                                                                                                                                                                                                                                                                                       | Access |
|-----|-------|---------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 7   | HCDO7 | Hardware Count Down Condition 7 | Condition: When the TMR6_<t>_PWMB port is high, the falling edge detected on the TMR6_<t>_PWMA port <ul style="list-style-type: none"> <li>● 0b0: Hardware count down is disabled when the condition is met.</li> <li>● 0b1: Hardware count down is enabled when the condition is met.</li> </ul> | RW     |
| 6   | HCDO6 | Hardware Count Down Condition 6 | Condition: When the TMR6_<t>_PWMB port is high, the rising edge detected on the TMR6_<t>_PWMA port <ul style="list-style-type: none"> <li>● 0b0: Hardware count down is disabled when the condition is met.</li> <li>● 0b1: Hardware count down is enabled when the condition is met.</li> </ul>  | RW     |
| 5   | HCDO5 | Hardware Count Down Condition 5 | Condition: When the TMR6_<t>_PWMB port is low, the falling edge detected on the TMR6_<t>_PWMA port <ul style="list-style-type: none"> <li>● 0b0: Hardware count down is disabled when the condition is met.</li> <li>● 0b1: Hardware count down is enabled when the condition is met.</li> </ul>  | RW     |
| 4   | HCDO4 | Hardware Count Down Condition 4 | Condition: When the TMR6_<t>_PWMB port is low, the rising edge detected on the TMR6_<t>_PWMA port <ul style="list-style-type: none"> <li>● 0b0: Hardware count down is disabled when the condition is met.</li> <li>● 0b1: Hardware count down is enabled when the condition is met.</li> </ul>   | RW     |
| 3   | HCDO3 | Hardware Count Down Condition 3 | Condition: When the TMR6_<t>_PWMA port is high, the falling edge detected on the TMR6_<t>_PWMB port <ul style="list-style-type: none"> <li>● 0b0: Hardware count down is disabled when the condition is met.</li> <li>● 0b1: Hardware count down is enabled when the condition is met.</li> </ul> | RW     |
| 2   | HCDO2 | Hardware Count Down Condition 2 | Condition: When the TMR6_<t>_PWMA port is high, the rising edge detected on the TMR6_<t>_PWMB port <ul style="list-style-type: none"> <li>● 0b0: Hardware count down is disabled when the condition is met.</li> <li>● 0b1: Hardware count down is enabled when the condition is met.</li> </ul>  | RW     |
| 1   | HCDO1 | Hardware Count Down Condition 1 | Condition: When the TMR6_<t>_PWMA port is low, the falling edge detected on the TMR6_<t>_PWMB port <ul style="list-style-type: none"> <li>● 0b0: Hardware count down is disabled when the condition is met.</li> <li>● 0b1: Hardware count down is enabled when the condition is met.</li> </ul>  | RW     |

| Bit | Name  | Function                        | Description                                                                                                                                                                                                                                                                                     | Access |
|-----|-------|---------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 0   | HCDO0 | Hardware Count Down Condition 0 | Condition: When the TMR6_<t>_PWMA port is low, the rising edge detected on the TMR6_<t>_PWMB port <ul style="list-style-type: none"> <li>● 0b0: Hardware count down is disabled when the condition is met.</li> <li>● 0b1: Hardware count down is enabled when the condition is met.</li> </ul> | RW     |

**Note**

In hardware count mode, the GCONR.CKDIV must be set to 0.

## 21.5.25 PTO Pulse Number Register (TMR6\_PTNUMR)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0x0200 | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | PTNUM        |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access | RW           |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit  | Name  | Function     | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | Access |
|------|-------|--------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:0 | PTNUM | Pulse Number | <p>The number of pulses that need to be sent during one algorithm period.</p> <div>  <b>Note</b><br/> PTNUM is a signed number, PTNUM[31] indicates the direction of rotation: <ul style="list-style-type: none"> <li>● 0b0: AB phase outputs follow the sequence Bit0 -&gt; Bit1 -&gt; Bit2 -&gt; Bit3 -&gt; Bit0 -&gt; ...</li> <li>● 0b1: AB phase outputs follow the sequence Bit3 -&gt; Bit2 -&gt; Bit1 -&gt; Bit0 -&gt; Bit3 -&gt; ...</li> </ul> </div> | RW     |

## 21.5.26 PTO Pulse Period Register (TMR6\_PTPERR)

| Offset | Bit Position |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0x0204 | 15           | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x0000       |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | PTPER        |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access | RW           |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit  | Name  | Function     | Description                                                   | Access |
|------|-------|--------------|---------------------------------------------------------------|--------|
| 15:0 | PTPER | pulse period | Pulse period: $PTPER = \text{algorithm period} / (PTNUM + 1)$ | RW     |



21.5.27 PTO Algorithm Period Remainder Register (TMR6\_PTRMDR)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0x0208 | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | PTRMD        |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access | RW           |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit  | Name  | Function                   | Description                                                                  | Access |
|------|-------|----------------------------|------------------------------------------------------------------------------|--------|
| 31:0 | PTRMD | Algorithm Period Remainder | Algorithm period remainder: $PTPER = \text{Algorithm period} \% (PTNUM + 1)$ | RW     |

## 21.5.28 PTO Mode Control Register (TMR6\_PTCTLR)

| Offset | Bit Position |    |    |    |    |    |   |       |          |   |   |   |         |         |         |        |
|--------|--------------|----|----|----|----|----|---|-------|----------|---|---|---|---------|---------|---------|--------|
| 0x020C | 15           | 14 | 13 | 12 | 11 | 10 | 9 | 8     | 7        | 6 | 5 | 4 | 3       | 2       | 1       | 0      |
| Reset  | 0x0000       |    |    |    |    |    |   |       |          |   |   |   |         |         |         |        |
| Name   | Reserved     |    |    |    |    |    |   | WORKF | Reserved |   |   |   | ZOUTENA | BOUTENA | AOUTENA | PTOENA |
| Access |              |    |    |    |    |    |   | RW    |          |   |   |   | RW      | RW      | RW      | RW     |

| Bit  | Name     | Function              | Description                                                                                                                                                                     | Access |
|------|----------|-----------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 15:9 | Reserved | Reserved              | -                                                                                                                                                                               | -      |
| 8    | WORKF    | Work flag             | <ul style="list-style-type: none"> <li>0b0: PTO is stopped</li> <li>0b1: PTO is operating</li> </ul>                                                                            | RW     |
| 7:4  | Reserved | Reserved              | -                                                                                                                                                                               | -      |
| 3    | ZOUTENA  | Z-phase output enable | <ul style="list-style-type: none"> <li>0b0: Disable</li> <li>0b1: Enable</li> </ul>                                                                                             | RW     |
| 2    | BOUTENA  | B-phase output enable | <ul style="list-style-type: none"> <li>0b0: Disable</li> <li>0b1: Enable</li> </ul>                                                                                             | RW     |
| 1    | AOUTENA  | A-phase output enable | <ul style="list-style-type: none"> <li>0b0: Disable</li> <li>0b1: Enable</li> </ul>                                                                                             | RW     |
| 0    | PTOENA   | PTO mode enable       | <ul style="list-style-type: none"> <li>0b0: PTO mode is disabled, the default is PWM output mode</li> <li>0b1: PTO mode is enabled, the port outputs ABZ pulse train</li> </ul> | RW     |

## 21.5.29 PTO A-phase Output Position Register (TMR6\_PTAOPR)

| Offset | Bit Position |    |    |    |    |    |   |   |   |   |   |   |       |   |   |   |
|--------|--------------|----|----|----|----|----|---|---|---|---|---|---|-------|---|---|---|
| 0x0220 | 15           | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3     | 2 | 1 | 0 |
| Reset  | 0x0000       |    |    |    |    |    |   |   |   |   |   |   |       |   |   |   |
| Name   | Reserved     |    |    |    |    |    |   |   |   |   |   |   | PTAOP |   |   |   |
| Access |              |    |    |    |    |    |   |   |   |   |   |   | RW    |   |   |   |

| Bit  | Name     | Function                | Description                                                  | Access |
|------|----------|-------------------------|--------------------------------------------------------------|--------|
| 15:4 | Reserved | Reserved                | -                                                            | -      |
| 3:0  | PTAOP    | Phase A Output Position | Each bit represents the phase position output high/low level | RW     |

### 21.5.30 PTO B-phase Output Position Register (TMR6\_PTBOPR)

| Offset | Bit Position |    |    |    |    |    |   |   |   |   |   |   |       |   |   |   |
|--------|--------------|----|----|----|----|----|---|---|---|---|---|---|-------|---|---|---|
| 0x0228 | 15           | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3     | 2 | 1 | 0 |
| Reset  | 0x0000       |    |    |    |    |    |   |   |   |   |   |   |       |   |   |   |
| Name   | Reserved     |    |    |    |    |    |   |   |   |   |   |   | PTBOP |   |   |   |
| Access |              |    |    |    |    |    |   |   |   |   |   |   | RW    |   |   |   |

| Bit  | Name     | Function                | Description                                                  | Access |
|------|----------|-------------------------|--------------------------------------------------------------|--------|
| 15:4 | Reserved | Reserved                | -                                                            | -      |
| 3:0  | PTBOP    | Phase B Output Position | Each bit represents the phase position output high/low level | RW     |

### 21.5.31 PTO Z-phase Output Start Position Register (TMR6\_PTZOSPR)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0x0240 | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0xFFFFFFFF   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | PTZOSP       |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access | RW           |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit  | Name   | Function                      | Description                                       | Access |
|------|--------|-------------------------------|---------------------------------------------------|--------|
| 31:0 | PTZOSP | Z Phase Output Start Position | Z phase output high level starting pulse position | RW     |

### 21.5.32 PTO Z-phase Output End Position Register (TMR6\_PTZOEPR)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0x0248 | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0xFFFFFFFF   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | PTZOEP       |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access | RW           |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit  | Name   | Function                    | Description                                  | Access |
|------|--------|-----------------------------|----------------------------------------------|--------|
| 31:0 | PTZOEP | Z phase output end position | Z phase output high-level end pulse position | RW     |

## 21.5.33 PTO Z-phase Output Pulse Width Compression B Register (TMR6\_PTZCMPBR)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |          |       |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----------|-------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|--|
| 0x0264 | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16       | 15    | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |  |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |          |       |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    | ZCMPBENA | ZCMPB |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RW       |       |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |

| Bit   | Name     | Function                           | Description                                                                                                 | Access |
|-------|----------|------------------------------------|-------------------------------------------------------------------------------------------------------------|--------|
| 31:17 | Reserved | Reserved                           | -                                                                                                           | -      |
| 16    | ZCMPBENA | Z phase B compression enabled      | <ul style="list-style-type: none"> <li>0b0: Disable</li> <li>0b1: Enable</li> </ul>                         | RW     |
| 15:0  | ZCMPB    | Z phase B compression clock number | Set the number of clock of the Z output rising edge lag pulling up and falling edge pulling down in advance | RW     |

## 21.5.34 PTO Full Circle Pulse Count Register (TMR6\_PTZPERR)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0x0268 | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | PTZPER       |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access | RW           |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit  | Name   | Function                | Description                        | Access |
|------|--------|-------------------------|------------------------------------|--------|
| 31:0 | PTZPER | Full Circle Pulse Count | Pulse count per revolution setting | RW     |

## 21.5.35 PTO Pulse Initial Position Register (TMR6\_PTPSPR)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0x0280 | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | PTSP         |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access | RW           |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit  | Name  | Function         | Description               | Access |
|------|-------|------------------|---------------------------|--------|
| 31:0 | PTPSP | Initial Position | Set the initial position. | RW     |

## 21.5.36 PTO Pulse Initial Port Register (TMR6\_PTPSTR)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |       |    |    |    |          |    |    |    |         |         |         |         |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|-------|----|----|----|----------|----|----|----|---------|---------|---------|---------|---|---|---|---|---|---|
| 0x0284 | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17    | 16 | 15 | 14 | 13       | 12 | 11 | 10 | 9       | 8       | 7       | 6       | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |       |    |    |    |          |    |    |    |         |         |         |         |   |   |   |   |   |   |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    | ABSTP |    |    |    | Reserved |    |    |    | SDIRIND | ZPORTST | BPORTST | APORTST |   |   |   |   |   |   |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    | RW    |    |    |    |          |    |    |    | RW      | RW      | RW      | RW      |   |   |   |   |   |   |

| Bit   | Name     | Function                   | Description                                                                                                                                                                                                                                                          | Access |
|-------|----------|----------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:13 | Reserved | Reserved                   | -                                                                                                                                                                                                                                                                    | -      |
| 12:8  | ABSTP    | Initial Phase              | Set the AB pulse phase before startup                                                                                                                                                                                                                                | RW     |
| 7:4   | Reserved | Reserved                   | -                                                                                                                                                                                                                                                                    | -      |
| 3     | SDIRIND  | Initial Rotation Direction | <ul style="list-style-type: none"> <li>0b0: AB phase outputs follow the sequence Bit0 -&gt; Bit1 -&gt; Bit2 -&gt; Bit3 -&gt; Bit0 -&gt; ...</li> <li>0b1: AB phase outputs follow the sequence Bit3 -&gt; Bit2 -&gt; Bit1 -&gt; Bit0 -&gt; Bit3 -&gt; ...</li> </ul> | RW     |
| 2     | ZPORTST  | Z Phase Port Initial State | <ul style="list-style-type: none"> <li>0b0: Set Z port output level to low</li> <li>0b1: Set Z port output level to high</li> </ul>                                                                                                                                  | RW     |
| 1     | BPORTST  | B Phase Port Initial State | <ul style="list-style-type: none"> <li>0b0: Set B port output level to low</li> <li>0b1: Set B port output level to high</li> </ul>                                                                                                                                  | RW     |
| 0     | APORTST  | A Phase Port Initial State | <ul style="list-style-type: none"> <li>0b0: Set A port output level to low</li> <li>0b1: Set A port output level to high</li> </ul>                                                                                                                                  | RW     |

## 21.5.37 PTO Pulse Current Position Register (TMR6\_PTPCPR)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0x0290 | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | PTPCP        |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access | RO           |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit  | Name  | Function         | Description                | Access |
|------|-------|------------------|----------------------------|--------|
| 31:0 | PTPCP | Current Position | Read the current position. | RO     |

## 21.5.38 PTO Pulse Current Port Register (TMR6\_PTPCTR)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |       |    |    |    |          |    |   |   |         |         |         |         |    |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-------|----|----|----|----------|----|---|---|---------|---------|---------|---------|----|---|---|---|
| 0x0294 | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15    | 14 | 13 | 12 | 11       | 10 | 9 | 8 | 7       | 6       | 5       | 4       | 3  | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |       |    |    |    |          |    |   |   |         |         |         |         |    |   |   |   |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | ABCRP |    |    |    | Reserved |    |   |   | CDIRIND | ZPORTCR | BPORTCR | APORTCR |    |   |   |   |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RO    |    |    |    |          |    |   |   |         | RO      | RO      | RO      | RO |   |   |   |

| Bit   | Name     | Function                   | Description                                                                                                                                                                                                                                                          | Access |
|-------|----------|----------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:13 | Reserved | Reserved                   | -                                                                                                                                                                                                                                                                    | -      |
| 12:8  | ABCRP    | Current Phase              | Current AB pulse phase                                                                                                                                                                                                                                               | RO     |
| 7:4   | Reserved | Reserved                   | -                                                                                                                                                                                                                                                                    | -      |
| 3     | CDIRIND  | Current Rotation Direction | <ul style="list-style-type: none"> <li>0b0: AB phase outputs follow the sequence Bit0 -&gt; Bit1 -&gt; Bit2 -&gt; Bit3 -&gt; Bit0 -&gt; ...</li> <li>0b1: AB phase outputs follow the sequence Bit3 -&gt; Bit2 -&gt; Bit1 -&gt; Bit0 -&gt; Bit3 -&gt; ...</li> </ul> | RO     |
| 2     | ZPORTCR  | Z Phase Port Current State | <ul style="list-style-type: none"> <li>0b0: The current Z port output level is low.</li> <li>0b1: The current Z port output level is high.</li> </ul>                                                                                                                | RO     |
| 1     | BPORTCR  | B Phase Port Current State | <ul style="list-style-type: none"> <li>0b0: The current B port output level is low.</li> <li>0b1: The current B port output level is high.</li> </ul>                                                                                                                | RO     |
| 0     | APORTCR  | A Phase Port Current State | <ul style="list-style-type: none"> <li>0b0: The current A port output level is low.</li> <li>0b1: The current A port output level is high.</li> </ul>                                                                                                                | RO     |

## 21.5.39 Common Filter Control Register (TMR6\_FCNTNTR)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |          |          |          |          |          |          |          |          |          |          |          |   |    |    |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|---|----|----|---|
| 0x00EC | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14       | 13       | 12       | 11       | 10       | 9        | 8        | 7        | 6        | 5        | 4        | 3 | 2  | 1  | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |          |          |          |          |          |          |          |          |          |          |          |   |    |    |   |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | NOFICKTD | NOFIENTD | Reserved | NOFICKTC | NOFIENTC | Reserved | NOFICKTB | NOFIENTB | Reserved | NOFICKTA | NOFIENTA |   |    |    |   |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RW       | RW       |          | RW       | RW       |          | RW       | RW       |          | RW       | RW       |   | RW | RW |   |

| Bit   | Name     | Function                                     | Description                                                                                                                                                             | Access |
|-------|----------|----------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:15 | Reserved | Reserved                                     | -                                                                                                                                                                       | -      |
| 14:13 | NOFICKTD | Filter sampling reference clock selection TD | <ul style="list-style-type: none"> <li>0b00: PCLK0</li> <li>0b01: PCLK0/4</li> <li>0b10: PCLK0/16</li> <li>0b11: PCLK0/64</li> </ul>                                    | RW     |
| 12    | NOFIENTD | Capture input port filter TD                 | <ul style="list-style-type: none"> <li>0b0: TMR6_TRIGD port input filter function is disabled</li> <li>0b1: TMR6_TRIGD port input filter function is enabled</li> </ul> | RW     |
| 11    | Reserved | Reserved                                     | -                                                                                                                                                                       | -      |
| 10:9  | NOFICKTC | Filter sampling reference clock selection TC | <ul style="list-style-type: none"> <li>0b00: PCLK0</li> <li>0b01: PCLK0/4</li> <li>0b10: PCLK0/16</li> <li>0b11: PCLK0/64</li> </ul>                                    | RW     |
| 8     | NOFIENTC | Capture input port filter TC                 | <ul style="list-style-type: none"> <li>0b0: TMR6_TRIGC port input filter function is disabled</li> <li>0b1: TMR6_TRIGC port input filter function is enabled</li> </ul> | RW     |
| 7     | Reserved | Reserved                                     | -                                                                                                                                                                       | -      |
| 6:5   | NOFICKTB | Filter sampling reference clock selection TB | <ul style="list-style-type: none"> <li>0b00: PCLK0</li> <li>0b01: PCLK0/4</li> <li>0b10: PCLK0/16</li> <li>0b11: PCLK0/64</li> </ul>                                    | RW     |
| 4     | NOFIENTB | Capture input port filter TB                 | <ul style="list-style-type: none"> <li>0b0: TMR6_TRIGB port input filter function is disabled</li> <li>0b1: TMR6_TRIGB port input filter function is enabled</li> </ul> | RW     |
| 3     | Reserved | Reserved                                     | -                                                                                                                                                                       | -      |

| Bit | Name     | Function                                     | Description                                                                                                                                                              | Access |
|-----|----------|----------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 2:1 | NOFICKTA | Filter sampling reference clock selection TA | <ul style="list-style-type: none"><li>● 0b00: PCLK0</li><li>● 0b01: PCLK0/4</li><li>● 0b10: PCLK0/16</li><li>● 0b11: PCLK0/64</li></ul>                                  | RW     |
| 0   | NOFIENTA | Capture input port filter TA                 | <ul style="list-style-type: none"><li>● 0b0: TMR6_TRIGA port input filter function is disabled</li><li>● 0b1: TMR6_TRIGA port input filter function is enabled</li></ul> | RW     |



## 21.5.40 Common Software Synchronous Start Control Register (TMR6\_SSTAR)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |    |       |    |       |    |       |    |       |    |       |    |       |    |       |    |       |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|----|-------|----|-------|----|-------|----|-------|----|-------|----|-------|----|-------|----|-------|
| 0x00F0 | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7  | 6     | 5  | 4     | 3  | 2     | 1  | 0     |    |       |    |       |    |       |    |       |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |    |       |    |       |    |       |    |       |    |       |    |       |    |       |    |       |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |    |       |    |       |    |       |    |       |    |       |    |       |    |       |    |       |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   | RW | SSTA8 | RW | SSTA7 | RW | SSTA6 | RW | SSTA5 | RW | SSTA4 | RW | SSTA3 | RW | SSTA2 | RW | SSTA1 |

| Bit  | Name     | Function              | Description                                                                                                                                                                            | Access |
|------|----------|-----------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:8 | Reserved | Reserved              | -                                                                                                                                                                                      | -      |
| 7    | SSTA8    | Unit 8 software start | Write 1 to start timer unit 8, write 0 has no effect<br>Read:<br><ul style="list-style-type: none"> <li>0b0: Timer unit 8 is not started</li> <li>0b1: Timer unit 8 started</li> </ul> | RW     |
| 6    | SSTA7    | Unit 7 software start | Write 1 to start timer unit 7, write 0 has no effect<br>Read:<br><ul style="list-style-type: none"> <li>0b0: Timer unit 7 is not started</li> <li>0b1: Timer unit 7 started</li> </ul> | RW     |
| 5    | SSTA6    | Unit 6 software start | Write 1 to start timer unit 6, write 0 has no effect<br>Read:<br><ul style="list-style-type: none"> <li>0b0: Timer unit 6 is not started</li> <li>0b1: Timer unit 6 started</li> </ul> | RW     |
| 4    | SSTA5    | Unit 5 software start | Write 1 to start timer unit 5, write 0 has no effect<br>Read:<br><ul style="list-style-type: none"> <li>0b0: Timer unit 5 is not started</li> <li>0b1: Timer unit 5 started</li> </ul> | RW     |
| 3    | SSTA4    | Unit 4 software start | Write 1 to start timer unit 4, write 0 has no effect<br>Read:<br><ul style="list-style-type: none"> <li>0b0: Timer unit 4 is not started</li> <li>0b1: Timer unit 4 started</li> </ul> | RW     |
| 2    | SSTA3    | Unit 3 software start | Write 1 to start timer unit 3, write 0 has no effect<br>Read:<br><ul style="list-style-type: none"> <li>0b0: Timer unit 3 is not started</li> <li>0b1: Timer unit 3 started</li> </ul> | RW     |
| 1    | SSTA2    | Unit 2 software start | Write 1 to start timer unit 2, write 0 has no effect<br>Read:<br><ul style="list-style-type: none"> <li>0b0: Timer unit 2 is not started</li> <li>0b1: Timer unit 2 started</li> </ul> | RW     |

| Bit | Name  | Function              | Description                                                                                                                                                                          | Access |
|-----|-------|-----------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 0   | SSTA1 | Unit 1 software start | Write 1 to start timer unit 1, write 0 has no effect<br>Read: <ul style="list-style-type: none"><li>● 0b0: Timer unit 1 is not started</li><li>● 0b1: Timer unit 1 started</li></ul> | RW     |

## 21.5.41 Common Software Synchronous Stop Control Register (TMR6\_SSTPR)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |       |   |   |   |   |   |   |   |       |  |  |  |  |  |  |  |       |  |  |  |  |  |  |  |       |  |  |  |  |  |  |  |       |  |  |  |  |  |  |  |       |  |  |  |  |  |  |  |       |  |  |  |  |  |  |  |       |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |      |  |  |  |  |  |  |  |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|-------|---|---|---|---|---|---|---|-------|--|--|--|--|--|--|--|-------|--|--|--|--|--|--|--|-------|--|--|--|--|--|--|--|-------|--|--|--|--|--|--|--|-------|--|--|--|--|--|--|--|-------|--|--|--|--|--|--|--|-------|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|----|--|--|--|--|--|--|--|------|--|--|--|--|--|--|--|
| 0x00F4 | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7     | 6 | 5 | 4 | 3 | 2 | 1 | 0 |       |  |  |  |  |  |  |  |       |  |  |  |  |  |  |  |       |  |  |  |  |  |  |  |       |  |  |  |  |  |  |  |       |  |  |  |  |  |  |  |       |  |  |  |  |  |  |  |       |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |      |  |  |  |  |  |  |  |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |       |   |   |   |   |   |   |   |       |  |  |  |  |  |  |  |       |  |  |  |  |  |  |  |       |  |  |  |  |  |  |  |       |  |  |  |  |  |  |  |       |  |  |  |  |  |  |  |       |  |  |  |  |  |  |  |       |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |      |  |  |  |  |  |  |  |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   | SSTP8 |   |   |   |   |   |   |   | SSTP7 |  |  |  |  |  |  |  | SSTP6 |  |  |  |  |  |  |  | SSTP5 |  |  |  |  |  |  |  | SSTP4 |  |  |  |  |  |  |  | SSTP3 |  |  |  |  |  |  |  | SSTP2 |  |  |  |  |  |  |  | SSTP1 |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |      |  |  |  |  |  |  |  |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   | WO    |   |   |   |   |   |   |   | WO    |  |  |  |  |  |  |  | WO    |  |  |  |  |  |  |  | WO    |  |  |  |  |  |  |  | WO    |  |  |  |  |  |  |  | WO    |  |  |  |  |  |  |  | WO    |  |  |  |  |  |  |  | WO    |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO |  |  |  |  |  |  |  | WO</ |  |  |  |  |  |  |  |

| Bit  | Name     | Function             | Description                                         | Access |
|------|----------|----------------------|-----------------------------------------------------|--------|
| 31:8 | Reserved | Reserved             | -                                                   | -      |
| 7    | SSTP8    | Unit 8 Software Stop | Write 1 to stop timer unit 8, Write 0 has no effect | WO     |
| 6    | SSTP7    | Unit 7 Software Stop | Write 1 to stop timer unit 7, Write 0 has no effect | WO     |
| 5    | SSTP6    | Unit 6 Software Stop | Write 1 to stop timer unit 6, Write 0 has no effect | WO     |
| 4    | SSTP5    | Unit 5 Software Stop | Write 1 to stop timer unit 5, Write 0 has no effect | WO     |
| 3    | SSTP4    | Unit 4 Software Stop | Write 1 to stop timer unit 4, Write 0 has no effect | WO     |
| 2    | SSTP3    | Unit 3 Software Stop | Write 1 to stop timer unit 3, Write 0 has no effect | WO     |
| 1    | SSTP2    | Unit 2 Software Stop | Write 1 to stop timer unit 2, Write 0 has no effect | WO     |
| 0    | SSTP1    | Unit 1 Software Stop | Write 1 to stop timer unit 1, Write 0 has no effect | WO     |

## 21.5.42 Common Software Synchronous Clear Control Register (TMR6\_SCLRR)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |       |   |       |   |       |   |       |   |       |  |       |  |       |  |       |  |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|-------|---|-------|---|-------|---|-------|---|-------|--|-------|--|-------|--|-------|--|
| 0x00F8 | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7     | 6 | 5     | 4 | 3     | 2 | 1     | 0 |       |  |       |  |       |  |       |  |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |       |   |       |   |       |   |       |   |       |  |       |  |       |  |       |  |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   | SCLE8 |   | SCLE7 |   | SCLE6 |   | SCLE5 |   | SCLE4 |  | SCLE3 |  | SCLE2 |  | SCLE1 |  |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   | WO    |   | WO    |   | WO    |   | WO    |   | WO    |  | WO    |  | WO    |  | WO    |  |

| Bit  | Name     | Function              | Description                                          | Access |
|------|----------|-----------------------|------------------------------------------------------|--------|
| 31:8 | Reserved | Reserved              | -                                                    | -      |
| 7    | SCLE8    | Unit 8 Software Clear | Write 1 to clear timer unit 8, write 0 has no effect | WO     |
| 6    | SCLE7    | Unit 7 Software Clear | Write 1 to clear timer unit 7, write 0 has no effect | WO     |
| 5    | SCLE6    | Unit 6 Software Clear | Write 1 to clear timer unit 6, write 0 has no effect | WO     |
| 4    | SCLE5    | Unit 5 Software Clear | Write 1 to clear timer unit 5, write 0 has no effect | WO     |
| 3    | SCLE4    | Unit 4 Software Clear | Write 1 to clear timer unit 4, write 0 has no effect | WO     |
| 2    | SCLE3    | Unit 3 Software Clear | Write 1 to clear timer unit 3, write 0 has no effect | WO     |
| 1    | SCLE2    | Unit 2 Software Clear | Write 1 to clear timer unit 2, write 0 has no effect | WO     |
| 0    | SCLE1    | Unit 1 Software Clear | Write 1 to clear timer unit 1, write 0 has no effect | WO     |

## 21.5.43 Common Software Synchronous Refresh Control Register (TMR6\_SUPDR)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |       |   |       |   |       |   |       |   |       |  |       |  |       |  |       |  |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|-------|---|-------|---|-------|---|-------|---|-------|--|-------|--|-------|--|-------|--|
| 0x00FC | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7     | 6 | 5     | 4 | 3     | 2 | 1     | 0 |       |  |       |  |       |  |       |  |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |       |   |       |   |       |   |       |   |       |  |       |  |       |  |       |  |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   | SUPD8 |   | SUPD7 |   | SUPD6 |   | SUPD5 |   | SUPD4 |  | SUPD3 |  | SUPD2 |  | SUPD1 |  |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   | WO    |   | WO    |   | WO    |   | WO    |   | WO    |  | WO    |  | WO    |  | WO    |  |

| Bit  | Name     | Function                | Description                                            | Access |
|------|----------|-------------------------|--------------------------------------------------------|--------|
| 31:8 | Reserved | Reserved                | -                                                      | -      |
| 7    | SUPD8    | Unit 8 Software Refresh | Write 1 to refresh timer unit 8, write 0 has no effect | WO     |
| 6    | SUPD7    | Unit 7 Software Refresh | Write 1 to refresh timer unit 7, write 0 has no effect | WO     |
| 5    | SUPD6    | Unit 6 Software Refresh | Write 1 to refresh timer unit 6, write 0 has no effect | WO     |
| 4    | SUPD5    | Unit 5 Software Refresh | Write 1 to refresh timer unit 5, write 0 has no effect | WO     |
| 3    | SUPD4    | Unit 4 Software Refresh | Write 1 to refresh timer unit 4, write 0 has no effect | WO     |
| 2    | SUPD3    | Unit 3 Software Refresh | Write 1 to refresh timer unit 3, write 0 has no effect | WO     |
| 1    | SUPD2    | Unit 2 Software Refresh | Write 1 to refresh timer unit 2, write 0 has no effect | WO     |
| 0    | SUPD1    | Unit 1 Software Refresh | Write 1 to refresh timer unit 1, write 0 has no effect | WO     |

## 21.6 Precautions

- The CPU cannot write to the CNTER register when the counter is counting, and can only write to the CNTER register when the counter stops.
- When the corresponding buffer function is enabled, the interrupt and event outputs of GCMCR-GCMFR are disabled.
- When using the double-edge complementary symmetric PWM output function, users must calculate and set the relevant references and port output states in advance to ensure the output ports are in the expected state.
- When using the PTO function, if a value of 0 is written to TMR6\_PTNUMR, TMR6\_CNTER continues counting and the PTO output remains unchanged. When a non-zero value is subsequently written to TMR6\_PTNUMR, ABZ phase output will resume.
- When using the PTO function, the timing for writing TMR6\_PTNUMR/PTPERR must align with TMR6\_CNTER equaling TMR6\_GCMAR. It is recommended that TMR6\_GCMAR be set within 20% to 80% of TMR6\_PEAR (the algorithm period).
- In specific scenarios, multiple control events may target the counter simultaneously. In such cases, the priority order of CNTER behavior is determined according to the following table.

Table 21-90 CNTER Control Priority

| Timer Control Events                |                                     | CNTER Value                         | Priority |
|-------------------------------------|-------------------------------------|-------------------------------------|----------|
| CPU write to CNTER                  |                                     | CPU write value                     | High     |
| Update event (software or hardware) |                                     | Value from UPDAR                    | ↓        |
| Clear event (software or hardware)  | In hardware count or saw-tooth mode | 0 or PERAR based on count direction | ↓        |
|                                     | In triangular wave mode             | 0                                   | ↓        |
| Hardware counting                   |                                     | Normal counting value               | ↓        |
| Triangular wave mode counting       |                                     | Normal counting value               | ↓        |
| Sawtooth mode counting              |                                     | Normal counting value               | Low      |

- In specific scenarios, control events targeting the PWM port outputs may occur simultaneously. In such cases, the output change priority for the two ports (TMR6\_<t>\_PWMA and TMR6\_<t>\_PWMB) is defined separately as follows:

Table 21-91 PWMA Port Output Control Priority

| PWMA Port Output Control Events | Port State                   | Priority |
|---------------------------------|------------------------------|----------|
| EMB-triggered event occurs      | State defined by PCNAR.EMBCA | High     |
| Forced port output enabled      | State defined by PCNAR.FORCA | ↓        |
| At the start of counting        | State defined by PCNAR.STACA | ↓        |

| PWMA Port Output Control Events |             | Port State                    | Priority |
|---------------------------------|-------------|-------------------------------|----------|
| In sawtooth up-count            | CNTER=PERAR | State defined by PCNAR.OVFCA  | ↓        |
|                                 | CNTER=GCMAR | State defined by PCNAR.CMAUCA | ↓        |
|                                 | CNTER=GCMBR | State defined by PCNAR.CMBUCA | ↓        |
| In sawtooth down-count          | CNTER=0     | State defined by PCNAR.UDFCA  | ↓        |
|                                 | CNTER=GCMAR | State defined by PCNAR.CMADCA | ↓        |
|                                 | CNTER=GCMBR | State defined by PCNAR.CMBDCA | ↓        |
| In triangular up-count          | CNTER=GCMAR | State defined by PCNAR.CMAUCA | ↓        |
|                                 | CNTER=GCMBR | State defined by PCNAR.CMBUCA | ↓        |
|                                 | CNTER=PERAR | State defined by PCNAR.OVFCA  | ↓        |
| In triangular down-count        | CNTER=GCMAR | State defined by PCNAR.CMADCA | ↓        |
|                                 | CNTER=GCMBR | State defined by PCNAR.CMBDCA | ↓        |
|                                 | CNTER=0     | State defined by PCNAR.UDFCA  | ↓        |
| When counting stops             |             | State defined by PCNAR.STPCA  | Low      |

Table 21-92 PWMB Port Output Control Priority

| PWMB Port Output Control Events |             | Port State                    | Priority |
|---------------------------------|-------------|-------------------------------|----------|
| EMB-triggered event occurs      |             | State defined by PCNBR.EMBCEB | High     |
| Forced port output enabled      |             | State defined by PCNBR.FORCEB | ↓        |
| At the start of counting        |             | State defined by PCNBR.STACB  | ↓        |
| In sawtooth up-count            | CNTER=PERAR | State defined by PCNBR.OVFCEB | ↓        |
|                                 | CNTER=GCMBR | State defined by PCNBR.CMBUCB | ↓        |
|                                 | CNTER=GCMAR | State defined by PCNBR.CMAUCB | ↓        |

| PWMB Port Output Control Events |             | Port State                    | Priority |
|---------------------------------|-------------|-------------------------------|----------|
| In sawtooth down-count          | CNTER=0     | State defined by PCNBR.UDFCB  | ↓        |
|                                 | CNTER=GCMBR | State defined by PCNBR.CMBDCB | ↓        |
|                                 | CNTER=GCMAR | State defined by PCNBR.CMADCB | ↓        |
| In triangular up-count          | CNTER=GCMBR | State defined by PCNBR.CMBUCB | ↓        |
|                                 | CNTER=GCMAR | State defined by PCNBR.CMAUCB | ↓        |
|                                 | CNTER=PERAR | State defined by PCNBR.OVFCB  | ↓        |
| In triangular down-count        | CNTER=GCMBR | State defined by PCNBR.CMBDCB | ↓        |
|                                 | CNTER=GCMAR | State defined by PCNBR.CMADCB | ↓        |
|                                 | CNTER=0     | State defined by PCNBR.UDFCB  | ↓        |
| When counting stops             |             | State defined by PCNBR.STPCB  | Low      |



## 22 General Control Timer (Timer4)

### 22.1 Introduction

The General Control Timer 4 is a timer module designed for three-phase motor control, offering a variety of solutions for different three-phase motor control applications. It supports both triangular and sawtooth waveform modes, enabling the generation of various PWM waveforms. It also features buffer functionality and supports EMB control. 3 unit of Timer4 is carried in this product family.

### 22.2 Features

The following table outlines the basic functions and features of Timer4.

Table 22-1 Basic Functions and Features of Timer4

|                |                                             |
|----------------|---------------------------------------------|
| Waveform mode  | Sawtooth wave, triangle wave                |
| Basic function | Up/Down-counting direction                  |
|                | Buffer function                             |
|                | General PWM output                          |
|                | General compare match event                 |
|                | Overflow/Underflow/Reload count match event |
|                | Special compare match event                 |
|                | Software-synchronized counter start         |
|                | EMB control                                 |
| Interrupt type | General compare match interrupt             |
|                | Count period match interrupt                |
|                | Reload count match interrupt                |

## 22.3 Functional Description

### 22.3.1 Timer4 Block Diagram

The figure below describes the basic architecture of the general control timer Timer4. In the diagram, the symbol "<t>" denotes the unit number, which ranges from 1-3. In the remainder of this section, "<t>" will consistently refer to the unit number and will not be further explained.

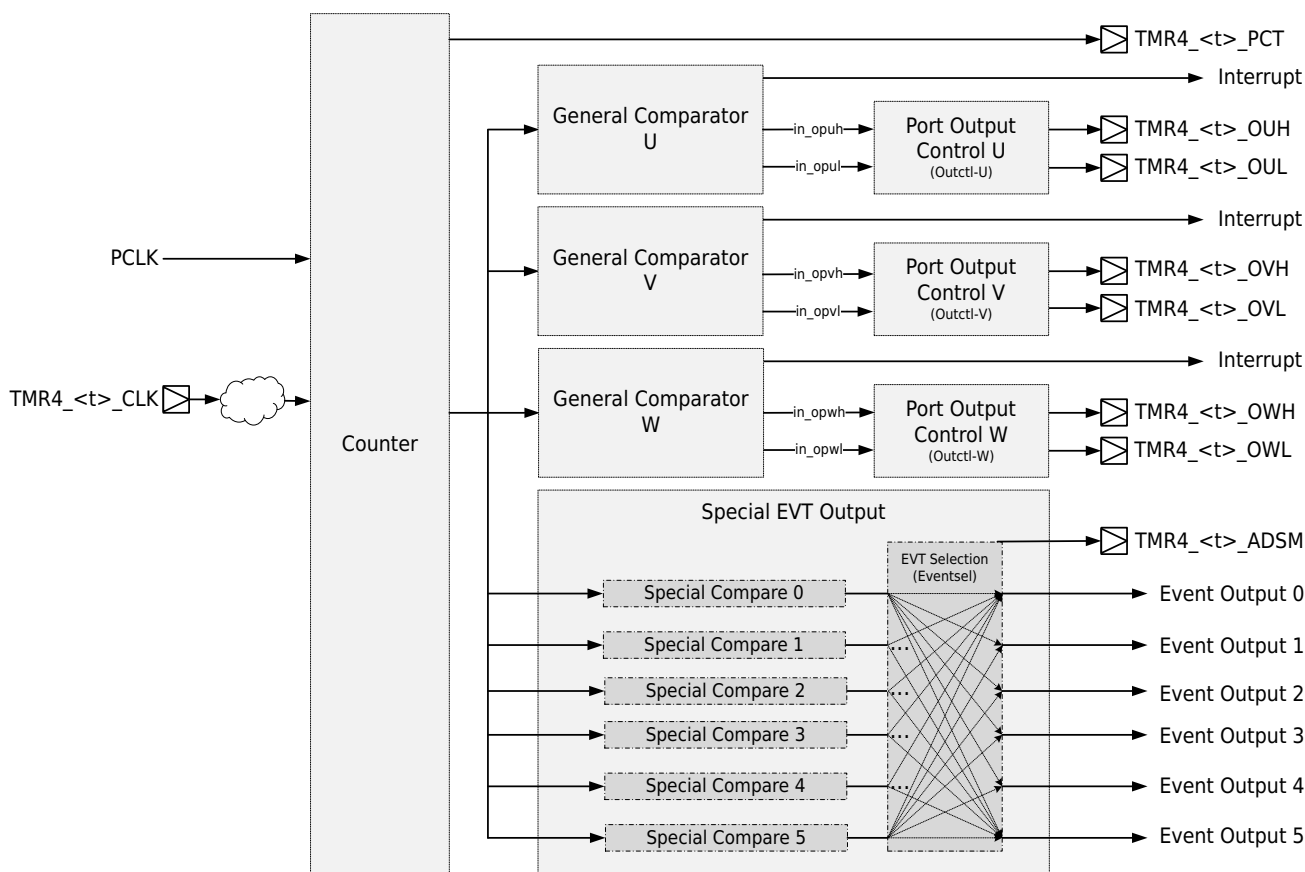


Figure 22-1 Timer4 Basic Block Diagram

The following table lists the input and output ports of Timer4.

Table 22-2 Timer4 Port List

| Port Name     | Direction | Function                        |
|---------------|-----------|---------------------------------|
| TMR4_<t>_CLK  | Input     | Count clock input port          |
| TMR4_<t>_PCT  | Output    | PWM periodic output monitoring  |
| TMR4_<t>_ADSM | Output    | Special event output monitoring |

| Port Name    | Direction | Function        |
|--------------|-----------|-----------------|
| TMR4_<t>_OUH | Output    | PWM output port |
| TMR4_<t>_OUL |           |                 |
| TMR4_<t>_OVH |           |                 |
| TMR4_<t>_OVL |           |                 |
| TMR4_<t>_OWH |           |                 |
| TMR4_<t>_OWL |           |                 |

## 22.3.2 Basic Actions

### 22.3.2.1 Waveform Mode

Timer4 has two basic counting waveform modes, sawtooth wave mode and triangle wave mode. The two waveform modes can be implemented based on the setting of CCSR.MODE.

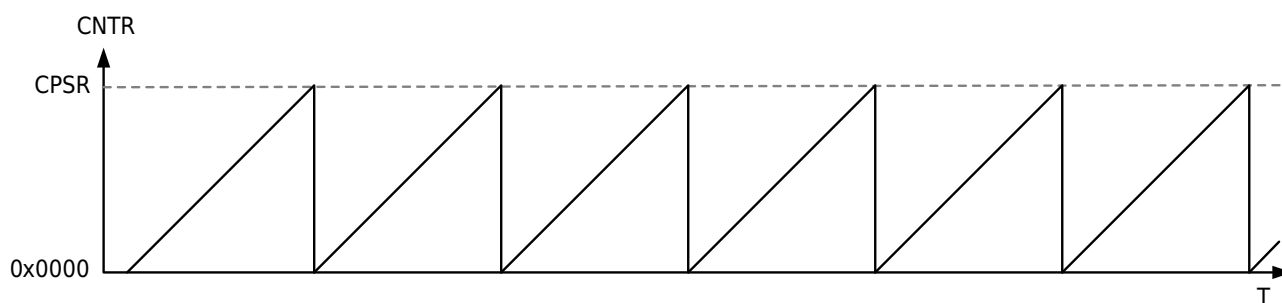


Figure 22-2 Timer4 Sawtooth Waveform

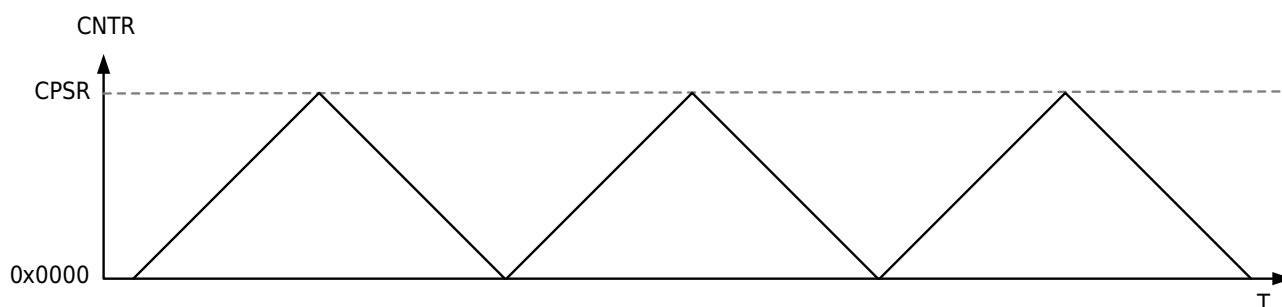


Figure 22-3 Timer4 Triangle Waveform

### 22.3.2.2 Counting Action

- The sawtooth wave counting operation and control process are shown in the figure below:

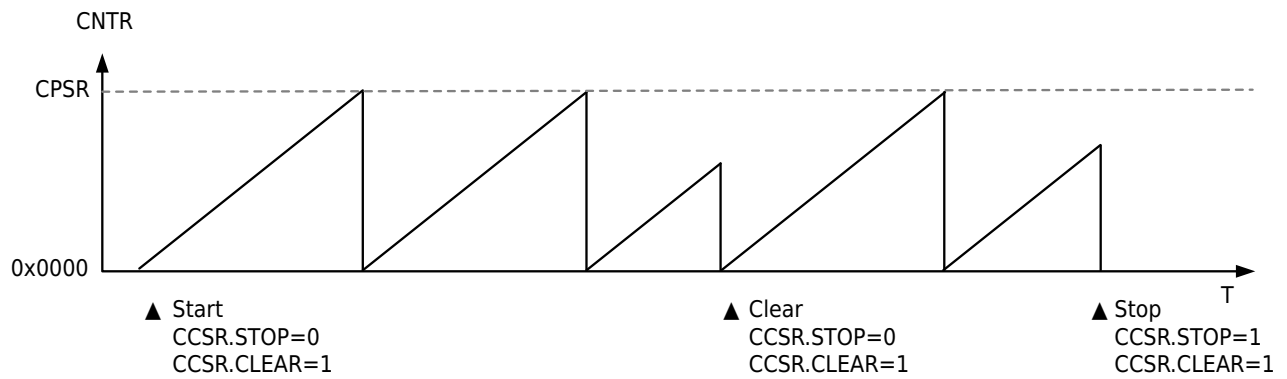


Figure 22-4 Timer4 Sawtooth Wave Mode Counting Action

1. Setting mode `CCSR.MODE=0`.
  2. Set the count peak CPSR register.
  3. By writing `CCSR.STOP=0` and `CCSR.CLEAR=1`, the counter count value (CNTR) is initialized to 0x0000 and the counting operation is started. The counter value starts counting up from 0x0000. When it reaches the peak value (CPSR), the count value returns to 0x0000, and this operation is repeated in sequence.
  4. Counting period =  $(\text{CPSR} + 1) \times \text{counting clock period}$ .
  5. During the counting process, writing `CCSR.STOP=0` and `CCSR.CLEAR=1` can initialize the counting value to 0x0000 and continue counting operation.
  6. During the counting process, writing `CCSR.STOP=1` and `CCSR.CLEAR=1` can initialize the count value to 0x0000 and stop the counting operation.
- The triangle wave counting operation and control process are shown in the figure below:

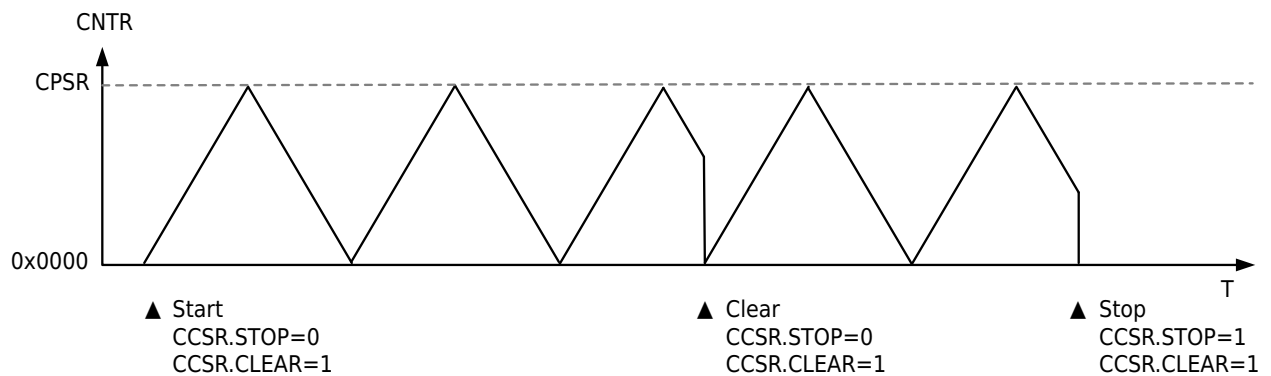


Figure 22-5 Timer4 Triangle Wave Mode Counting Action

1. Setting mode `CCSR.MODE=1`.
2. Set the count peak CPSR register.
3. By writing `CCSR.STOP=0` and `CCSR.CLEAR=1`, the counter count value (CNTR) is initialized to 0x0000 and the counting operation is started. The counter value starts counting up from 0x0000 and counts up before reaching the counting peak value; when the peak value (CPSR) is reached, the counter starts counting down until the count value returns to 0x0000; then the counting operation is repeated again, and this operation is repeated in sequence.
4. Counting period =  $\text{CPSR} \times 2 \times \text{counting clock period}$ .
5. During the counting process, writing `CCSR.STOP=0` and `CCSR.CLEAR=1` can initialize the count value to 0x0000 and restart the increment counting operation, and then repeat the above operation.

6. During the counting process, writing `CCSR.STOP=1` and `CCSR.CLEAR=1` can initialize the count value to 0x0000 and stop the counting operation.

### 22.3.2.3 Compare Output

The compare output function consists of universal comparators U, V, and W. The general comparator includes the compare reference register (`TMR4_OCCR`), the general control status register (`TMR4_OCSR`), the general extension control register (`TMR4_OCER`), and the general mode control register (`TMR4_OCMR`). The general comparator outputs an internal signal, which outputs a high or low level or PWM according to the configuration of the above registers.

Table 22-3 Internal Signal of General Comparator Output

| General Compa-rator   | Internal Signal | Description                                                                                                                                                                                                                                                                   |
|-----------------------|-----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| General Compa-rator U | in_opuh         | When <code>OCSRu.OCEH=0</code> , the output high or low level is specified by <code>OCSRu.OCPh</code> ;<br>When <code>OCSRu.OCEH=1</code> , the PWM output is configured by the relevant bits of registers <code>OCCRuH</code> , <code>OCMRuH</code> and <code>OCERu</code> ; |
|                       | in_opul         | When <code>OCSRu.OCEL=0</code> , the output high or low level is specified by <code>OCSRu.OCPL</code> ;<br>When <code>OCSRu.OCEL=1</code> , the PWM output is configured by the relevant bits of registers <code>OCCRuL</code> , <code>OCMRuL</code> and <code>OCERu</code> ; |
| General Compa-rator V | in_opvh         | When <code>OCSRv.OCEH=0</code> , the output high or low level is specified by <code>OCSRv.OCPh</code> ;<br>When <code>OCSRv.OCEH=1</code> , the PWM output is configured by the relevant bits of registers <code>OCCRvH</code> , <code>OCMRvH</code> and <code>OCERv</code> ; |
|                       | in_opvl         | When <code>OCSRv.OCEL=0</code> , the output high or low level is specified by <code>OCSRv.OCPL</code> ;<br>When <code>OCSRv.OCEL=1</code> , the PWM output is configured by the relevant bits of registers <code>OCCRvL</code> , <code>OCMRvL</code> and <code>OCERv</code> ; |
| General Compa-rator W | in_opwh         | When <code>OCSRw.OCEH=0</code> , the output high or low level is specified by <code>OCSRw.OCPh</code> ;<br>When <code>OCSRw.OCEH=1</code> , the PWM output is configured by the relevant bits of registers <code>OCCRwH</code> , <code>OCMRwH</code> and <code>OCERw</code> ; |
|                       | in_opwl         | When <code>OCSRw.OCEL=0</code> , the output high or low level is specified by <code>OCSRw.OCPL</code> ;<br>When <code>OCSRw.OCEL=1</code> , the PWM output is configured by the relevant bits of registers <code>OCCRwL</code> , <code>OCMRwL</code> and <code>OCERw</code> ; |

- The figure below is an example of the waveform output of the comparison output module in sawtooth counting mode.

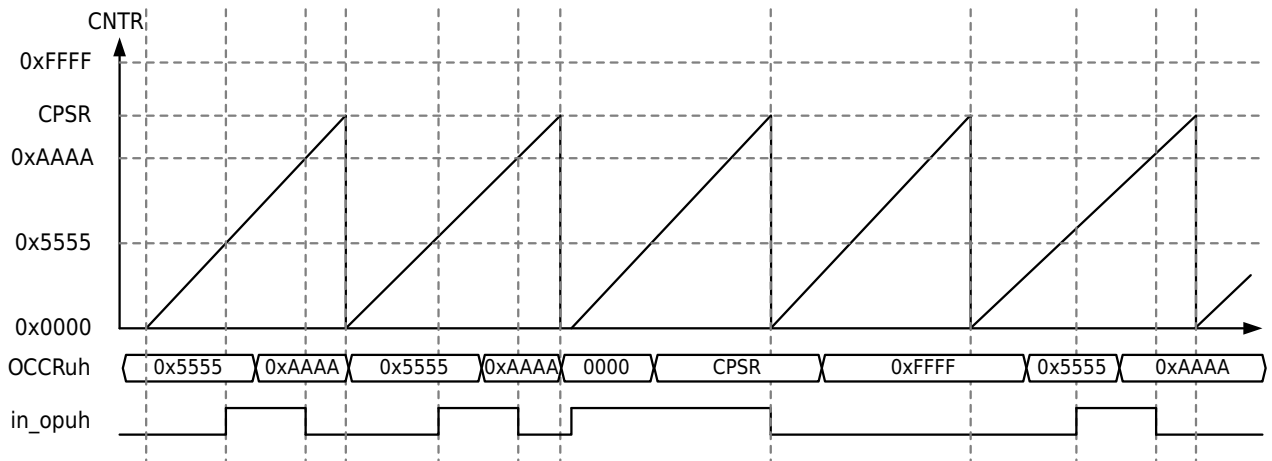


Figure 22-6 Sawtooth Mode Waveform Output Example

- The figure below is an example of the waveform output of the comparison output module in triangle wave counting mode.

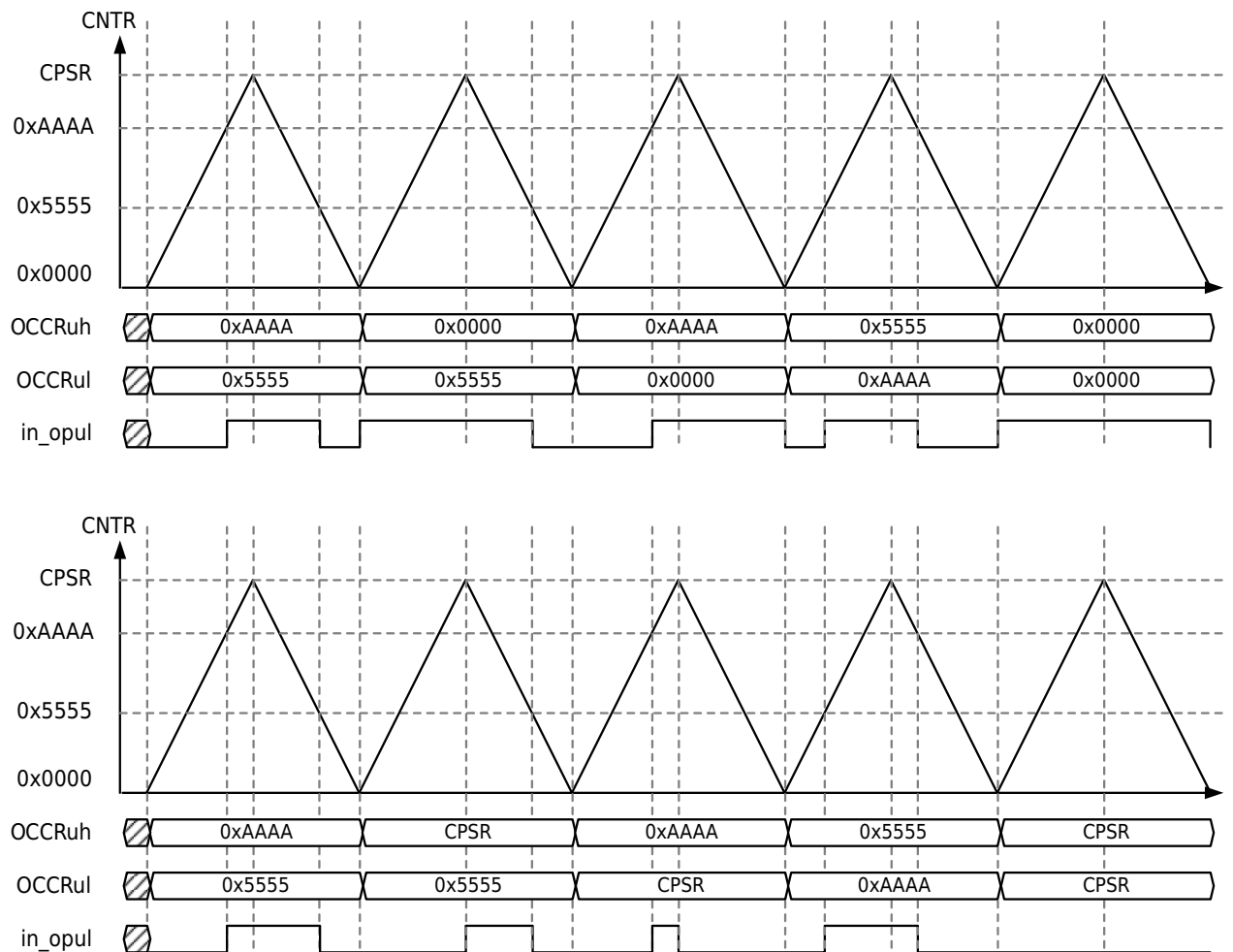


Figure 22-7 Triangular Wave Mode Waveform Output Example

### 22.3.3 Buffer Function

The Timer4 module provides buffer functionality for the CPSR Register, the OCCR Register, the OCMR Register, the SCCR Register, and the SCMR Register.

### 22.3.3.1 Count Period Set Register Buffer Function

The CPSR Register supports buffer function. When a peak count value is written, it is first stored in the buffer register. The data is transferred from the buffer register to the CPSR register under the following conditions:

1. When the buffer function is disabled (CCSR.BUFEN=0), the written data is immediately transferred from the buffer register to the CPSR register.
2. When the buffer function is enabled (CCSR.BUFEN=1), the data is transferred from the buffer register to the CPSR register either when the counter is stopped (CCSR.STOP=1) or when the counter value reaches 0x0000.



#### Notice

When reading from the CPSR, the value read is not from the buffer register but from the CPSR register itself. If the buffer function is enabled, the value read before the transfer is completed is not the most recently written value, but the previously loaded value of CPSR.

- When the sawtooth wave mode is on and the buffer function is disabled, modify the operation of the count peak CPSR:

CCSR.BUFEN=0

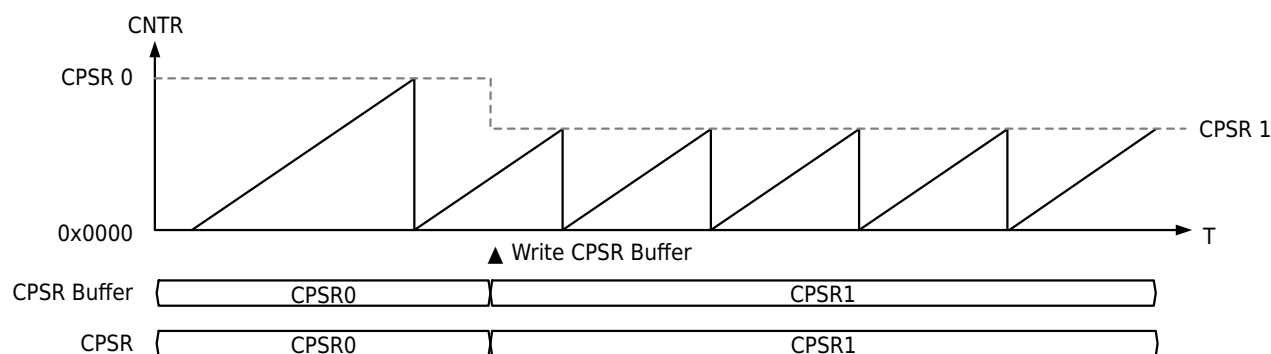


Figure 22-8 Modify Sawtooth Count Period When the Buffer is disabled



#### Notice

When the buffer function is disabled, the written data is immediately transferred from the buffer register to the CPSR register, and the counting period changes immediately after the write operation is completed. In this case, if the written value is smaller than the current counter value, the counter will continue incrementing until it reaches 0xFFFF. Therefore, special caution is required in such scenarios.

- When the sawtooth wave mode is on and the buffer function is enabled, modify the operation of the count peak CPSR:

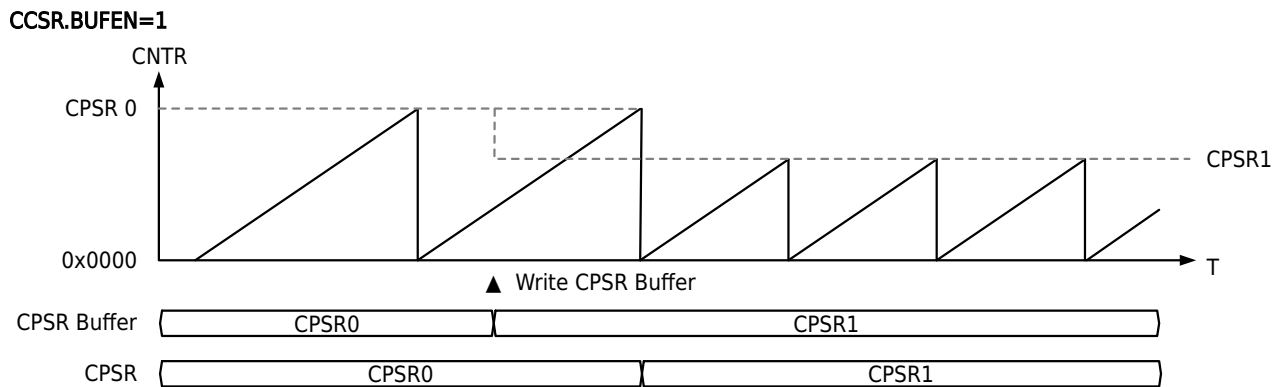


Figure 22-9 Modify Sawtooth Wave Count Period when Buffer is Enabled

When the buffer function is enabled, the written data is transferred from the buffer register to the CPSR register when the counter stops or when the counter value is "0x0000".

- When the triangle wave mode is on and the buffer function is enabled, modify the operation of the count peak CPSR:

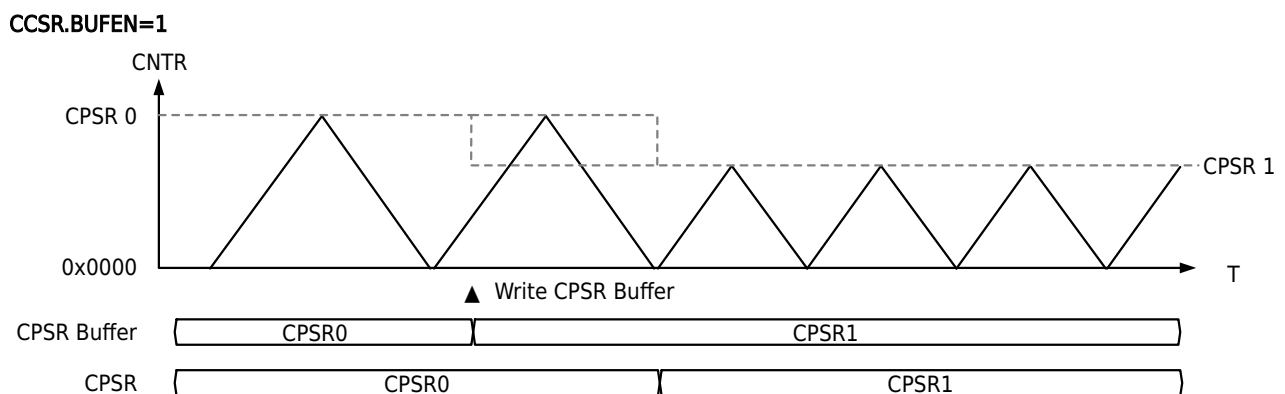


Figure 22-10 Modify Triangular Wave Count Period when Buffer is Enabled

In triangle wave counting mode, when the buffer function is enabled, the data written to the buffer register is transferred to the CPSR register when the counter stops or the next counter value is "0x0000." The change in the counter period occurs at the start of the next counter period after the write operation is completed.

### 22.3.3.2 General Compare Register Buffer Function

The general compare reference register (OCCR) and the general mode control register (OCMR) both have buffer register functions. When the buffer function is enabled, the data is loaded into the OCCR and OCMR registers at the specified transmission time. The OCCR buffer function is used to synchronize the change of the comparison value during the counting process, while the OCMR buffer function is used to synchronize the change of the internal PWM output at the counting overflow point (sawtooth wave), valley point, or peak point (triangle wave).

1. When the link transmission of output comparison with the counter period interval response function is disabled, the buffer value is loaded into the register at the set counting state. In this case, the loading process is independent of the counter period interval counter.

The figure below shows the waveform (x=L or H) when the general output compare OCCR buffer function is enabled, the counter is loaded at zero value (OCER.CxBUFEN=0b01), and the counter period interval response link is disabled (OCER.LMCx=0).



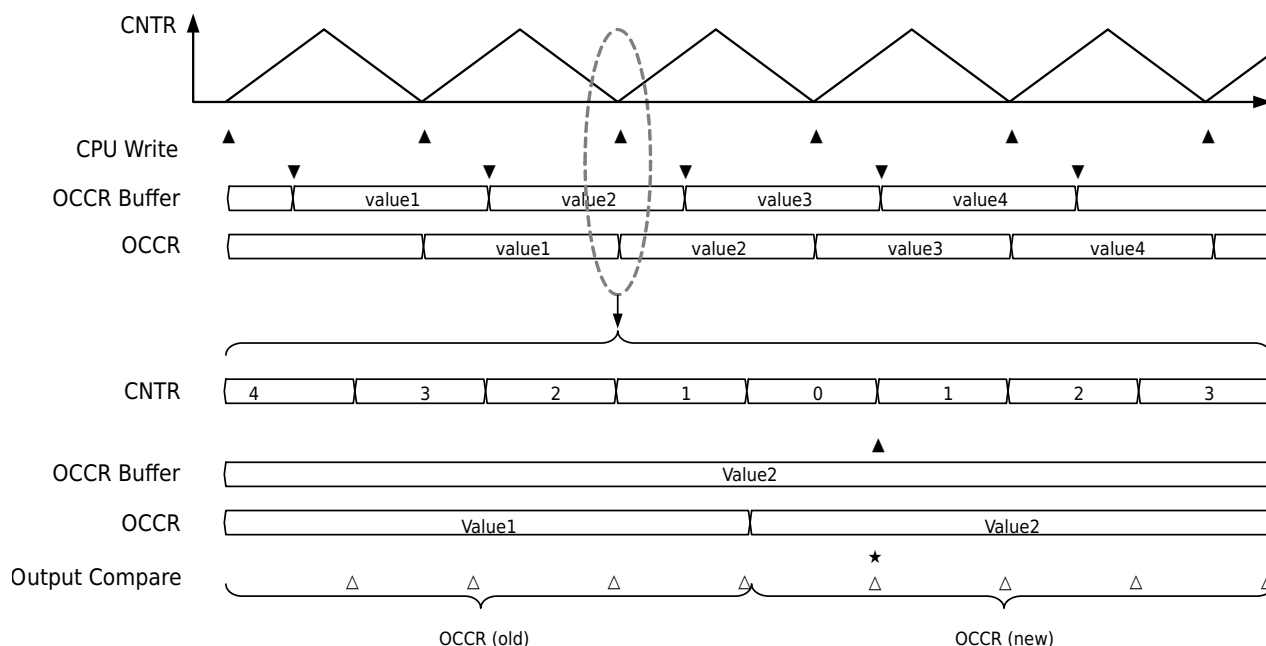


**Figure 22-11 OCCR Buffer Data Transmission (Period Interval Response Link Disabled)**

The upper part of the figure is a global schematic diagram, and the lower part is an enlarged view of the transmission operation.

The counter is in triangle wave counting mode, and the underflow interrupt is triggered at the flag ▲ (valley point). At ▼, the CPU rewrites the OCCR register and the data written is stored in the OCCR buffer register. After that, when the counter count is 0x0000, the data loading operation is performed, from the buffer register to the OCCR register, and the interrupt flag IRQZF is generated.

At Δ, the output comparison is based on the OCCR register value and counter value matching event, changing the PWM output and setting the OCSR.OCFx bit (x=L or H). After the ★ point (counter value=0x0000), the port output control operates based on the new OCCR data. Before the ★ point, the operation is based on the original OCCR data.

The figure illustrates the transmission operation of the OCCR buffer register at the counting valley point, and the buffer transfer operation of the OCMR buffer register is similar; similarly, the transmission operation at the counting peak point is also similar. New data becomes active immediately after the transmission time (newly written data will control the PWM output and set the interrupt flag).

2. **When the counter period interval response link is enabled, the buffer register transfer operation for the buffer value is executed when the counter is in the set counting state and the count value of the period interval counter is 0.**

The figure below shows the waveform (x=L or H) when the general output compare OCCR buffer function is enabled, the counter is loaded at zero value (OCER.CxBUFEN=0b01), and the counter period interval response link is enabled (OCER.LMCx=1).

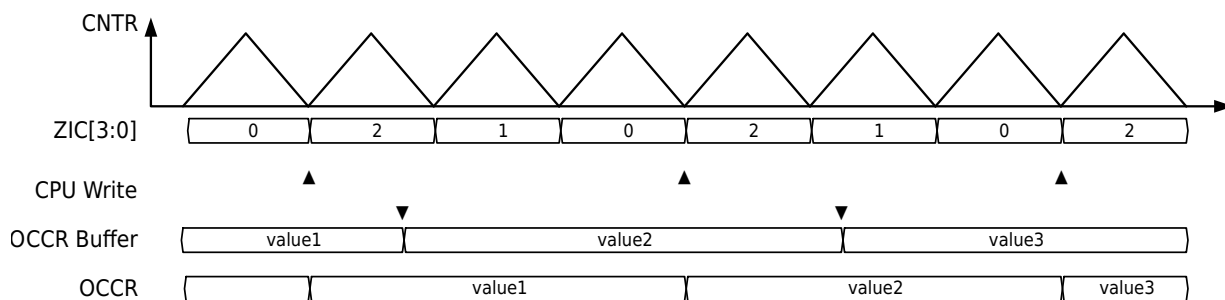


Figure 22-12 OCCR Buffer Data Transmission (Period Interval Response Link Enabled)

The counter is in triangle wave counting mode, and the period interval counter (CVPR.ZIC) counts down from 2 to 0. The underflow interrupt is triggered at the flag ▲ (valley point). At ▼, the CPU rewrites the OCCR register, and the written data is stored in the OCCR buffer register. After that, when the counter count value is 0x0000 and the period interval counter (CVPR.ZIC) is 0, the data is loaded from the buffer register to the OCCR register, and the interrupt flag IRQZF is generated.

The figure shows the OCCR buffer register transfer operation at the counting valley point, and the OCMR buffer register transfer operation is similar. Similarly, the transfer operation at the counting peak is similar. The new data takes effect immediately after the transmission time (the new write data will control the PWM output and interrupt flag).

When using the channel link operation mode, enabling both the OCCRuh and OCCRul buffer functions can generate a variety of different PWM output waveforms. As shown in the figure below, while the output compare registers OCCRuh and OCCRul remain unchanged, changing the OCMR register value will make in\_opul output different waveform signals.

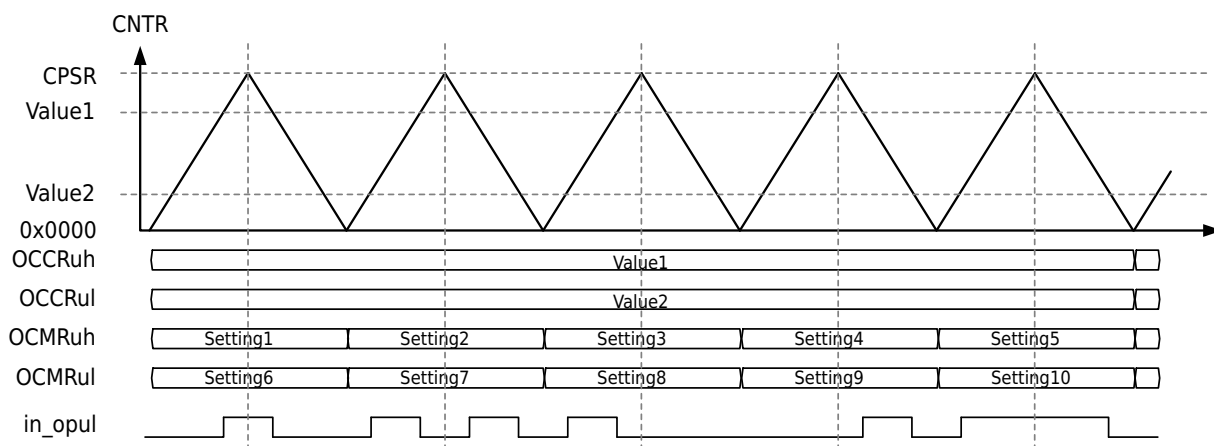


Figure 22-13 Output compare Buffer Data Transfer (OCMR Buffer Enabled)

### 22.3.3.3 Special Compare Register Buffer Function

The special compare control register (SCCR) and the special control mode register (SCMR) both have buffer function registers. When the buffer function is enabled, the values written by the CPU to the SCCR and SCMR buffer registers are loaded into the SCCR and SCMR registers at the set counter state.

1. When the counter period interval response link transmission is disabled, the buffer transmission operation is solely related to the counter state and is not affected by the counter period interval counter.

The following figure illustrates the operation when the register buffer function is enabled, the counter period interval response link transmission is disabled (SCSR.LMC=0), and the counter is at zero value (SCSR.BUFEN=0b01), transferring data to the SCCR and SCMR registers.

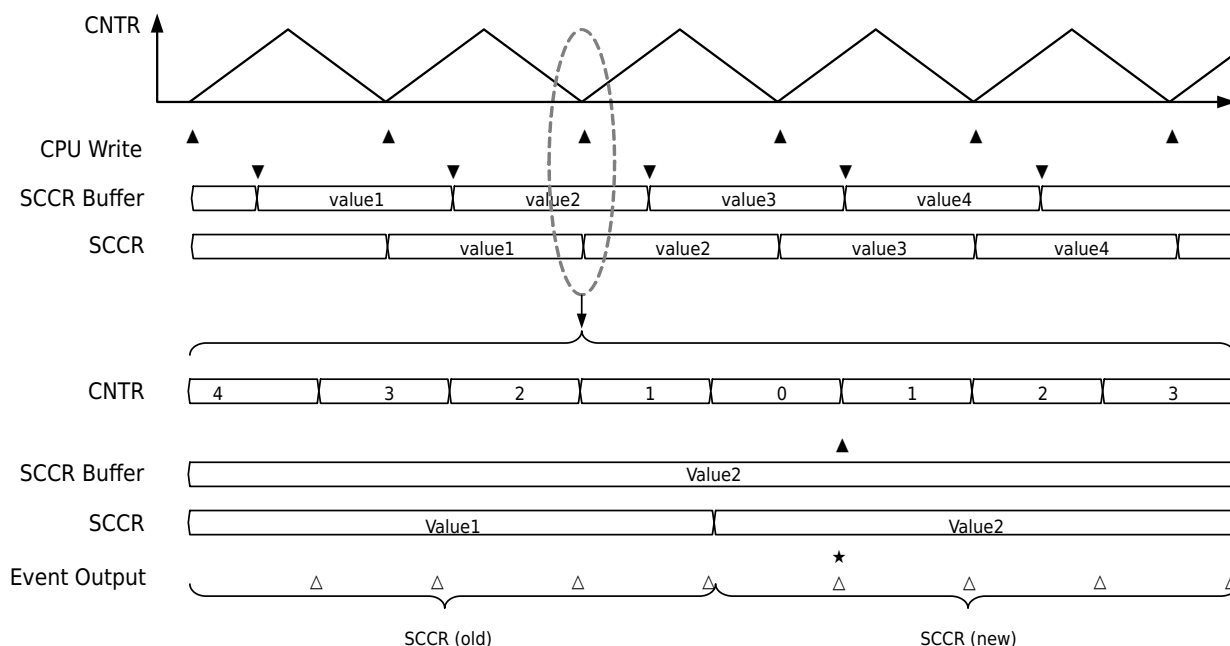


**Figure 22-14 SCCR Buffer Transmission Operation (Period Interval Response Link Transmission Disabled)**

The upper part of the figure is a global schematic diagram, and the lower part is a local enlarged diagram of the Buffer register transfer operation.

The counter is in triangle wave counting mode, and the underflow interrupt is triggered at the flag ▲ (valley point). At ▼, the CPU rewrites the SCCR register and the data written is stored in the SCCR buffer register. Afterwards, when the counter count value is 0x0000, the data loading operation is performed, and the data is transferred from the buffer register to the SCCR register, and the interrupt flag IRQZF is generated at the same time.

At time △, a comparison operation is performed based on the configured value in the SCCR register and the counter value. After time ★ (when the counter value equals 0x0000), the special event output signal operates according to the new SCCR data. Before the ★ point, the operation is based on the original SCCR data.

The figure illustrates the transmission operation of the SCCR buffer register at the counting valley point, and the buffer transfer operation of the SCMR buffer register is similar; similarly, the transmission operation at the counting peak point is also similar. The new data takes effect immediately after the transmission moment (the newly written data will set the special event output signal and interrupt flag).

2. When the counter period interval response link is enabled, the buffer register transfer operation is executed when the buffer value is in the set counting state and the count value of the period interval counter is 0.

The following figure shows the schematic diagram of enabling the SCCR buffer function, loading the counter at zero value (SCSR.BUFEN=0b01), and enabling the counter periodic interval response link transmission (SCSR.LMC=1).

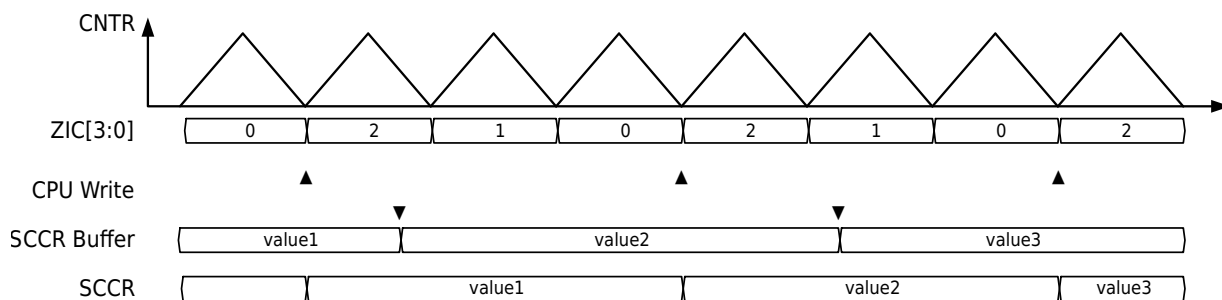


Figure 22-15 SCCR Buffer Transmission Operation (when Periodic Interval Response Link Transmission Enabled)

The counter is in triangle wave counting mode, and the period interval counter (CVPR.ZIC) counts down from 2 to 0. The underflow interrupt is triggered at the flag ▲ (valley point). At ▼, the CPU rewrites the SCCR register and the data written is stored in the SCCR buffer register. After the counter value reaches 0x0000 and the period interval counter (CVPR.ZIC) is 0, the data load operation occurs, and the data is transferred from the buffer register to the SCCR register, triggering the interrupt flag IRQZF.

## 22.3.4 General PWM Output

### 22.3.4.1 Overview

When OCCR\*h, OCCR\*l, OCMR\*h, OCMR\*l, PSCR.OE\*H, PSCR.OE\*L, PDAR\*, PDBR\*, PSCR.OS\*H, PSCR.OS\*L, and OCER\* are mentioned later in this section, the wildcard character "\*" represents u, v, w; when referring to internal output signals in\_op\*h, in\_op\*l, the wildcard character "\*" represents u, v, w; when referring to ports TMR4\_<t>\_O\*H, TMR4\_<t>\_O\*L, the wildcard character "\*" represents U, V, W.

The port output control (U, V, W) function controls the output of the internal signal of the general comparator to the external ports TMR4\_<t>\_O\*H and TMR4\_<t>\_O\*L. The port output control includes the TMR4\_POCSR register, TMR4\_PSCR register, TMR4\_PFSR register, TMR4\_PDA<B>R register, and TMR4\_RCSR register.

The corresponding relationship between the internal signal of the general comparator output and the port output control is shown in the following table.

Table 22-4 Corresponding Relationship Between Internal Signal of General Comparator Output and Port Output Control

| General Comparator   | General Comparator Internal Signals | Port Output Control   |
|----------------------|-------------------------------------|-----------------------|
| General Comparator U | in_opuh                             | Port Output Control U |
|                      | in_opul                             |                       |
| General Comparator V | in_opvh                             | Port Output Control V |
|                      | in_opvl                             |                       |
| General Comparator W | in_opwh                             | Port Output Control W |
|                      | in_opwl                             |                       |

### 22.3.4.2 Independent PWM Output

In the pass-through mode (POCR.PWMMD=0b00), the references of OCCR\*h and OCCR\*l and the port output states of OCMR\*h and OCMR\*l are set respectively to achieve different PWM outputs. At this time, the PWM output of each port is controlled independently. Before using the independent PWM output

mode, you need to set the MOE, OE\*H, OE\*L bits (PSCR register) to enable the output of Timer4. For detailed description, please refer to "[PWM Status Control Register \(TMR4\\_PSCR\)](#)". The following two figures are examples of independent PWM output under sawtooth wave and triangle wave of unit 1.

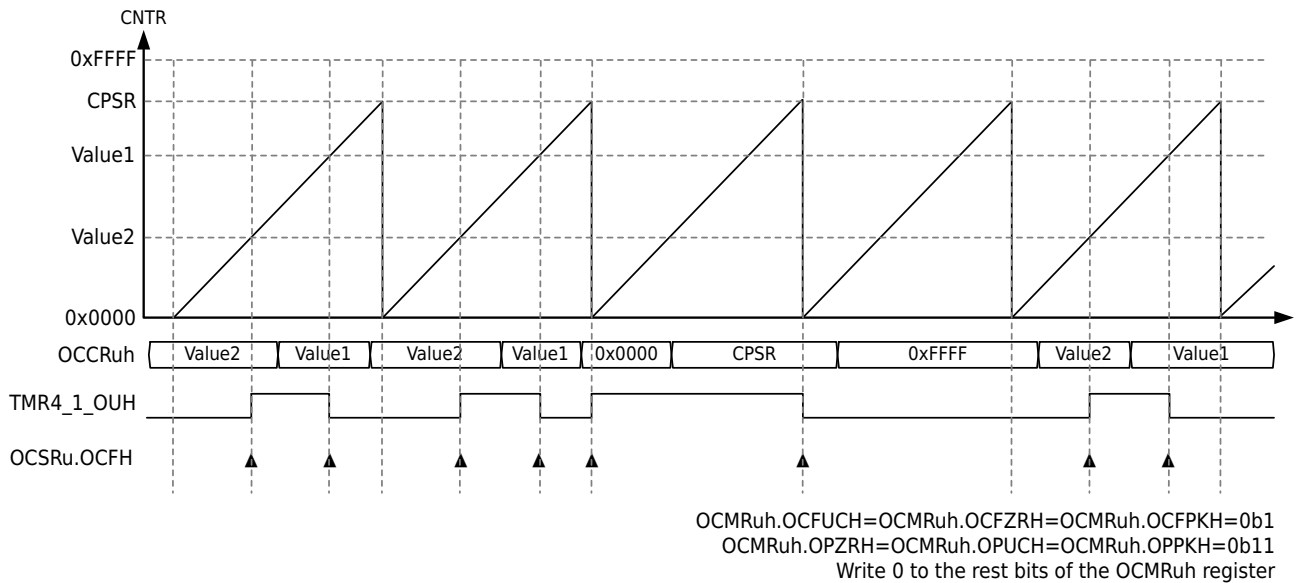


Figure 22-16 Sawtooth Independent PWM Output Example

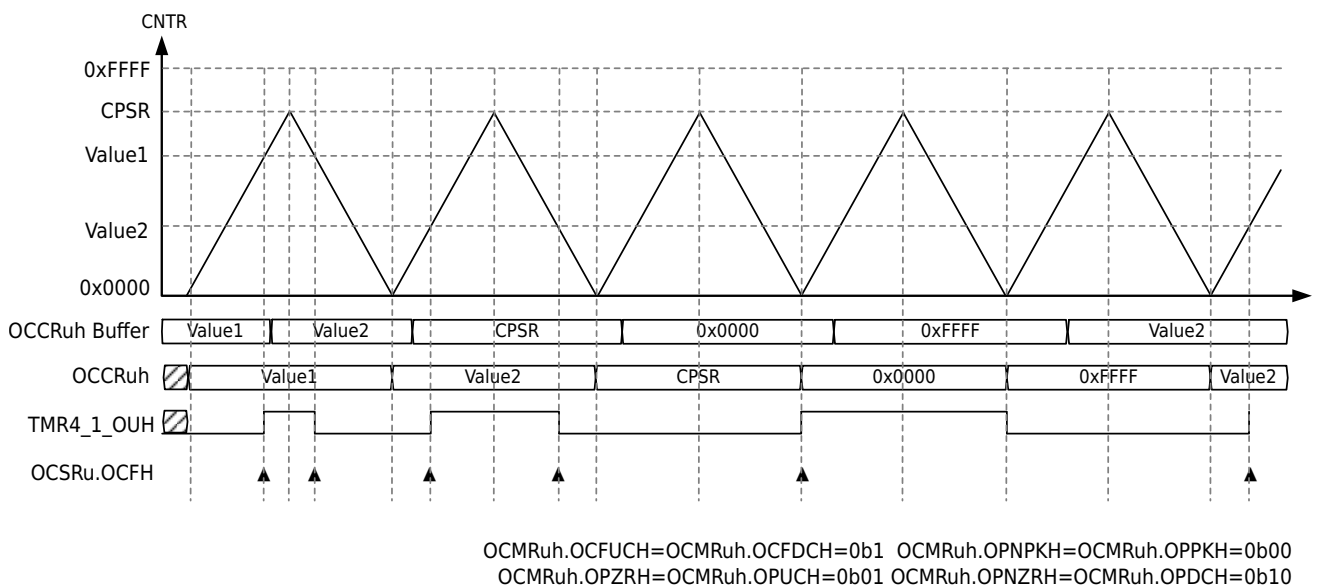


Figure 22-17 Triangle Independent PWM Output Example



#### Note

The pass-through mode means that the internal output signal (in\_op\*h, in\_op\*i) generated by the comparison match of the value of the general compare reference register (OCCR\*h, OCCR\*i) is directly output to the corresponding port (TMR4\_<t>\_O\*h, TMR4\_<t>\_O\*L).

### 22.3.4.3 Extended PWM Output

In pass-through mode (POCR.PWMMD=0b00), the port output status of TMR4\_<t>\_O\*L can also be determined by the extension bit (Bit32-16) in the OCMR\*i register. The setting of this extension bit is related to the OCCR\*h reference, thereby realizing extended PWM output on the TMR4\_<t>\_O\*L port. Before using the extended PWM output mode, you need to set the MOE, OE\*H, OE\*L bits (PSCR register) to enable the output of Timer4. For detailed description, please refer to the "[PWM Status Control Register](#)".

(TMR4\_PSCR)". The figure below shows the PWM output of TMR4\_<t>\_OUH and TMR4\_<t>\_OUL ports in this mode.

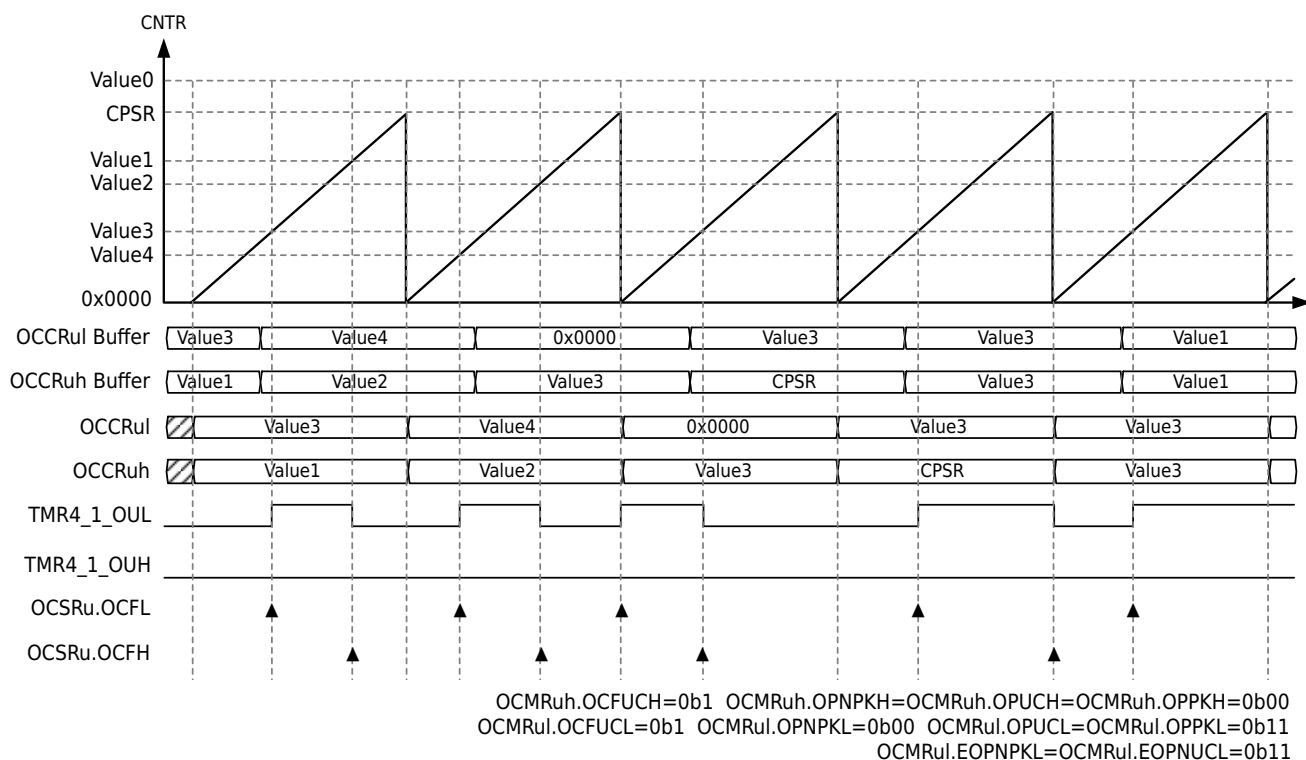


Figure 22-18 Sawtooth Wave Extended PWM Output



#### Note

In independent PWM output mode, the port status of TMR4\_<t>\_O\*L is determined by Bit15-Bit4 of the OCMRxl register and is only related to the OCCR\*I reference.

### 22.3.4.4 Complementary PWM Output

#### Software Setting for Complementary PWM Output

In the pass-through mode (POCR.PWMMD=0b00), directly set the OCCR\*h and OCCR\*l references to realize a pair of complementary PWM waveforms output to the port. The same setting of 3 groups of ports can realize 3-phase complementary PWM output.

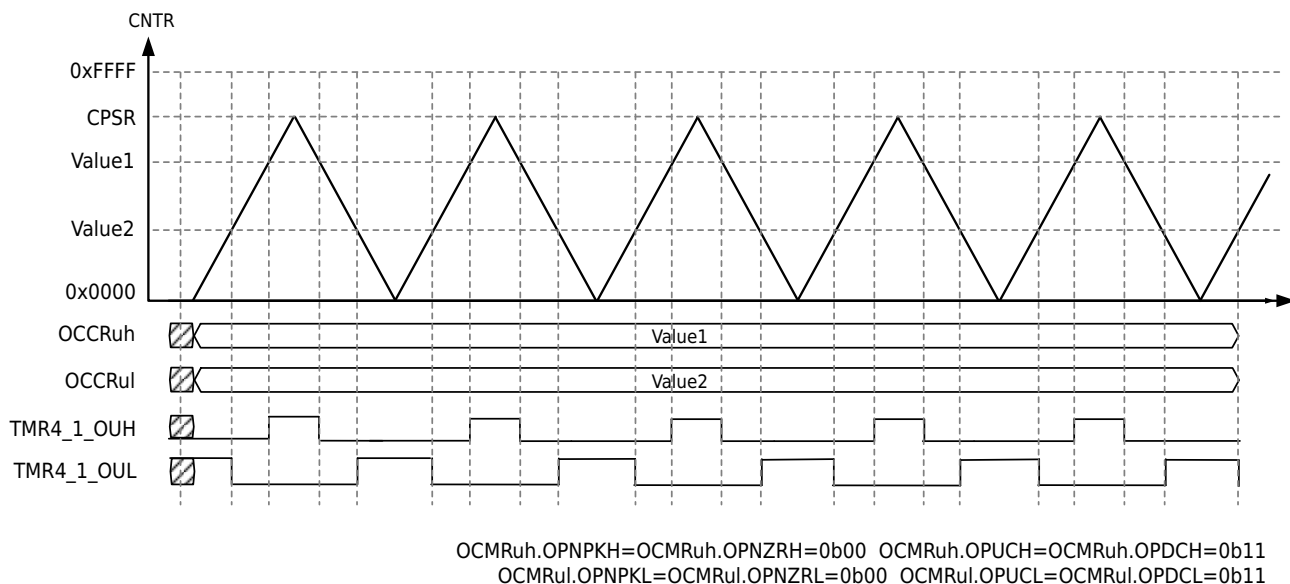


Figure 22-19 Software Implementation of Complementary PWM Output

### Hardware Implementation of Complementary PWM Output

In the dead-zone generator mode (POCR.PWMMD=0b01), the internal output signal (in\_op\*I) generated by a comparison match of the general compare control register (OCCR\*I) and the set values in the PWM dead-zone control registers (PDAR/PDBR) are used to generate complementary PWM output through hardware, with timing offsets.

In this mode, the polarity of the output of the TMR4\_<t>\_O\*H port is the same as that of in\_op\*I, and the polarity of the output of the TMR4\_<t>\_O\*L port is opposite to that of in\_op\*I. The figure below is an example of complementary PWM output in dead-zone timer mode.

If the rising edge of in\_op\*I is detected, the TMR4\_<t>\_O\*L output becomes low level, and the dead zone counter loads the set value of the PDBR\* register and starts counting down. When the count value becomes 0x0000, the counter stops and TMR4\_<t>\_O\*H outputs high level; If the falling edge of in\_op\*I is detected, the TMR4\_<t>\_O\*H output becomes low level, and the dead zone counter loads the set value of the PDAR\* register and starts counting down. When the count value becomes 0x0000, the counter stops and makes TMR4\_<t>\_O\*L output high level.

By setting the PWM dead zone control registers PDAR and PDBR, the dead zone time of output rise and fall changes can be set accordingly.

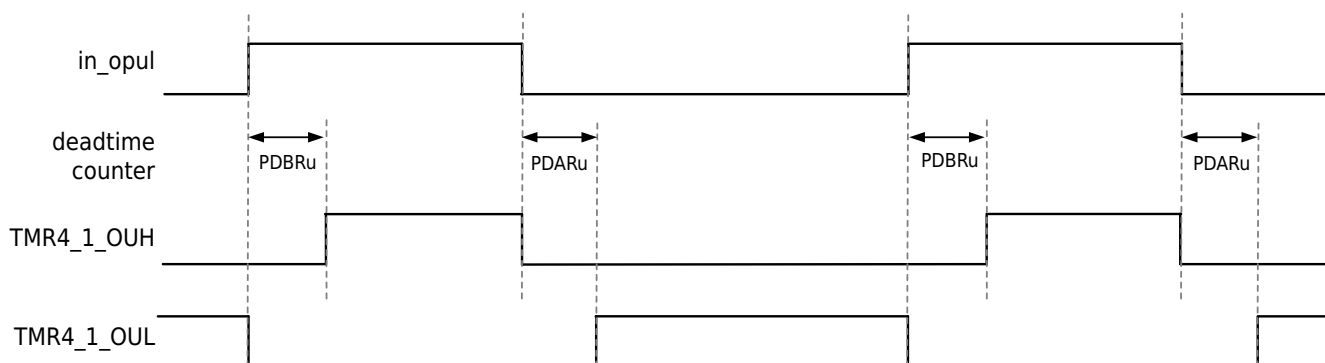


Figure 22-20 Complementary PWM Output in Dead-Zone Generator Mode

When the high-level pulse width of in\_op\*I is less than the dead time set by PDBR, only the TMR4\_<t>\_O\*L output becomes low level. The condition for the TMR4\_<t>\_O\*L output level to become

high is when the dead time set by the PDAR register has passed after the falling edge of in\_op\*I. In this case, the TMR4\_<t>\_O\*H output will continue to remain at a low level.

When the low level pulse width of in\_op\*I is less than the dead time set by PDBR, only the TMR4\_<t>\_O\*H output becomes low level. The condition for the TMR4\_<t>\_O\*H output level to become high is after the dead time set by the PDBR register after the rising edge of in\_op\*I. In this case, the TMR4\_<t>\_O\*L output will continue to remain at a low level. As shown in the figure below.

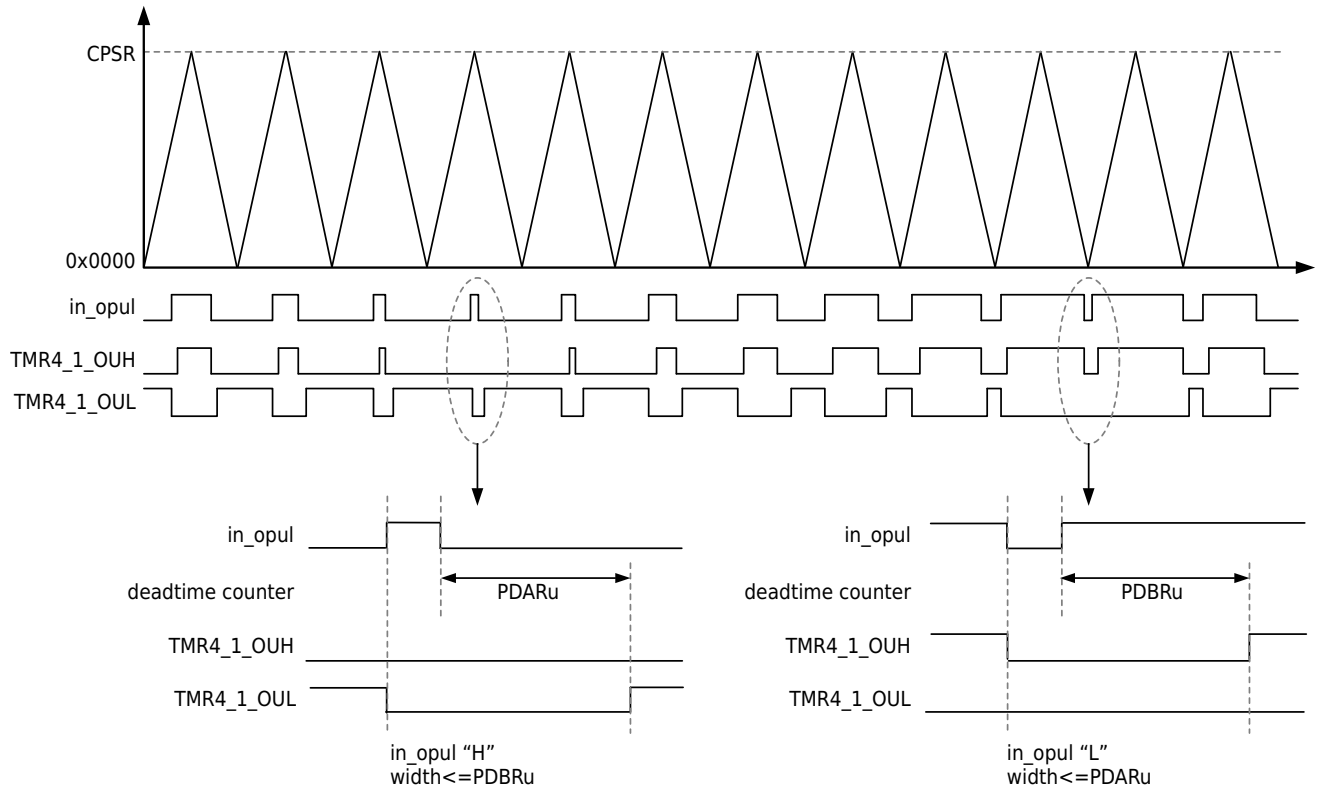


Figure 22-21 Waveform Output in Dead-Zone Timer Mode with Pulse Width Anomalies

Based on the hardware implementation of dead-zone output, pulse width monitoring can also be applied to the internal compare match signal (in\_op\*I), enabling filtering control of the in\_op\*I signal. This method, where the in\_op\*I signal undergoes pulse width filtering, is called the Dead-Zone Counter Filter Mode (POCR.PWMMD=0b10).

In the Dead-Zone Counter Filter Mode, the filter width is determined by the value set in the PFSRn register. When the pulse width of in\_op\*I exceeds the time set in the PFSRn register, the filter counter will delay the in\_op\*I signal by the time set in PFSRn before outputting it. The output will then follow the dead-zone timer mode to generate complementary PWM signals.

If the rising edge of the signal in\_op\*I is detected, the filter counter loads the PFSR register value and starts measuring the high level width of in\_op\*I. When the high level pulse width of in\_op\*I is greater than the time set by the register PFSR, after the time set by PFSR, the TMR4\_<t>\_O\*L output becomes low level, and the dead zone counter loads the set value of the PDBR register and starts counting down. When the count value becomes 0x0000, the counter stops and makes TMR4\_<t>\_O\*H output high level. If the falling edge of the signal in\_op\*I is detected, the filter counter loads the PFSR register value and starts to measure the low-level width of in\_op\*I. When the low-level pulse width of in\_op\*I is greater than the time set by the register PFSR, after PFSR After the set time, the TMR4\_<t>\_O\*H output becomes low level, and the dead zone counter loads the set value of the PDAR register and starts counting down. When the count value becomes 0x0000, the counter stops and makes TMR4\_<t>\_O\*L output



high level. When the level pulse width of  $\text{in\_op}^*l$  is less than the time set by PFSR register, the outputs  $\text{TMR4\_}<t>\_O^*H$  and  $\text{TMR4\_}<t>\_O^*L$  will remain unchanged.

The following figure shows an example of complementary PWM output in dead-zone counter filter mode.

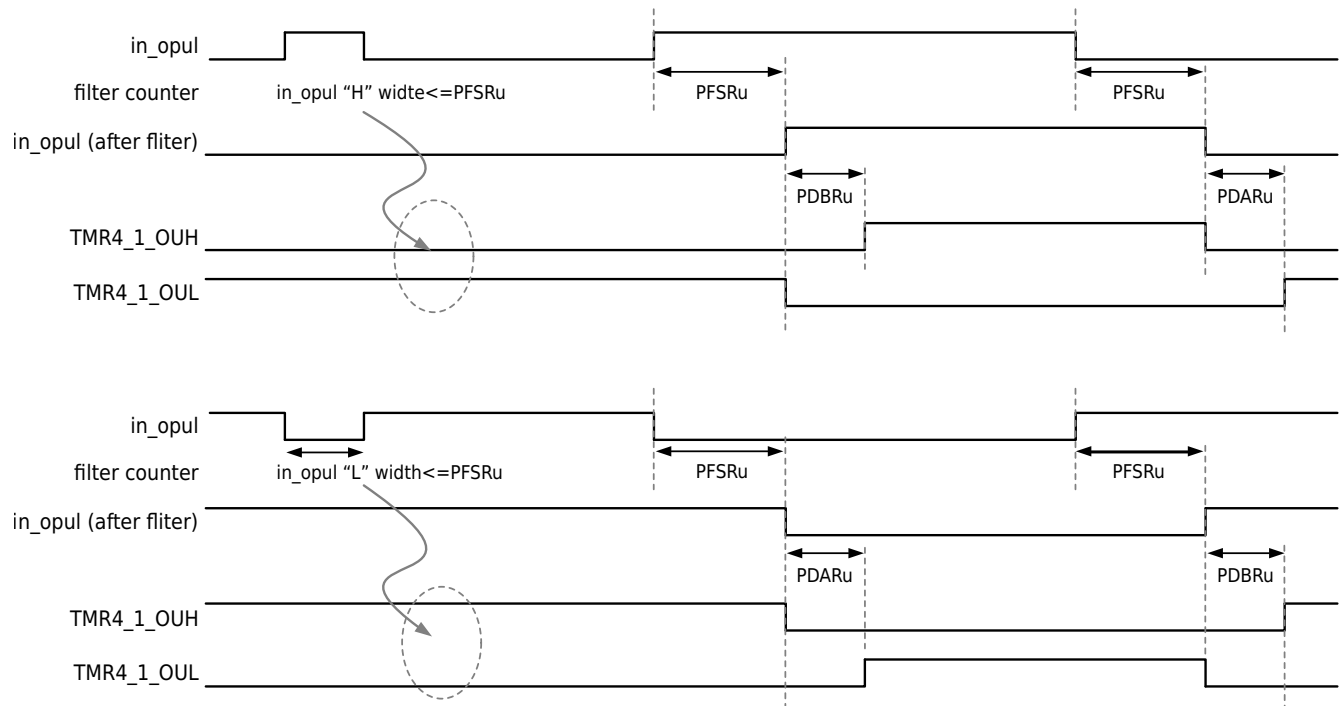


Figure 22-22 Complementary PWM Output in Dead-Zone Timer Filter Mode



#### Note

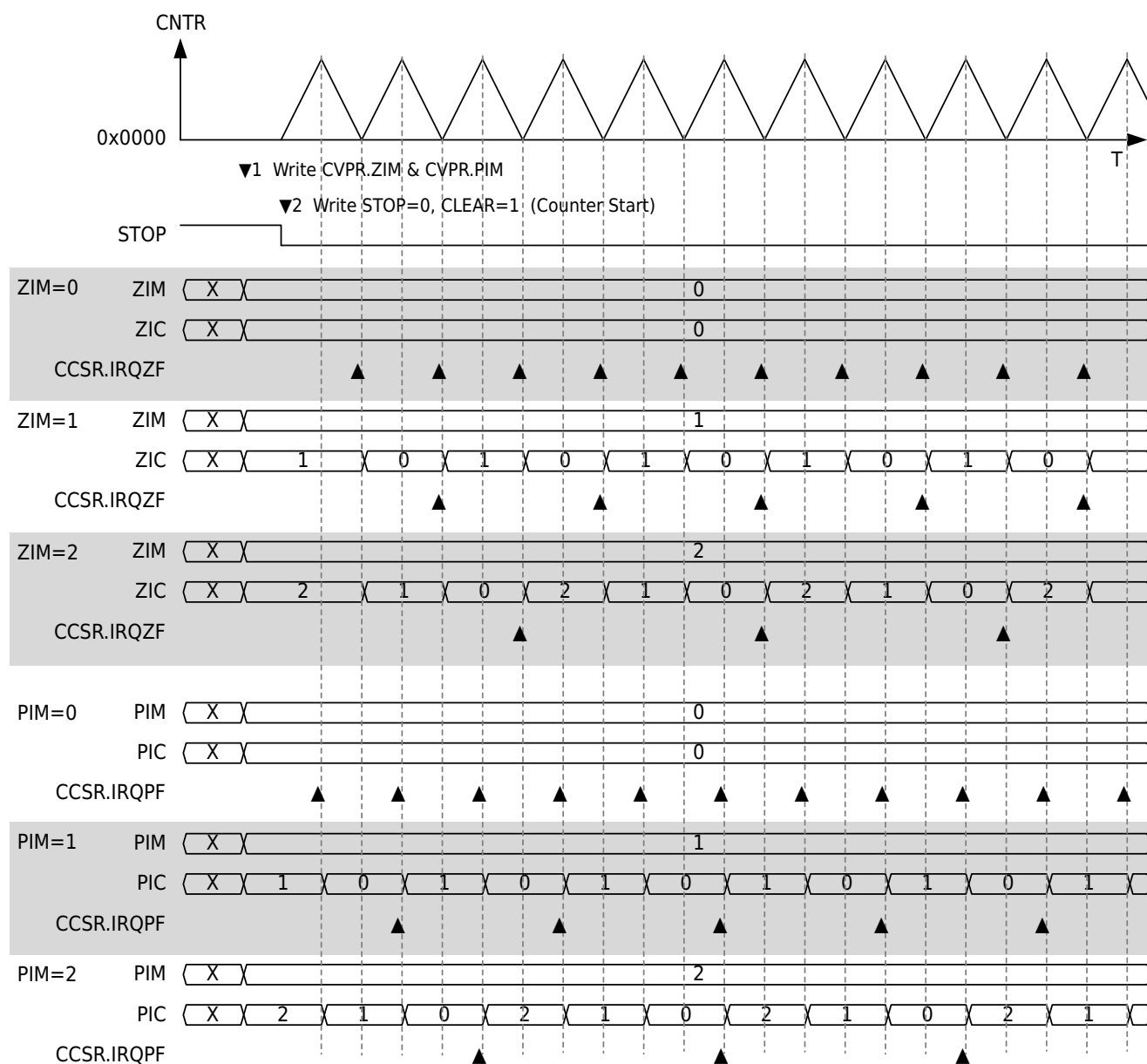
Before using software or hardware complementary PWM output mode, you need to set the MOE, OE\*H, OE\*L bits (PSCR register) to enable the output of Timer4. For detailed description, please refer to the "[PWM Status Control Register \(TMR4\\_PSCR\)](#)".

### 22.3.5 Periodic Interval Response

The underflow interrupt mask counter is used to reduce (mask) the number of times the underflow flag (CCSR.IRQZF) is set. The underflow interrupt mask counter (CVPR.ZIC) operates as a down counter and is loaded with the value set by CVPR.ZIM at the beginning. When CVPR.ZIC=0, the underflow flag (CCSR.IRQZF) is set to 1.

The overflow interrupt mask counter is used to reduce (mask) the number of times the overflow flag (CCSR.IRQPF) is set. The overflow interrupt mask counter (CVPR.PIC) operates as a down counter and is loaded with the value set by CVPR.PIM at the beginning. When CVPR.PIC=0, the overflow flag (CCSR.IRQPF) is set to 1.

The figure below is the setting timing diagram of IRQZF and IRQPF in periodic interval response.



### Figure 22-23 Periodic Interval Response Timing Diagram

- When the counter is stopped, write the initialization values of CVPR.ZIM and CVPR.PIM, and the initial values are immediately reflected in the internal counters (CVPR.ZIC, CVPR.PIC).
- Initialize and start the counter (STOP=0 and CLEAR=1). The counter starts counting up from zero after the bus is reset or CLEAR=1 is initialized from software. At this moment, the CCSR.IRQZF flag is not set immediately. After that, whenever the count value of the interrupt mask counter is 0x0000 and the count value of the counter is 0x0000 and CPSR, the flag ▲ is the moment when CCSR.IRQZF or CCSR.IRQPF is set.
- When the counter is running, if CVPR.ZIM and CVPR.PIM are written, the set value will not be immediately reflected in the interrupt mask counter (CVPR.ZIC and CVPR.PIC). If a soft reset (CLEAR=1) is written, the written CVPR.ZIM and CVPR.PIM values will be immediately loaded as the initial values of the interrupt mask counter.

The compare match event (special event output) of the SCCRM register also has a period interval response function. The figure below shows the periodic interval response output of the special event output signal.

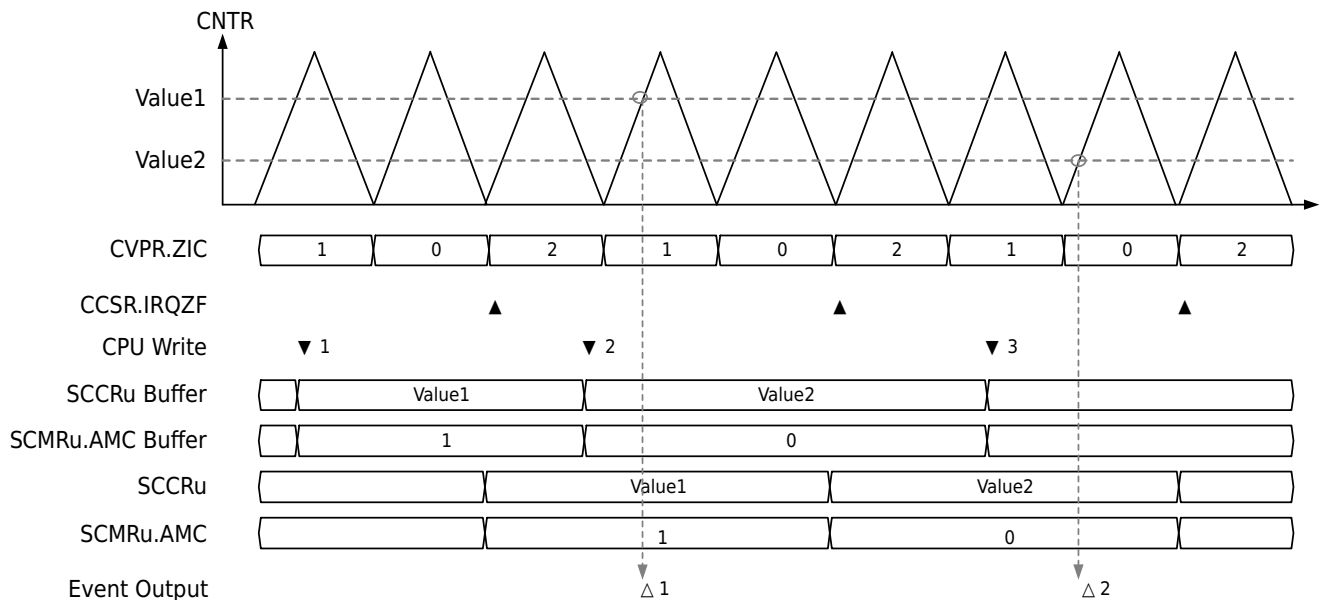


Figure 22-24 Special Event Output Signal Periodic Interval Response Output

The counter is in triangle wave counting mode, and the underflow interrupt mask counter (CVPR.ZIC) counts down from 2-0. An underflow interrupt is generated at time ▲.

At time ▼1, value1 is written to the SCCRu buffer register. Meanwhile, MZCE=1, MPCE=0, AMC=0001 are written to the SCMRu buffer register. After that, the buffer registers SCCRu and SCMRu transfer operations are performed at time ▲. Because MZCE=1 and AMC=1 and the count value=SCCRu=Value1, at time Δ1, the special event output signal is set.

At time ▼2, value2 is written to the SCCRu buffer register. Meanwhile, MZCE=1, MPCE=0, AMC=0000 are written to the SCMRu buffer register. After that, the buffer registers SCCRu and SCMRu transfer operations are performed at time ▲. Because MZCE=1 and AMC=0 and the count value=SCCRu=Value2, at time Δ2, the special event output signal is set.

### 22.3.6 EMB Control

Each Timer 4 unit has an output inactive event interface connected to the EMB event output from the EMB module. The abnormal condition event selected on this interface can be set from the EMB side (refer to the "Emergency Brake Module (EMB) " chapter).

During the normal output period of the 3 groups of PWM ports in each unit, if the EMB abnormal event from the EMB is detected, the PSCR.MOE bit is cleared asynchronously by hardware, and the output state of each PWM port changes to the preset state. According to the settings of PSCR.OS\*H and PSCR.OS\*L, the preset port state can be output high impedance, output low level or output high level.

For example, if PSCR.OSUH=0b01 is set, an EMB event is generated during the normal output period of the TMR4\_<t>\_OUH port of Timer4, and the output on the TMR4\_<t>\_OUH port changes to a high impedance state.

After the EMB exception event is cleared, if PSCR.AOE=1 at this time, PSCR.MOE will be automatically set by hardware, and then the PWM port will resume normal output; if PSCR.AOE=0 at this time, PSCR.MOE needs to be set by software to restore the PWM port to normal output.

It should be noted that the asynchronous signal of MOE is used to control the PWM output enable in hardware. When the value of MOE changes, it is necessary to wait for several clock periods to read the actual value on the bus (it is necessary to wait for 4 Timer4 bus clock periods after the EMB abnormal

event occurs, and it is necessary to wait for 6 Timer4 bus clock periods after the EMB abnormal event is resolved).

For detailed operation methods of EMB, please refer to the relevant description of the "Emergency Brake Module (EMB)" chapter.

### 22.3.7 Monitoring Output

Each Timer4 unit has a PWM period output monitoring port TMR4\_<t>\_PCT and a special event output monitoring port TMR4\_<t>\_ADSM. These monitoring ports are output to the outside, and users can flexibly control the application system based on the port change information.

The PWM period output monitoring port TMR4\_<t>\_PCT is used to monitor the current counter count state. In the triangle wave mode, each time the count reaches the peak or trough, a flip occurs on the monitoring port TMR4\_<t>\_PCT. The following figure is an example of the counting direction signal.

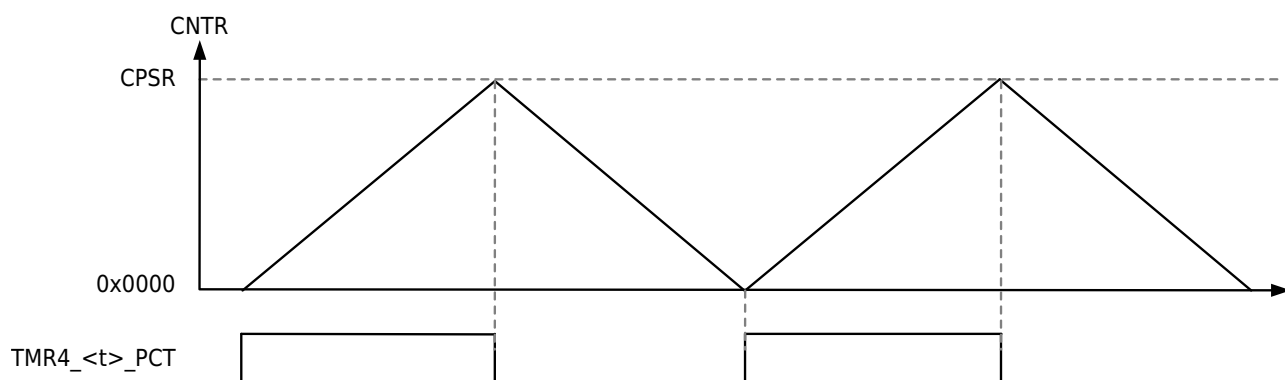


Figure 22-25 PWM Period Output Monitoring Example

The special event output monitoring port TMR4\_<t>\_ADSM can be used to monitor the special comparison event of the internal counter. Select the special comparison output channel that needs to be monitored through the SCER.EVTRS bit. When the compare reference on this channel and the counter produce a comparison match, the monitoring port TMR4\_<t>\_ADSM will be set (the output is high); when the counter count on this channel returns to zero (the sawtooth wave overflows and the triangle wave counts to the trough), the monitoring port TMR4\_<t>\_ADSM will be reset (the output is low).

It should be noted that when the configuration register TMR4\_SCSRm enables both up-counting EVT and down-counting EVT (TMR4\_SCSRm.UEN=1 and TMR4\_SCSRm.DEN=1), the synchronization signal output to the external port is only active for up-counting EVT.

The following figure shows an example of the synchronized signal output to the port in comparison start mode (SCSRm.EVTMS=0). After the EVT is enable in up-counting mode, TMR4\_<t>\_ADSM is set. After the counter underflows, TMR4\_<t>\_ADSM is cleared.

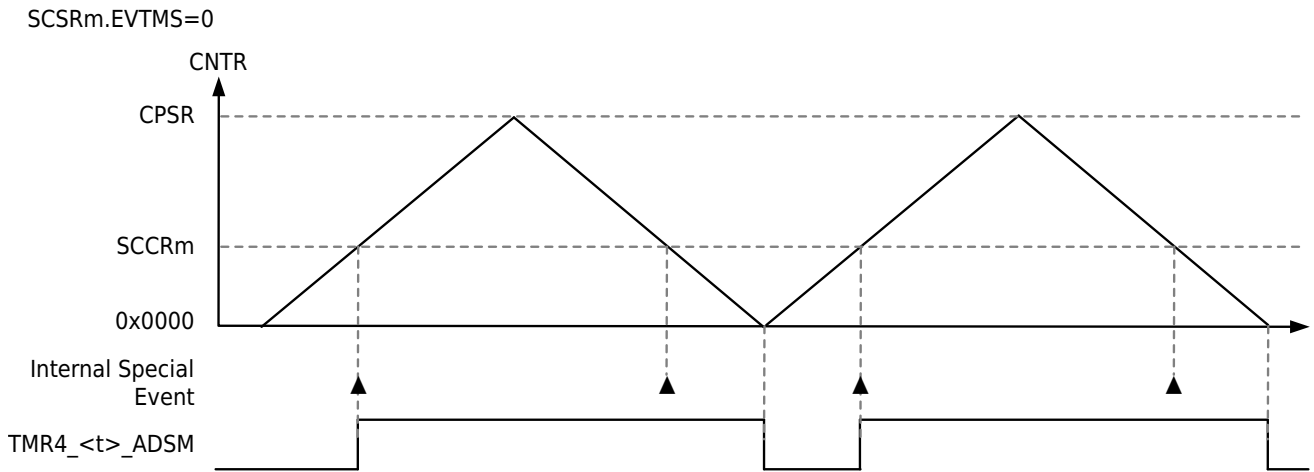


Figure 22-26 Special Event Output TMR4\_&lt;t&gt;\_ADSM Timing in Compare Start Mode

### 22.3.8 Synchronous Starting

This product is equipped with three units of Timer4, which can realize synchronous software startup. Unit 1 can start units 2 and 3 synchronously.

When the CCSR.SYNST bit in the synchronized unit m (m=2, 3) is set to 1, the synchronized start function of the unit is enabled. At this time, if the software sets the CCSR.STOP bit of unit 1 to 0, the counter of synchronized unit m starts counting synchronously with the counter of unit 1.

In addition, the product is equipped with a count synchronization enable register TMR\_SYNENRx (x=1-2) to synchronously start all the timers (including Timer0, Timer2, Timer4, Timer6, TimerA) carried by the product. For usage and register description, please refer to the description in the "General Timer (Timer0) " section. Please do not use this function when this timer is counting asynchronously.

### 22.3.9 Interrupt and Event

#### General Count Compare Match Interrupt and Event

There are a total of 6 general compare reference registers (OCCRm), each of which can generate a compare match enable signal when matched with the counter value. When the count compare matches, the OCSRn.OCFH and OCSRn.OCFL bits in the common control status register (OCSRn) will be set to 1 respectively. At this time, if OCSRn.OCIEH and OCSRn.OCIEL are set to enable interrupts, the corresponding interrupt request (TMR4\_<t>\_GCMmn, m=U, V, W; n=H, L) will also be triggered.

The general count compare match event can be used to trigger other modules. For the event number, please refer to the "Interrupt Event List" in the "Interrupt Controller (INTC) " chapter.

#### Count Period Match Interrupt and Event

When the sawtooth wave counts up to the overflow point, the triangle wave counts to the valley point, or the triangle wave counts to the peak point, the CCSR.IRQPF or CCSR.IRQZF bit will be set to 1. At this time, if the CCSR.IRQPEN or CCSR.IRQZEN bit is set to enable interrupt, the count period match interrupt (TMR4\_<t>\_GOVF and TMR4\_<t>\_GUDF) can be triggered at the corresponding time point.

The count period match event can be used to trigger other modules. For event numbers, refer to the 'Interrupt Event List' in the 'Interrupt Controller (INTC) ' chapter.

#### Reload Count Match Interrupt and Event

When the reload function is enabled, TMR4\_PFSRn serves as the period count value for the reload timer. When counting is enabled, the counter is reloaded with the value in TMR4\_PFSRn and performs

a decrement operation. After completing one period, a reload count interrupt request is generated, the RCSR.RTIFU, RCSR.RTIFV, RCSR.RTIFW bits are set to 1, respectively. If the interrupt masks RCSR.RTIDU, RCSR.RTIDV, RCSR.RTIDW are disabled, the corresponding reload compare match interrupt requests (TMR4\_<t>\_RLOm, where m=U, V, W) will be triggered.

Reload compare match events can be used to trigger other modules; for the event numbers, refer to the "Interrupt Event List" in the "Interrupt Controller (INTC) " chapter.

### Special Compare Match Events

The 6 special compare reference registers (SCCRm) of Timer4 generate 6 special event output signals corresponding to them, which can be used to select and trigger other modules, such as starting the ADC.

During the clock counting process, if the special compare reference (SCCRm) has a count comparison match event (TMR4\_<t>\_SCM0-5), a corresponding enable request signal will be generated. This request signal can be assigned to any EVT event output signal (configured via the SCSR.EVTOS bit) to trigger other modules.

The output of this event request signal can be configured for either Comparison Start Mode or Delay Start Mode. In comparison start mode (SCSR.EVTMS=0), after the count compare match event of SCCR is generated, the special event output signal is directly output; in delay start mode (SCSR.EVTMS=1), after the count compare match event of OCCR\*h or OCCR\*l (selected by SCSR.EVTDS bit) is generated, the special event output signal is output after the reference period time set by SCCR. The figure below illustrates an example of the event request signal output in Delay Start Mode.

Special compare match events can be used to trigger other modules. For event numbers, please refer to the "Interrupt Event List" in the "Interrupt Controller (INTC) " chapter.

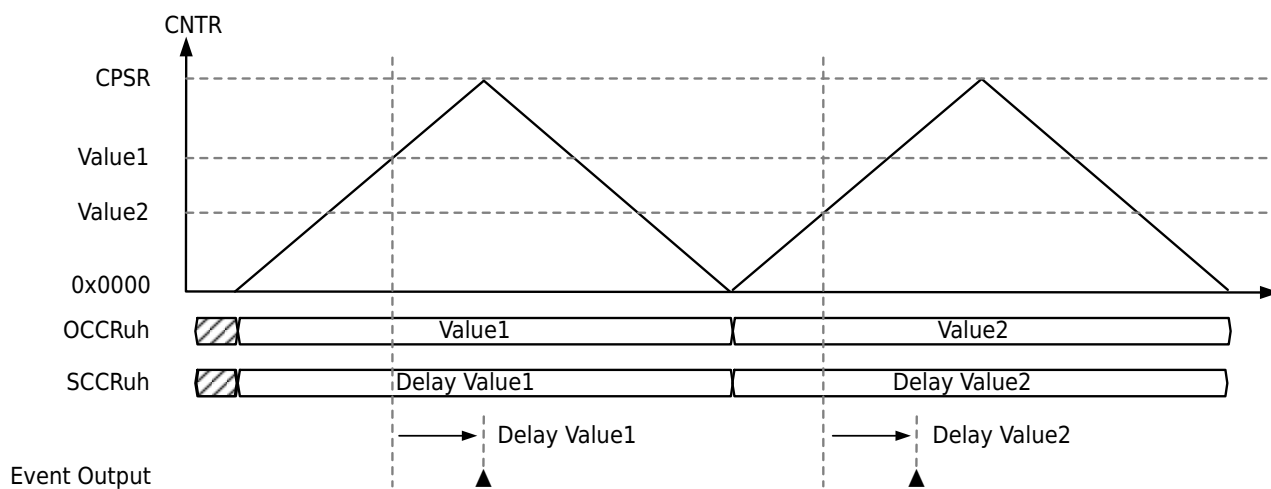


Figure 22-27 Output Timing of Special Event Output Signal in Delayed Start Mode

#### Note

During delay counting, if another match event between OCCR and the counter occurs, the delay counter reloads and begins decrementing again. Therefore, if the interval between OCCR match events is shorter than the delay period defined by SCCR, the special event output request signal may never be generated.

## 22.4 Register Description

### 22.4.1 Register List

- Timer4\_1 base address: 0x40038000
- Timer4\_2 base address: 0x40038400
- Timer4\_3 base address: 0x40038800



#### Note

Timer4 is the name of the timer, and Timer4\_<t> (<t>: 1-3) is the timer unit.

Table 22-5 General Control Timer (Timer4) Register Offset

| Offset | Register    | Description                           | Width |
|--------|-------------|---------------------------------------|-------|
| 0x0046 | TMR4_CNTR   | Count Value Register                  | 16    |
| 0x0042 | TMR4_CPSR   | Period Reference Register             | 16    |
| 0x0048 | TMR4_CCSR   | Control Status Register               | 16    |
| 0x004A | TMR4_CVPR   | Valid Period Register                 | 16    |
| 0x0002 | TMR4_OCCRuh | General Compare Reference Register UH | 16    |
| 0x0006 | TMR4_OCCRuI | General Compare Reference Register UL | 16    |
| 0x000A | TMR4_OCCRvh | General Compare Reference Register VH | 16    |
| 0x000E | TMR4_OCCRvI | General Compare Reference Register VL | 16    |
| 0x0012 | TMR4_OCCRwh | General Compare Reference Register WH | 16    |
| 0x0016 | TMR4_OCCRwI | General Compare Reference Register WL | 16    |
| 0x0018 | TMR4_OCSRu  | General Control Status Register U     | 16    |
| 0x001C | TMR4_OCSRv  | General Control Status Register V     | 16    |
| 0x0020 | TMR4_OCSRw  | General Control Status Register W     | 16    |
| 0x001A | TMR4_OCERu  | General Extended Control Register U   | 16    |
| 0x001E | TMR4_OCERv  | General Extended Control Register V   | 16    |
| 0x0022 | TMR4_OCERw  | General Extended Control Register W   | 16    |
| 0x0024 | TMR4_OCMRuh | General Mode Control Register UH      | 16    |
| 0x0028 | TMR4_OCMRuI | General Mode Control Register UL      | 32    |
| 0x002C | TMR4_OCMRvh | General Mode Control Register VH      | 16    |
| 0x0030 | TMR4_OCMRvI | General Mode Control Register VL      | 32    |
| 0x0034 | TMR4_OCMRwh | General Mode Control Register WH      | 16    |

| Offset | Register    | Description                           | Width |
|--------|-------------|---------------------------------------|-------|
| 0x0038 | TMR4_OCMRwl | General Mode Control Register WL      | 32    |
| 0x00B2 | TMR4_SCCRuh | Special Compare Reference Register UH | 16    |
| 0x00B6 | TMR4_SCCRul | Special Compare Reference Register UL | 16    |
| 0x00BA | TMR4_SCCRvh | Special Compare Reference Register VH | 16    |
| 0x00BE | TMR4_SCCrvl | Special Compare Reference Register VL | 16    |
| 0x00C2 | TMR4_SCCRwh | Special Compare Reference Register WH | 16    |
| 0x00C6 | TMR4_SCCRwl | Special Compare Reference Register WL | 16    |
| 0x00C8 | TMR4_SCSRuh | Special Control Status Register UH    | 16    |
| 0x00CC | TMR4_SCSRul | Special Control Status Register UL    | 16    |
| 0x00D0 | TMR4_SCSRvh | Special Control Status Register VH    | 16    |
| 0x00D4 | TMR4_SCCrvl | Special Control Status Register VL    | 16    |
| 0x00D8 | TMR4_SCSRwh | Special Control Status Register WH    | 16    |
| 0x00DC | TMR4_SCSRwl | Special Control Status Register WL    | 16    |
| 0x00CA | TMR4_SCMRuh | Special Mode Control Register UH      | 16    |
| 0x00CE | TMR4_SCMRul | Special Mode Control Register UL      | 16    |
| 0x00D2 | TMR4_SCMRvh | Special Mode Control Register VH      | 16    |
| 0x00D6 | TMR4_SCMRvl | Special Mode Control Register VL      | 16    |
| 0x00DA | TMR4_SCMRwh | Special Mode Control Register WH      | 16    |
| 0x00DE | TMR4_SCMRwl | Special Mode Control Register WL      | 16    |
| 0x00E4 | TMR4_SCER   | Special Extended Control Register     | 16    |
| 0x0098 | TMR4_POCRu  | PWM Basic Control Register U          | 16    |
| 0x009C | TMR4_POCRv  | PWM Basic Control Register V          | 16    |
| 0x00A0 | TMR4_POCRw  | PWM Basic Control Register W          | 16    |
| 0x00E0 | TMR4_PSCR   | PWM Status Control Register           | 32    |
| 0x0082 | TMR4_PFSRu  | PWM Filter Control Register U         | 16    |
| 0x008A | TMR4_PFSRv  | PWM Filter Control Register V         | 16    |
| 0x0092 | TMR4_PFSRw  | PWM Filter Control Register W         | 16    |
| 0x0084 | TMR4_PDARu  | PWM Dead Zone Control Register AU     | 16    |
| 0x0086 | TMR4_PDBRu  | PWM Dead Zone Control Register BU     | 16    |



| Offset | Register                   | Description                       | Width |
|--------|----------------------------|-----------------------------------|-------|
| 0x008C | <a href="#">TMR4_PDARv</a> | PWM Dead Zone Control Register AV | 16    |
| 0x008E | <a href="#">TMR4_PDBRv</a> | PWM Dead Zone Control Register BV | 16    |
| 0x0094 | <a href="#">TMR4_PDARw</a> | PWM Dead Zone Control Register AW | 16    |
| 0x0096 | <a href="#">TMR4_PDBRw</a> | PWM Dead Zone Control Register BW | 16    |
| 0x00A4 | <a href="#">TMR4_RCSR</a>  | Reload Control Status Register    | 16    |

**Note**

In the detailed description of registers that follow in this section, m=uh, ul, vh, vl, wh, wl, and n=u, v, w. The registers indicated by m correspond to the output control of ports TMR4\_<t>\_OUH, TMR4\_<t>\_OUL, TMR4\_<t>\_OVH, TMR4\_<t>\_OVL, TMR4\_<t>\_OWH, TMR4\_<t>\_OWL when the function is implemented; The registers indicated by n correspond to the output controls of ports TMR4\_<t>\_OUx, TMR4\_<t>\_OVx, TMR4\_<t>\_OWx, etc. when the function is implemented, where x=H or L, and the specific control of H or L has corresponding symmetrical bits in these registers.

### 22.4.2 Count Value Register (TMR4\_CNTR)

| Offset | Bit Position |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0x0046 | 15           | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x0000       |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | CNTR         |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access | RW           |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit  | Name | Function              | Description                                                                                                                                                                                                                                                                                                                                                                         | Access |
|------|------|-----------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 15:0 | CNTR | Counter Current Value | <p>When counting is stopped, the counter value can be initialized by writing a value to this register. During counting, this bit indicates the current counter value.</p> <div>  <b>Notice</b><br/>           Writing to this register is prohibited while counting is in progress.         </div> | RW     |

### 22.4.3 Period Reference Register (TMR4\_CPSR)

| Offset | Bit Position |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0x0042 | 15           | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0xFFFF       |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | CPSR         |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access | RW           |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit  | Name | Function                 | Description                                                                                                                                                                                                                                                                           | Access |
|------|------|--------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 15:0 | CPSR | General Period Reference | <p>Counter period value</p> <div>  <b>Note</b><br/>           When reading data from this address area, the value read is not from the buffer register, but from the CPSR register.         </div> | RW     |

## 22.4.4 Control Status Register (TMR4\_CCSR)

| Offset | Bit Position |       |        |       |          |    |       |        |       |      |      |       |   |   |       |   |
|--------|--------------|-------|--------|-------|----------|----|-------|--------|-------|------|------|-------|---|---|-------|---|
| 0x0048 | 15           | 14    | 13     | 12    | 11       | 10 | 9     | 8      | 7     | 6    | 5    | 4     | 3 | 2 | 1     | 0 |
| Reset  | 0x0040       |       |        |       |          |    |       |        |       |      |      |       |   |   |       |   |
| Name   | ECKEN        | IRQZF | IRQZEN | SYNST | Reserved |    | IRQPF | IRQPEN | BUFEN | STOP | MODE | CLEAR |   |   | CKDIV |   |
| Access | RW           | RW    | RW     | RW    |          |    | RW    | RW     | RW    | RW   | RW   | RW    |   |   | RW    |   |

| Bit   | Name     | Function                   | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | Access |
|-------|----------|----------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 15    | ECKEN    | Clock Source Selection     | <ul style="list-style-type: none"> <li>0b0: Internal PCLK0 clock</li> <li>0b1: External TMR4_&lt;t&gt;_CLK port input clock</li> </ul> <div>  <b>Note</b> <ul style="list-style-type: none"> <li>This bit is set when the counter stops.</li> <li>When operating with an external input clock, after the counter operation starts, the first edge of the external input clock, whether rising or falling, is ignored and the counting operation starts from the subsequent edge.</li> </ul> </div> | RW     |
| 14    | IRQZF    | Underflow Status           | <ul style="list-style-type: none"> <li>0b0: No counter underflow occurred</li> <li>0b1: Counter underflow occurred</li> </ul> <div>  <b>Note</b> <ul style="list-style-type: none"> <li>When using the period interval response function, the condition for setting this bit is determined by the period interval counter settings configured in the CVPR.</li> <li>When the counter is reset by the bus or when CLEAR=1 is written, the IRQZF bit will not be set.</li> </ul> </div>            | RW     |
| 13    | IRQZEN   | Underflow Interrupt Enable | <ul style="list-style-type: none"> <li>0b0: Disables the IRQZF interrupt from being generated to the CPU</li> <li>0b1: Enables the IRQZF interrupt from being generated to the CPU</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                       | RW     |
| 12    | SYNST    | Synchronous Start Enabled  | <ul style="list-style-type: none"> <li>0b0: The synchronization start function with unit 1 is disabled for this unit</li> <li>0b1: The synchronization start function with unit 1 is enabled for this unit</li> </ul> <div>  <b>Note</b> <p>Only unit 2 and unit 3 support the SYNST function bit. For unit 1, this bit is reserved and should be kept at its initial value of 0.</p> </div>                                                                                                     | RW     |
| 11:10 | Reserved | Reserved                   | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | -      |

| Bit | Name   | Function                       | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | Access |
|-----|--------|--------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 9   | IRQPF  | Overflow Status                | <ul style="list-style-type: none"> <li>● 0b0: No count overflow occurred</li> <li>● 0b1: Count overflow occurred</li> </ul> <div>  <b>Note</b> <ul style="list-style-type: none"> <li>● When using the period interval response function, the condition for setting this bit is determined by the period interval counter settings configured in the CVPR.</li> <li>● When the counter is reset by the bus or when CLEAR=1 is written, the IRQPF bit will not be set.</li> </ul> </div>                                                                                                                                                                                                                                                                             | RW     |
| 8   | IRQPEN | Overflow Interrupt Enabled     | <ul style="list-style-type: none"> <li>● 0b0: Disable the IRQPF interrupt from being generated to the CPU</li> <li>● 0b1: Enable the IRQPF interrupt from being generated to the CPU</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | RW     |
| 7   | BUFEN  | Buffer Enabled                 | <ul style="list-style-type: none"> <li>● 0b0: Disable CPSR buffer function</li> <li>● 0b1: Enable CPSR buffer function</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | RW     |
| 6   | STOP   | Counter Enabled                | <ul style="list-style-type: none"> <li>● 0b0: Counter started</li> <li>● 0b1: Counter stopped</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | RW     |
| 5   | MODE   | Waveform Mode                  | <ul style="list-style-type: none"> <li>● 0b0: Sawtooth wave mode (supports only up counting)</li> <li>● 0b1: Triangle wave mode</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | RW     |
| 4   | CLEAR  | Counter Clear                  | Write 1 to clear the counter, write 0 has no effect; read always returns 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | RW     |
| 3:0 | CKDIV  | Count Clock Division Frequency | <p>This bit indicates the count clock frequency division of the basic counter</p> <ul style="list-style-type: none"> <li>● 0b0000: Count clock is PCLK0</li> <li>● 0b0001: Count clock is PCLK0/2</li> <li>● 0b0010: Count clock is PCLK0/4</li> <li>● 0b0011: Count clock is PCLK0/8</li> <li>● 0b0100: Count clock is PCLK0/16</li> <li>● 0b0101: Count clock is PCLK0/32</li> <li>● 0b0110: Count clock is PCLK0/64</li> <li>● 0b0111: Count clock is PCLK0/128</li> <li>● 0b1000: Count clock is PCLK0/256</li> <li>● 0b1001: Count clock is PCLK0/512</li> <li>● 0b1010: Count clock is PCLK0/1024</li> <li>● Others: Setting prohibited</li> </ul> <div>  <b>Note</b> <p>When the clock source is internal PCLK0, frequency division is enabled.</p> </div> | RW     |

## 22.4.5 Valid Period Register (TMR4\_CVPR)

| Offset | Bit Position |    |    |    |     |    |   |   |     |   |   |   |     |   |   |   |
|--------|--------------|----|----|----|-----|----|---|---|-----|---|---|---|-----|---|---|---|
| 0x004A | 15           | 14 | 13 | 12 | 11  | 10 | 9 | 8 | 7   | 6 | 5 | 4 | 3   | 2 | 1 | 0 |
| Reset  | 0x0000       |    |    |    |     |    |   |   |     |   |   |   |     |   |   |   |
| Name   | PIC          |    |    |    | ZIC |    |   |   | PIM |   |   |   | ZIM |   |   |   |
| Access | RW           |    |    |    | RW  |    |   |   | RW  |   |   |   | RW  |   |   |   |

| Bit   | Name | Function                         | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | Access |
|-------|------|----------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 15:12 | PIC  | Overflow Interrupt Mask Status   | <p>PIC is a down counter. When CNTR counts overflow interrupts, the value of PIC is reduced by 1. Reading PIC can obtain the number of overflow interrupts that need to be masked.</p> <p>Write condition: counter stopped state (CCSR.STOP=1)</p> <p>The value of PIM is loaded into PIC if one of the following is met:</p> <ol style="list-style-type: none"> <li>1. Counter clearing action occurs (CCSR.CLEAR writes 1)</li> <li>2. PIC=0 and CNTR count overflow interrupt occurs</li> </ol>   | RW     |
| 11:8  | ZIC  | Underflow Interrupt Mask Status  | <p>ZIC is a down counter. When CNTR counts underflow interrupts, the value of ZIC decreases by 1. Reading ZIC can obtain the number of underflow interrupts that need to be masked.</p> <p>Write condition: counter stopped state (CCSR.STOP=1)</p> <p>The value of ZIM is loaded into ZIC if one of the following is met:</p> <ol style="list-style-type: none"> <li>1. Counter clearing action occurs (CCSR.CLEAR writes 1)</li> <li>2. ZIC=0 and CNTR count underflow interrupt occurs</li> </ol> | RW     |
| 7:4   | PIM  | Overflow Interrupt Mask Setting  | <p>Set the number of overflow interrupts to be masked</p> <p>The value of PIM is loaded into PIC if one of the following is met:</p> <ol style="list-style-type: none"> <li>1. Counter clearing action occurs (CCSR.CLEAR writes 1)</li> <li>2. PIC=0 and CNTR count overflow interrupt occurs</li> </ol>                                                                                                                                                                                            | RW     |
| 3:0   | ZIM  | Underflow Interrupt Mask Setting | <p>Set the number of underflow interrupts to be masked</p> <p>The value of ZIM is loaded into ZIC if one of the following is met:</p> <ol style="list-style-type: none"> <li>1. Counter clearing action occurs (CCSR.CLEAR writes 1)</li> <li>2. ZIC=0 and CNTR count underflow interrupt occurs</li> </ol>                                                                                                                                                                                          | RW     |

22.4.6 General Compare Reference Register (TMR4\_OCCRm)

Offset: 0x0002 (TMR4\_OCCRUh), 0x0006 (TMR4\_OCCRul), 0x000A (TMR4\_OCCRVh), 0x000E (TMR4\_OCCRVl), 0x0012 (TMR4\_OCCRwh), 0x0016 (TMR4\_OCCRWl),

| Offset | Bit Position |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| -      | 15           | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x0000       |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | OCCR         |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access | RW           |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit  | Name | Function                  | Description                                                                                                                                                                                                                                                             | Access |
|------|------|---------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 15:0 | OCCR | General Compare Reference | <div>General compare reference</div> <div> <b>Note</b><br/>When reading data from this address area, the value read is not from the buffer register, but from the OCCR register.</div> | RW     |

## 22.4.7 General Control Status Register (TMR4\_OCSRn)

Offset: 0x0018 (TMR4\_OCSRu), 0x001C(TMR4\_OCSRv), 0x0020(TMR4\_OCSRw)

| Offset | Bit Position |    |    |    |    |    |   |   |      |      |       |       |      |      |      |      |
|--------|--------------|----|----|----|----|----|---|---|------|------|-------|-------|------|------|------|------|
| -      | 15           | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7    | 6    | 5     | 4     | 3    | 2    | 1    | 0    |
| Reset  | 0xFF00       |    |    |    |    |    |   |   |      |      |       |       |      |      |      |      |
| Name   | Reserved     |    |    |    |    |    |   |   | OCFL | OCFH | OCIEL | OCIEH | OCPL | OCPH | OCEL | OCEH |
| Access |              |    |    |    |    |    |   |   | RW   | RW   | RW    | RW    | RW   | RW   | RW   | RW   |

| Bit  | Name     | Function                                         | Description                                                                                                                                                                                                                                                                                                                                                 | Access |
|------|----------|--------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 15:8 | Reserved | Reserved                                         | All bits read as 1, Write as 1                                                                                                                                                                                                                                                                                                                              | -      |
| 7    | OCFL     | Count match flag L                               | <ul style="list-style-type: none"> <li>0b0: The counter value is not equal to the OCCR*L setting value</li> <li>0b1: The counter value is equal to the OCCR*L setting value</li> </ul> <div>  <b>Note</b><br/>This bit is enabled only when OCEL=1.         </div>         | RW     |
| 6    | OCFH     | Count match flag H                               | <ul style="list-style-type: none"> <li>0b0: The counter value is not equal to the OCCR*h setting value</li> <li>0b1: The counter count value is equal to the OCCR*h setting value</li> </ul> <div>  <b>Note</b><br/>This bit is enabled only when OCEH=1.         </div> | RW     |
| 5    | OCIEL    | Count match interrupt enable L                   | <ul style="list-style-type: none"> <li>0b0: When OCFL is set, no interrupt occurs</li> <li>0b1: When OCFL is set, an interrupt occurs</li> </ul>                                                                                                                                                                                                            | RW     |
| 4    | OCIEH    | Count match interrupt enable H                   | <ul style="list-style-type: none"> <li>0b0: When OCFH is set, no interrupt occurs</li> <li>0b1: When OCFH is set, an interrupt occurs</li> </ul>                                                                                                                                                                                                            | RW     |
| 3    | OCPL     | Port status L when comparison output is disabled | Write: <ul style="list-style-type: none"> <li>0b0: When OCEL=0, in_op*L outputs low level</li> <li>0b1: When OCEL=0, in_op*L outputs high level</li> </ul> Read: <ul style="list-style-type: none"> <li>0b0: The current state of in_op*L is low level</li> <li>0b1: The current state of in_op*L is high level</li> </ul>                                  | RW     |
| 2    | OCPL     | Port status H when comparison output is disabled | Write: <ul style="list-style-type: none"> <li>0b0: When OCEH=0, in_op*h outputs low level</li> <li>0b1: When OCEH=0, in_op*h outputs high level</li> </ul> Read: <ul style="list-style-type: none"> <li>0b0: The current state of in_op*h is low level</li> <li>0b1: The current state of in_op*h is high level</li> </ul>                                  | RW     |

| Bit | Name | Function                | Description                                                                                                                                                                                                                                                                   | Access |
|-----|------|-------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 1   | OCEL | Port output selection L | <ul style="list-style-type: none"> <li>0b0: When compare output is disabled, the state of in_op*I port is determined by OCPL.</li> <li>0b1: When compare output is enabled, the state of in_op*I port is determined by the settings of OCMR*I and the OCFL status.</li> </ul> | RW     |
| 0   | OCEH | Port output selection H | <ul style="list-style-type: none"> <li>0b0: When compare output is disabled, the state of in_op*h port is determined by OCPH.</li> <li>0b1: When compare output is enabled, the state of in_op*h port is determined by the settings of OCMR*h and the OCFH status.</li> </ul> | RW     |



## 22.4.8 General Extended Control Register (TMR4\_OCERn)

Offset: 0x001A (TMR4\_OCERu), 0x001E(TMR4\_OCERv), 0x0022(TMR4\_OCERw)

| Offset | Bit Position |    |       |       |      |      |      |      |         |   |         |   |         |   |         |   |
|--------|--------------|----|-------|-------|------|------|------|------|---------|---|---------|---|---------|---|---------|---|
| -      | 15           | 14 | 13    | 12    | 11   | 10   | 9    | 8    | 7       | 6 | 5       | 4 | 3       | 2 | 1       | 0 |
| Reset  | 0x0000       |    |       |       |      |      |      |      |         |   |         |   |         |   |         |   |
| Name   | Reserved     |    | MCECL | MCECH | LMML | LMMH | LMCL | LMCH | MLBUFEN |   | MHBUFEN |   | CLBUFEN |   | CHBUFEN |   |
| Access |              |    | RW    | RW    | RW   | RW   | RW   | RW   | RW      |   | RW      |   | RW      |   | RW      |   |

| Bit   | Name     | Function                                  | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | Access |
|-------|----------|-------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 15:14 | Reserved | Reserved                                  | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | -      |
| 13    | MCECL    | Count match additional condition enable L | <p>Additional Condition: Counter overflow (CNTR=CPSR) and OCCR*I &gt; CPSR.</p> <ul style="list-style-type: none"> <li>0b0: Disable this condition as a match condition.</li> <li>0b1: Enable this condition as a match condition; when the condition is satisfied, OCCR*I will match the counter value.</li> </ul> <div>  <b>Note</b> <ol style="list-style-type: none"> <li>This bit can only be modified when the output comparison function is disabled (OCSR.OCEL=0).</li> <li>For details, refer to the precautions in the TMR4_OCMRm register description.</li> </ol> </div>     | RW     |
| 12    | MCECH    | Count match additional condition enable H | <p>Additional Condition: Count overflow (CNTR=CPSR) and OCCR*h &gt; CPSR.</p> <ul style="list-style-type: none"> <li>0b0: Disable this condition as a matching condition.</li> <li>0b1: Enable this condition as a matching condition. When the condition is met, OCCR*h will generate a count match.</li> </ul> <div>  <b>Note</b> <ol style="list-style-type: none"> <li>This bit can only be modified when the output comparison function is disabled (OCSR.OCEH=0).</li> <li>For detailed information, refer to the notes in the TMR4_OCMRm register description.</li> </ol> </div> | RW     |

| Bit | Name | Function                                          | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                              | Access |
|-----|------|---------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 11  | LMML | OCMR*I Periodic interval response function link L | <ul style="list-style-type: none"> <li>● 0b0: The periodic interval response function link is disabled. The buffer transmission of OCMR*I is determined by the MLBUFEN setting.</li> <li>● 0b1: The periodic interval response function link is enabled. The buffer transmission of OCMR*I, in addition to the MLBUFEN setting, must also meet the conditions where CVPR.PIC=0b0000 (when counting overflows) or CVPR.ZIC=0b0000 (when counting underflows).</li> </ul>  | RW     |
| 10  | LMMH | OCMR*h periodic interval response function link H | <ul style="list-style-type: none"> <li>● 0b0: The periodic interval response function link is disabled. The buffer transmission of OCMR*h is determined by the MLBUFEN setting.</li> <li>● 0b1: The periodic interval response function link is enabled. In addition to the MLBUFEN setting, the buffer transmission of OCMR*h must also satisfy the condition where CVPR.PIC=0b0000 (when counting overflows) or CVPR.ZIC=0b0000 (when counting underflows).</li> </ul> | RW     |
| 9   | LMCL | OCCR*I periodic interval response function link L | <ul style="list-style-type: none"> <li>● 0b0: The periodic interval response function link is disabled. The buffer transmission of OCMR*I is determined by the MLBUFEN setting.</li> <li>● 0b1: The periodic interval response function link is enabled. In addition to the MLBUFEN setting, the buffer transmission of OCCR*I must also satisfy the condition where CVPR.PIC=0b0000 (when counting overflows) or CVPR.ZIC=0b0000 (when counting underflows).</li> </ul> | RW     |
| 8   | LMCH | OCCR*h periodic interval response function link H | <ul style="list-style-type: none"> <li>● 0b0: The periodic interval response function link is disabled. The buffer transmission of OCCR*h is determined by the MLBUFEN setting.</li> <li>● 0b1: The periodic interval response function link is enabled. In addition to the MLBUFEN setting, the buffer transmission of OCCR*h must also satisfy the condition where CVPR.PIC=0b0000 (when counting overflows) or CVPR.ZIC=0b0000 (when counting underflows).</li> </ul> | RW     |

| Bit | Name    | Function                   | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | Access |
|-----|---------|----------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 7:6 | MLBUFEN | OCMR*I buffer transmission | <ul style="list-style-type: none"> <li>● 0b00: The value of the OCMR*I buffer register is directly written to OCMR*I</li> <li>● 0b01: The value of the OCMR*I buffer register is written to OCMR*I when the counter underflows</li> <li>● 0b10: The value of the OCMR*I buffer register is written to OCMR*I when the counter overflows</li> <li>● 0b11: The value of the OCMR*I buffer register is written to OCMR*I when the counter either overflows or underflows</li> </ul>     | RW     |
| 5:4 | MHBUFEN | OCMR*h buffer transmission | <ul style="list-style-type: none"> <li>● 0b00: The value of the OCMR*h buffer register is directly written to OCMR*h</li> <li>● 0b01: The value of the OCMR*h buffer register is written to OCMR*h when the counter underflows</li> <li>● 0b10: The value of the OCMR*h buffer register is written to OCMR*h when the counter overflows</li> <li>● 0b11: The value of the OCMR*h buffer register is written to OCMR*h when the counter either overflows or underflows</li> </ul>     | RW     |
| 3:2 | CLBUFEN | OCCR*I buffer transmission | <ul style="list-style-type: none"> <li>● 0b00: The value of the OCCR*I buffer register is directly written to OCCR*I.</li> <li>● 0b01: The value of the OCCR*I buffer register is written to OCCR*I when the counter underflows.</li> <li>● 0b10: The value of the OCCR*I buffer register is written to OCCR*I when the counter overflows.</li> <li>● 0b11: The value of the OCCR*I buffer register is written to OCCR*I when the counter either overflows or underflows.</li> </ul> | RW     |
| 1:0 | CHBUFEN | OCCR*h buffer transmission | <ul style="list-style-type: none"> <li>● 0b00: The value of the OCCR*h buffer register is directly written to OCCR*h</li> <li>● 0b01: The value of the OCCR*h buffer register is written to OCCR*h when the counter underflows</li> <li>● 0b10: The value of the OCCR*h buffer register is written to OCCR*h when the counter overflows</li> <li>● 0b11: The value of the OCCR*h buffer register is written to OCCR*h when the counter either overflows or underflows</li> </ul>     | RW     |

### 22.4.9 General Extended Control Register (TMR4\_OCMRm)

Offset: 0x0024 (TMR4\_OCMRuh), 0x0028 (TMR4\_OCMRul), 0x002C (TMR4\_OCMRvh), 0x0030 (TMR4\_OCMRvl), 0x0034 (TMR4\_OCMRwh), 0x0038 (TMR4\_OCMRwl),

**TMR4\_OCMRuh, TMR4\_OCMRvh, TMR4\_OCMRwh Register:**

| Offset | Bit Position |    |        |    |       |    |       |   |       |   |       |   |        |   |        |   |
|--------|--------------|----|--------|----|-------|----|-------|---|-------|---|-------|---|--------|---|--------|---|
| -      | 15           | 14 | 13     | 12 | 11    | 10 | 9     | 8 | 7     | 6 | 5     | 4 | 3      | 2 | 1      | 0 |
| Reset  | 0x0000       |    |        |    |       |    |       |   |       |   |       |   |        |   |        |   |
| Name   | OPNZRH       |    | OPNPKH |    | OPZRH |    | OPUCH |   | OPPKH |   | OPDCH |   | OCFZRH |   | OCFUCH |   |
| Access | RW           |    | RW     |    | RW    |    | RW    |   | RW    |   | RW    |   | RW     |   | RW     |   |

Table 22-21 OCCR\*h Count Match List

| Condition Number | Determines whether the OCCR*h count matches or not |
|------------------|----------------------------------------------------|
| 1                | No                                                 |
| 2                | Yes                                                |

| Bit   | Name   | Function                           | Description                                                                                                                                                                                                                                                                                                                                                                                                                                               | Access |
|-------|--------|------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 15:14 | OPNZRH | Underflow condition 1 port state H | Underflow condition 1 (Simultaneously satisfied): Count underflow and OCCR*h count does not match. <ul style="list-style-type: none"> <li>0b00: When the condition is satisfied, in_op*h remains unchanged</li> <li>0b01: When the condition is satisfied, in_op*h outputs high level</li> <li>0b10: When the condition is satisfied, in_op*h outputs low level</li> <li>0b11: When the condition is satisfied, in_op*h outputs inverted state</li> </ul> | RW     |
| 13:12 | OPNPKH | Overflow condition 1 port status H | Overflow condition 1 (Simultaneously satisfied): Count overflow and OCCR*h count does not match. <ul style="list-style-type: none"> <li>0b00: When the condition is satisfied, in_op*h remains unchanged</li> <li>0b01: When the condition is satisfied, in_op*h outputs high level</li> <li>0b10: When the condition is satisfied, in_op*h outputs low level</li> <li>0b11: When the condition is satisfied, in_op*h outputs inverted state</li> </ul>   | RW     |

| Bit   | Name   | Function                                | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                           | Access |
|-------|--------|-----------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 11:10 | OPZRH  | Underflow condition 2 port state H      | Underflow condition 2 (Simultaneously satisfied): Count overflow and OCCR*h count does not match.<br><ul style="list-style-type: none"> <li>0b00: When the condition is satisfied, in_op*h remains unchanged</li> <li>0b01: When the condition is satisfied, in_op*h outputs high level</li> <li>0b10: When the condition is satisfied, in_op*h outputs low level</li> <li>0b11: When the condition is satisfied, in_op*h outputs inverted state</li> </ul>           | RW     |
| 9:8   | OPUCH  | Up counting condition 2 port status H   | Up counting condition 2 (Simultaneously satisfied): the counter counts up, and the OCCR*h count matches<br><ul style="list-style-type: none"> <li>0b00: When the condition is satisfied, in_op*h remains unchanged</li> <li>0b01: When the condition is satisfied, in_op*h outputs high level</li> <li>0b10: When the condition is satisfied, in_op*h outputs low level</li> <li>0b11: When the condition is satisfied, in_op*h outputs inverted state</li> </ul>     | RW     |
| 7:6   | OPPKH  | Overflow condition 2 port status H      | Overflow Condition 2 (Simultaneously satisfied): Count overflow and OCCR*h count does match.<br><ul style="list-style-type: none"> <li>0b00: When the condition is satisfied, in_op*h remains unchanged</li> <li>0b01: When the condition is satisfied, in_op*h outputs high level</li> <li>0b10: When the condition is satisfied, in_op*h outputs low level</li> <li>0b11: When the condition is satisfied, in_op*h outputs inverted state</li> </ul>                | RW     |
| 5:4   | OPDCH  | Down-counting condition 2 port status H | Down-counting condition 2 (Simultaneously satisfied): the counter counts down, and the OCCR*h count matches<br><ul style="list-style-type: none"> <li>0b00: When the condition is satisfied, in_op*h remains unchanged</li> <li>0b01: When the condition is satisfied, in_op*h outputs high level</li> <li>0b10: When the condition is satisfied, in_op*h outputs low level</li> <li>0b11: When the condition is satisfied, in_op*h outputs inverted state</li> </ul> | RW     |
| 3     | OCFZRH | Underflow point OCFH status H           | Condition (Simultaneously satisfied): count underflow, OCCR*h count match<br><ul style="list-style-type: none"> <li>0b0: When the condition is met, the OCSR.OCFH bit remains unchanged</li> <li>0b1: When the condition is met, the OCSR.OCFH bit is set</li> </ul>                                                                                                                                                                                                  | RW     |

| Bit | Name   | Function                     | Description                                                                                                                                                                                                                                                                  | Access |
|-----|--------|------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 2   | OCFUCH | Count up OCFH state H        | Conditions (Simultaneously satisfied): counter counts up, OCCR*h count matches <ul style="list-style-type: none"> <li>● 0b0: When the condition is met, the OCSR.OCFH bit remains unchanged</li> <li>● 0b1: When the condition is met, the OCSR.OCFH bit is set</li> </ul>   | RW     |
| 1   | OCFPKH | Overflow point OCFH status H | Condition (Simultaneously satisfied): count overflow, OCCR*h count match <ul style="list-style-type: none"> <li>● 0b0: When the condition is met, the OCSR.OCFH bit remains unchanged</li> <li>● 0b1: When the condition is met, the OCSR.OCFH bit is set</li> </ul>         | RW     |
| 0   | OCFDCH | Count up OCFH status H       | Conditions (Simultaneously satisfied): counter counts down, OCCR*h count matches <ul style="list-style-type: none"> <li>● 0b0: When the condition is met, the OCSR.OCFH bit remains unchanged</li> <li>● 0b1: When the condition is met, the OCSR.OCFH bit is set</li> </ul> | RW     |

**TMR4\_OCMRul, TMR4\_OCMRvl, TMR4\_OCMRwl Register:**

| Offset | Bit Position |    |         |    |        |    |        |    |        |    |        |    |         |    |         |    |        |    |        |    |       |    |       |   |       |   |       |   |        |   |        |   |        |  |        |  |
|--------|--------------|----|---------|----|--------|----|--------|----|--------|----|--------|----|---------|----|---------|----|--------|----|--------|----|-------|----|-------|---|-------|---|-------|---|--------|---|--------|---|--------|--|--------|--|
| -      | 31           | 30 | 29      | 28 | 27     | 26 | 25     | 24 | 23     | 22 | 21     | 20 | 19      | 18 | 17      | 16 | 15     | 14 | 13     | 12 | 11    | 10 | 9     | 8 | 7     | 6 | 5     | 4 | 3      | 2 | 1      | 0 |        |  |        |  |
| Reset  | 0x00000000   |    |         |    |        |    |        |    |        |    |        |    |         |    |         |    |        |    |        |    |       |    |       |   |       |   |       |   |        |   |        |   |        |  |        |  |
| Name   | EOPNZRL      |    | EOPNPKL |    | EOPZRL |    | EOPUCL |    | EOPPKL |    | EOPDCL |    | EOPNUCL |    | EOPNDCL |    | OPNZRL |    | OPNPKL |    | OPZRL |    | OPUCL |   | OPPKL |   | OPDCL |   | OCFZRL |   | OCFUCL |   | OCFPKL |  | OCFDCL |  |
| Access | RW           |    | RW      |    | RW     |    | RW     |    | RW     |    | RW     |    | RW      |    | RW      |    | RW     |    | RW     |    | RW    |    | RW    |   | RW    |   | RW    |   | RW     |   | RW     |   | RW     |  | RW     |  |

**Table 22-24 OCCR\*h/1 Count Match List**

| Condition Number | Determines whether the OCCR*h count matches or not | Determines whether the OCCR*l count matches or not |
|------------------|----------------------------------------------------|----------------------------------------------------|
| 1                | No                                                 | No                                                 |
| 2                | No                                                 | Yes                                                |
| 3                | Yes                                                | No                                                 |
| 4                | Yes                                                | Yes                                                |

| Bit   | Name    | Function                           | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                  | Access |
|-------|---------|------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:30 | EOPNZRL | Underflow condition 3 port state L | Underflow condition 3 (Simultaneously satisfied): Counting underflow, OCCR*l mismatch, OCCR*h match <ul style="list-style-type: none"> <li>0b00: When the condition is satisfied, in_op*l remains unchanged</li> <li>0b01: When the condition is satisfied, in_op*l outputs high level</li> <li>0b10: When the condition is satisfied, in_op*l outputs low level</li> <li>0b11: When the condition is satisfied the output of in_op*l is reversed</li> </ul> | RW     |
| 29:28 | EOPNPKL | Overflow condition 3 port state L  | Overflow condition 3 (Simultaneously satisfied): Counting overflow, OCCR*l mismatch, OCCR*h match <ul style="list-style-type: none"> <li>0b00: When the condition is satisfied, in_op*l remains unchanged</li> <li>0b01: When the condition is satisfied, in_op*l outputs high level</li> <li>0b10: When the condition is satisfied, in_op*l outputs low level</li> <li>0b11: When the condition is satisfied the output of in_op*l is reversed</li> </ul>   | RW     |

| Bit   | Name   | Function                                | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | Access |
|-------|--------|-----------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 27:26 | EOPZRL | Underflow condition 4 port status L     | Underflow condition 4 (Simultaneously satisfied): Counting underflow, OCCR*I match, OCCR*h match <ul style="list-style-type: none"> <li>0b00: When the condition is satisfied, in_op*I remains unchanged</li> <li>0b01: When the condition is satisfied, in_op*I outputs high level</li> <li>0b10: When the condition is satisfied, in_op*I outputs low level</li> <li>0b11: When the condition is satisfied the output of in_op*I is reversed</li> </ul>                      | RW     |
| 25:24 | EOPUCL | Up counting condition 4 port status L   | Up counting condition 4 (Simultaneously satisfied): counter counts up, OCCR*I count matches, OCCR*h count matches <ul style="list-style-type: none"> <li>0b00: When the condition is satisfied, in_op*I remains unchanged</li> <li>0b01: When the condition is satisfied, in_op*I outputs high level</li> <li>0b10: When the condition is satisfied, in_op*I outputs low level</li> <li>0b11: When the condition is satisfied the output of in_op*I is reversed</li> </ul>     | RW     |
| 23:22 | EOPPKL | Overflow condition 4 port state L       | Overflow condition 4 (Simultaneously satisfied): count overflow, OCCR*I count match, OCCR*h count match <ul style="list-style-type: none"> <li>0b00: When the condition is satisfied, in_op*I remains unchanged</li> <li>0b01: When the condition is satisfied, in_op*I outputs high level</li> <li>0b10: When the condition is satisfied, in_op*I outputs low level</li> <li>0b11: When the condition is satisfied the output of in_op*I is reversed</li> </ul>               | RW     |
| 21:20 | EOPDCL | Down counting condition 4 port status L | Down counting condition 4 (Simultaneously satisfied): counter counts down, OCCR*I count matches, OCCR*h count matches <ul style="list-style-type: none"> <li>0b00: When the condition is satisfied, in_op*I remains unchanged</li> <li>0b01: When the condition is satisfied, in_op*I outputs high level</li> <li>0b10: When the condition is satisfied, in_op*I outputs low level</li> <li>0b11: When the condition is satisfied the output of in_op*I is reversed</li> </ul> | RW     |



| Bit   | Name    | Function                                | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | Access |
|-------|---------|-----------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 19:18 | EOPNUCL | Up counting condition 3 port status L   | Up counting condition 3 (Simultaneously satisfied): the counter counts up, OCCR*I counts do not match, OCCR*h counts match <ul style="list-style-type: none"> <li>● 0b00: When the condition is satisfied, in_op*I remains unchanged</li> <li>● 0b01: When the condition is satisfied, in_op*I outputs high level</li> <li>● 0b10: When the condition is satisfied, in_op*I outputs low level</li> <li>● 0b11: When the condition is satisfied the output of in_op*I is reversed</li> </ul> | RW     |
| 17:16 | EOPNDCL | Down counting condition 3 port status L | Down counting condition 3 (Simultaneously satisfied): counter counts down, OCCR*I counts do not match, OCCR*h counts match <ul style="list-style-type: none"> <li>● 0b00: When the condition is satisfied, in_op*I remains unchanged</li> <li>● 0b01: When the condition is satisfied, in_op*I outputs high level</li> <li>● 0b10: When the condition is satisfied, in_op*I outputs low level</li> <li>● 0b11: When the condition is satisfied the output of in_op*I is reversed</li> </ul> | RW     |
| 15:14 | OPNZRL  | Underflow condition 1 port status L     | Underflow condition 1 (Simultaneously satisfied): Counting underflow, OCCR*I mismatch, OCCR*h match <ul style="list-style-type: none"> <li>● 0b00: When the condition is satisfied, in_op*I remains unchanged</li> <li>● 0b01: When the condition is satisfied, in_op*I outputs high level</li> <li>● 0b10: When the condition is satisfied, in_op*I outputs low level</li> <li>● 0b11: When the condition is satisfied the output of in_op*I is reversed</li> </ul>                        | RW     |
| 13:12 | OPNPKL  | Overflow condition 1 port state L       | Overflow condition 1 (Simultaneously satisfied): count overflow, OCCR*I count match, OCCR*h count mismatch <ul style="list-style-type: none"> <li>● 0b00: When the condition is satisfied, in_op*I remains unchanged</li> <li>● 0b01: When the condition is satisfied, in_op*I outputs high level</li> <li>● 0b10: When the condition is satisfied, in_op*I outputs low level</li> <li>● 0b11: When the condition is satisfied the output of in_op*I is reversed</li> </ul>                 | RW     |

| Bit   | Name   | Function                                | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | Access |
|-------|--------|-----------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 11:10 | OPZRL  | Underflow condition 2 port status L     | Underflow condition 2 (Simultaneously satisfied): Counting underflow, OCCR*I match, OCCR*h mismatch <ul style="list-style-type: none"> <li>0b00: When the condition is satisfied, in_op*I remains unchanged</li> <li>0b01: When the condition is satisfied, in_op*I outputs high level</li> <li>0b10: When the condition is satisfied, in_op*I outputs low level</li> <li>0b11: When the condition is satisfied the output of in_op*I is reversed</li> </ul>                          | RW     |
| 9:8   | OPUCL  | Up counting condition 2 port status L   | Up counting condition 2 (Simultaneously satisfied): The counter counts up, OCCR*I count matches, OCCR*h count does not match <ul style="list-style-type: none"> <li>0b00: When the condition is satisfied, in_op*I remains unchanged</li> <li>0b01: When the condition is satisfied, in_op*I outputs high level</li> <li>0b10: When the condition is satisfied, in_op*I outputs low level</li> <li>0b11: When the condition is satisfied the output of in_op*I is reversed</li> </ul> | RW     |
| 7:6   | OPPKL  | Overflow condition 2 port state L       | Up counting condition 2 (Simultaneously satisfied): Count overflow, OCCR*I count matches, OCCR*h count does not match <ul style="list-style-type: none"> <li>0b00: When the condition is satisfied, in_op*I remains unchanged</li> <li>0b01: When the condition is satisfied, in_op*I outputs high level</li> <li>0b10: When the condition is satisfied, in_op*I outputs low level</li> <li>0b11: When the condition is satisfied the output of in_op*I is reversed</li> </ul>        | RW     |
| 5:4   | OPDCL  | Down counting condition 2 port status L | Down counting condition 2 (Simultaneously satisfied): counter counts down, OCCR*I counts match, OCCR*h counts mismatch <ul style="list-style-type: none"> <li>0b00: When the condition is satisfied, in_op*I remains unchanged</li> <li>0b01: When the condition is satisfied, in_op*I outputs high level</li> <li>0b10: When the condition is satisfied, in_op*I outputs low level</li> <li>0b11: When the condition is satisfied the output of in_op*I is reversed</li> </ul>       | RW     |
| 3     | OCFZRL | Underflow point OCFL status L           | Condition (Simultaneously satisfied): count underflow, OCCR*I count match <ul style="list-style-type: none"> <li>0b0: When the condition is met, the OCSR.OCFL bit remains unchanged</li> <li>0b1: When the condition is met, the OCSR.OCFL bit is set</li> </ul>                                                                                                                                                                                                                     | RW     |

| Bit | Name   | Function                     | Description                                                                                                                                                                                                                                                              | Access |
|-----|--------|------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 2   | OCFUCL | Up counting OCFL state L     | Conditions (Simultaneously satisfied): counter counts up, OCCR*I count matches <ul style="list-style-type: none"> <li>0b0: When the condition is met, the OCSR.OCFL bit remains unchanged</li> <li>0b1: When the condition is met, the OCSR.OCFL bit is set</li> </ul>   | RW     |
| 1   | OCFPKL | Overflow point OCFL status L | Conditions (Simultaneously satisfied): count overflow, OCCR*I count match <ul style="list-style-type: none"> <li>0b0: When the condition is met, the OCSR.OCFL bit remains unchanged</li> <li>0b1: When the condition is met, the OCSR.OCFL bit is set</li> </ul>        | RW     |
| 0   | OCFDCL | Down counting OCFL status L  | Conditions (Simultaneously satisfied): counter counts down, OCCR*I count matches <ul style="list-style-type: none"> <li>0b0: When the condition is met, the OCSR.OCFL bit remains unchanged</li> <li>0b1: When the condition is met, the OCSR.OCFL bit is set</li> </ul> | RW     |

**Notice**

- When reading data from this address area, the value read is not from the buffer register, but from the OCMR register.
- TMR4\_<t>\_O\*L can be determined by OCCR\*I and the count value of the counter (independent operation mode), or OCCR\*h and the count value of the counter and OCCR\*I and the count value of the counter (linked operation mode). Write the same 12-bit value to Bit[31:20] and Bit[15:4] of register OCMR\*I, and write 0b0000 to OCMR\*I[19:16]. At this time, the in\_op\*I output will not be affected by OCCR\*h, but only by OCCR\*I. Independent operation mode: channel\*H is configured by OCCR\*h, and channel\*L is configured by OCCR\*I. If the independent operation mode does not meet the above conditions, it is the linked operation mode: channel\*L output is affected by both OCCR\*h and OCCR\*I.
- When the following two conditions are met, it is also considered to meet the conditions set by OCMR\*h.OPPKH:
  - CCSR.MODE=1 && Count overflow && OCCR\*h=0xFFFF
  - OCER\*.MCECH=1 && Count overflow && OCCR\*h≥CNTR
- When the following two conditions are met, it is also considered to meet the conditions set by OCMR\*I.OPPKL:
  - CCSR.MODE=1 && Count overflow && OCCR\*I=0xFFFF
  - OCER\*.MCECL=1 && Count overflow && OCCR\*I≥CNTR

### 22.4.10 Special Compare Reference Register (TMR4\_SCCRm)

Offset: 0x00B2 (TMR4\_SCCRuh), 0x00B6 (TMR4\_SCCRul), 0x00BA (TMR4\_SCCRvh), 0x00BE (TMR4\_SCCRvl), 0x00C2 (TMR4\_SCCRwh), 0x00C6 (TMR4\_SCCRwl),

| Offset | Bit Position |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| -      | 15           | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x0000       |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | SCCR         |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access | RW           |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit  | Name | Function                  | Description                                                                                                                                                                                                                                                                                                                                       | Access |
|------|------|---------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 15:0 | SCCR | Special Compare Reference | Special compare reference (Compare start mode) or delayed reference (Delay start mode)<br><div>  <b>Note</b><br/>           When reading data from this address area, the value read is not from the buffer register, but from the SCCR register.         </div> | RW     |

### 22.4.11 General Control Status Register (TMR4\_SCSRm)

Offset: 0x00C8 (TMR4\_SCSRuh), 0x00CC (TMR4\_SCSRul), 0x00D0 (TMR4\_SCSRvh), 0x00D4 (TMR4\_SCSRvl), 0x00D8 (TMR4\_SCSRwh), 0x00DC (TMR4\_SCSRwl)

| Offset | Bit Position |     |     |     |          |       |       |          |     |       |   |       |   |    |   |   |
|--------|--------------|-----|-----|-----|----------|-------|-------|----------|-----|-------|---|-------|---|----|---|---|
| -      | 15           | 14  | 13  | 12  | 11       | 10    | 9     | 8        | 7   | 6     | 5 | 4     | 3 | 2  | 1 | 0 |
| Reset  | 0x0000       |     |     |     |          |       |       |          |     |       |   |       |   |    |   |   |
| Name   | ZEN          | UEN | PEN | DEN | Reserved | EVTDS | EVTMS | Reserved | LMC | EVTOS |   | BUFEN |   |    |   |   |
| Access | RW           | RW  | RW  | RW  |          | RW    | RW    |          | RW  | RW    |   | RW    |   | RW |   |   |

| Bit | Name | Function                    | Description                                                                                                                                                                                                                                                                                                                                                                 | Access |
|-----|------|-----------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 15  | ZEN  | Underflow point EVT enabled | <ul style="list-style-type: none"> <li>0b0: When the count underflows, EVT has no operation</li> <li>0b1: When the count underflows: <ul style="list-style-type: none"> <li>When EVTMS=0&amp;SCCR compare match&amp;SCMR setting match, EVT starts output</li> <li>When EVTMS=1&amp;OCCR compare match&amp;SCMR setting match, EVT delay mode starts</li> </ul> </li> </ul> | RW     |
| 14  | UEN  | Up-counting EVT enabled     | <ul style="list-style-type: none"> <li>0b0: When counting up, EVT has no operation</li> <li>0b1: When counting up: <ul style="list-style-type: none"> <li>When EVTMS=0&amp;SCCR compare match&amp;SCMR setting match, EVT starts output</li> <li>When EVTMS=1&amp;OCCR compare match&amp;SCMR setting match, EVT delay mode starts</li> </ul> </li> </ul>                   | RW     |
| 13  | PEN  | Overflow point EVT enabled  | <ul style="list-style-type: none"> <li>0b0: When the count overflows, EVT has no operation</li> <li>0b1: When the count overflows: <ul style="list-style-type: none"> <li>When EVTMS=0&amp;SCCR compare match&amp;SCMR setting match, EVT starts output</li> <li>When EVTMS=1&amp;OCCR compare match&amp;SCMR setting match, EVT delay mode starts</li> </ul> </li> </ul>   | RW     |
| 12  | DEN  | Down-counting EVT enabled   | <ul style="list-style-type: none"> <li>0b0: When counting down, EVT has no operation</li> <li>0b1: When counting down: <ul style="list-style-type: none"> <li>When EVTMS=0&amp;SCCR compare match&amp;SCMR setting match, EVT starts output</li> <li>When EVTMS=1&amp;OCCR compare match&amp;SCMR setting match, EVT delay mode starts</li> </ul> </li> </ul>               | RW     |

| Bit   | Name     | Function                                 | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | Access |
|-------|----------|------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 11:10 | Reserved | Reserved                                 | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | -      |
| 9     | EVTDS    | EVT Delay object selection               | <ul style="list-style-type: none"> <li>0b0: In delayed start mode, OCCR*h is used as the delayed comparison match object</li> <li>0b1: In delayed start mode, OCCR*l is used as the delayed comparison match object</li> </ul> <div style="border: 1px solid black; padding: 5px; margin-top: 10px;">  <b>Note</b><br/>This bit is invalid when EVTMS=0.         </div>                                                                                                               | RW     |
| 8     | EVTMS    | EVT mode selection                       | <ul style="list-style-type: none"> <li>0b0: Compare start mode (triggered by the comparison result of CNTR and SCCR)</li> <li>0b1: Delay start mode (triggered after the comparison match event is delayed by SCCR)</li> </ul>                                                                                                                                                                                                                                                                                                                                         | RW     |
| 7:6   | Reserved | Reserved                                 | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | -      |
| 5     | LMC      | Periodic interval response function link | <ul style="list-style-type: none"> <li>0b0: Periodic interval response function link is disabled, and the buffer transfer of SCCR is determined by the BUFEN setting</li> <li>0b1: Periodic interval response function link is enabled, and the buffer transfer of SCCR must satisfy CVPR.PIC=0b0000 (when the count overflows) or CVPR.ZIC=0b0000 (when the count underflows) based on the BUFEN setting</li> </ul>                                                                                                                                                   | RW     |
| 4:2   | EVTOS    | EVT output selection                     | <ul style="list-style-type: none"> <li>0b000: EVT output of Special Event 0 (TMR4_&lt;t&gt;_SCM0) is enabled</li> <li>0b001: EVT output of Special Event 1 (TMR4_&lt;t&gt;_SCM1) is enabled</li> <li>0b010: EVT output of Special Event 2 (TMR4_&lt;t&gt;_SCM2) is enabled</li> <li>0b011: EVT output of Special Event 3 (TMR4_&lt;t&gt;_SCM3) is enabled</li> <li>0b100: EVT output of Special Event 4 (TMR4_&lt;t&gt;_SCM4) is enabled</li> <li>0b101: EVT output of Special Event 5 (TMR4_&lt;t&gt;_SCM5) is enabled</li> <li>Others: Setting prohibited</li> </ul> | RW     |
| 1:0   | BUFEN    | SCCR&SCMR buffer transmission            | <ul style="list-style-type: none"> <li>0b00: The values of the SCCR and SCMR buffer registers are directly written to SCCR and SCMR</li> <li>0b01: The values of the SCCR and SCMR buffer registers are written to SCCR and SCMR when the count underflows</li> <li>0b10: The values of the SCCR and SCMR buffer registers are written to SCCR and SCMR when the count overflows</li> <li>0b11: The values of the SCCR and SCMR buffer registers are written to SCCR and SCMR when the count underflows or overflows</li> </ul>                                        | RW     |

## 22.4.12 Special Extended Control Register (TMR4\_SCER)

| Offset | Bit Position |    |    |    |    |    |   |   |   |   |   |   |      |       |   |   |
|--------|--------------|----|----|----|----|----|---|---|---|---|---|---|------|-------|---|---|
| 0x00E4 | 15           | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3    | 2     | 1 | 0 |
| Reset  | 0xFF00       |    |    |    |    |    |   |   |   |   |   |   |      |       |   |   |
| Name   | Reserved     |    |    |    |    |    |   |   |   |   |   |   | PCTS | EVTRS |   |   |
| Access |              |    |    |    |    |    |   |   |   |   |   |   | RW   | RW    |   |   |

| Bit  | Name     | Function                       | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | Access |
|------|----------|--------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 15:4 | Reserved | Reserved                       | All bits read as 1, Write as 1                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | -      |
| 3    | PCTS     | Count direction output enabled | When the counter works in triangle wave mode, control the count direction output to the port <ul style="list-style-type: none"> <li>0b0: Disable the count direction signal output to the TMR4_&lt;t&gt;_PCT port</li> <li>0b1: Enable the count direction signal output to the TMR4_&lt;t&gt;_PCT port</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | RW     |
| 2:0  | EVTRS    | Special event output selection | <ul style="list-style-type: none"> <li>0b000: Disable special event output to port</li> <li>0b001: The EVT of Special Event 0 (TMR4_&lt;t&gt;_SCM0) is output to TMR4_&lt;t&gt;_ADSM port</li> <li>0b010: The EVT of Special Event1 (TMR4_&lt;t&gt;_SCM1) is output to the TMR4_&lt;t&gt;_ADSM port</li> <li>0b011: The EVT of Special Event 2 (TMR4_&lt;t&gt;_SCM2) is output to the TMR4_&lt;t&gt;_ADSM port</li> <li>0b100: The EVT of Special Event 3 (TMR4_&lt;t&gt;_SCM3) is output to the TMR4_&lt;t&gt;_ADSM port</li> <li>0b101: The EVT of Special Event 4 (TMR4_&lt;t&gt;_SCM4) is output to the TMR4_&lt;t&gt;_ADSM port</li> <li>0b110: The EVT of Special Event 5 (TMR4_&lt;t&gt;_SCM5) is output to the TMR4_&lt;t&gt;_ADSM port</li> <li>0b111: Setting prohibited</li> </ul> | RW     |

### 22.4.13 Special Mode Control Register (TMR4\_SCMRm)

Offset: 0x00CA (TMR4\_SCMRuh), 0x00CE (TMR4\_SCMRul), 0x00D2 (TMR4\_SCMRvh), 0x00D6 (TMR4\_SCMRvl), 0x00DA (TMR4\_SCMRwh), 0x00DE (TMR4\_SCMRwl)

| Offset | Bit Position |    |    |    |    |    |   |   |      |      |          |   |     |   |   |   |
|--------|--------------|----|----|----|----|----|---|---|------|------|----------|---|-----|---|---|---|
| -      | 15           | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7    | 6    | 5        | 4 | 3   | 2 | 1 | 0 |
| Reset  | 0xFF00       |    |    |    |    |    |   |   |      |      |          |   |     |   |   |   |
| Name   | Reserved     |    |    |    |    |    |   |   | MPCE | MZCE | Reserved |   | AMC |   |   |   |
| Access |              |    |    |    |    |    |   |   | RW   | RW   |          |   | RW  |   |   |   |

| Bit  | Name     | Function                                   | Description                                                                                                                                                              | Access |
|------|----------|--------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 15:8 | Reserved | Reserved                                   | All bits read as 1, Write as 1                                                                                                                                           | -      |
| 7    | MPCE     | Periodic interval response enabled         | <ul style="list-style-type: none"> <li>0b0: Disable AMC and CVPR.PIC comparison</li> <li>0b1: Enable AMC and CVPR.PIC comparison</li> </ul>                              | RW     |
| 6    | MZCE     | Periodic interval response enabled         | <ul style="list-style-type: none"> <li>0b0: Disable comparison between AMC and CVPR.ZIC</li> <li>0b1: Enable comparison between AMC and CVPR.ZIC</li> </ul>              | RW     |
| 5:4  | Reserved | Reserved                                   | -                                                                                                                                                                        | -      |
| 3:0  | AMC      | Special event output period interval value | This bit sets the period interval value of the special event output function. When AMC and CVPR.PIC or CVPR.ZIC are equal, the special event output function is disabled | RW     |



#### Note

When reading data from this address area, the value read is not from the buffer register, but from the SCMR register.



### 22.4.14 PWM Control Register (TMR4\_POCRn)

Offset: 0x0098 (TMR4\_POCRu), 0x009C(TMR4\_POCRv), 0x00A0(TMR4\_POCRw)

| Offset | Bit Position |    |    |    |    |    |   |   |      |   |       |   |          |       |   |   |
|--------|--------------|----|----|----|----|----|---|---|------|---|-------|---|----------|-------|---|---|
| -      | 15           | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7    | 6 | 5     | 4 | 3        | 2     | 1 | 0 |
| Reset  | 0xFF00       |    |    |    |    |    |   |   |      |   |       |   |          |       |   |   |
| Name   | Reserved     |    |    |    |    |    |   |   | LVLS |   | PWMMD |   | Reserved | DIVCK |   |   |
| Access |              |    |    |    |    |    |   |   | RW   |   | RW    |   |          | RW    |   |   |

| Bit  | Name     | Function                       | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | Access |
|------|----------|--------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 15:8 | Reserved | Reserved                       | All bits read as 1, Write as 1                                                                                                                                                                                                                                                                                                                                                                                                                                                      | -      |
| 7:6  | LVLS     | PWM output polarity control    | <ul style="list-style-type: none"> <li>0b00: neither TMR4_&lt;t&gt;_O*H nor TMR4_&lt;t&gt;_O*L outputs are inverted</li> <li>0b01: The outputs of TMR4_&lt;t&gt;_O*H and TMR4_&lt;t&gt;_O*L are both inverted</li> <li>0b10: The output of TMR4_&lt;t&gt;_O*H is inverted, and the output of TMR4_&lt;t&gt;_O*L is not inverted</li> <li>0b11: The output of TMR4_&lt;t&gt;_O*H is not inverted, and the output of TMR4_&lt;t&gt;_O*L is inverted</li> </ul>                        | RW     |
| 5:4  | PWMMD    | PWM output mode                | <ul style="list-style-type: none"> <li>0b00: Pass-through mode</li> <li>0b01: Dead-zone timer mode</li> <li>0b10: Dead-zone timer filter mode</li> <li>0b11: Setting prohibited</li> </ul>                                                                                                                                                                                                                                                                                          | RW     |
| 3    | Reserved | Reserved                       | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | -      |
| 2:0  | DIVCK    | Count clock division frequency | <p>This bit indicates the count clock division of the filter counter, dead zone counter and reload counter.</p> <ul style="list-style-type: none"> <li>0b000: Count clock is PCLK0</li> <li>0b001: Count clock is PCLK0/2</li> <li>0b010: Count clock is PCLK0/4</li> <li>0b011: Count clock is PCLK0/8</li> <li>0b100: Count clock is PCLK0/16</li> <li>0b101: Count clock is PCLK0/32</li> <li>0b110: Count clock is PCLK0/64</li> <li>0b111: Count clock is PCLK0/128</li> </ul> | RW     |

## 22.4.15 PWM Status Control Register (TMR4\_PSCR)

| Offset | Bit Position |    |    |    |      |    |      |    |      |    |      |    |      |    |      |    |          |    |    |    |     |    |     |   |     |   |      |   |      |   |      |   |      |  |      |  |      |  |
|--------|--------------|----|----|----|------|----|------|----|------|----|------|----|------|----|------|----|----------|----|----|----|-----|----|-----|---|-----|---|------|---|------|---|------|---|------|--|------|--|------|--|
| 0x00E0 | 31           | 30 | 29 | 28 | 27   | 26 | 25   | 24 | 23   | 22 | 21   | 20 | 19   | 18 | 17   | 16 | 15       | 14 | 13 | 12 | 11  | 10 | 9   | 8 | 7   | 6 | 5    | 4 | 3    | 2 | 1    | 0 |      |  |      |  |      |  |
| Reset  | 0x05550000   |    |    |    |      |    |      |    |      |    |      |    |      |    |      |    |          |    |    |    |     |    |     |   |     |   |      |   |      |   |      |   |      |  |      |  |      |  |
| Name   | Reserved     |    |    |    | OSWL |    | OSWH |    | OSVL |    | OSVH |    | OSUL |    | OSUH |    | Reserved |    |    |    | AOE |    | MOE |   | ODT |   | OEWL |   | OEWH |   | OEVL |   | OEVH |  | OEUL |  | OEUH |  |
| Access |              |    |    |    | RW   |    | RW   |    | RW   |    | RW   |    | RW   |    | RW   |    |          |    |    |    | RW  |    | RW  |   | RW  |   | RW   |   | RW   |   | RW   |   | RW   |  | RW   |  | RW   |  |

| Bit   | Name     | Function                           | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | Access |
|-------|----------|------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:28 | Reserved | Reserved                           | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | -      |
| 27:26 | OSWL     | TMR4_<t>_OWL output disable status | <p>Conditions: 1. EMB event occurs; 2. Software writes MOE=0; 3. MOE=1&amp;OEWL=0</p> <ul style="list-style-type: none"> <li>0b00: When one of the above conditions is met, the TMR4_&lt;t&gt;_OWL port outputs normally</li> <li>0b01: When one of the above conditions is met, the TMR4_&lt;t&gt;_OWL port output is Hi-Z</li> <li>0b10: When one of the above conditions is met, the TMR4_&lt;t&gt;_OUL port output is fixed at a low level</li> <li>0b11: When one of the above conditions is met, the TMR4_&lt;t&gt;_OWL port output is fixed at a high level</li> </ul> <p> <b>Note</b><br/>This bit can only be modified when Timer4 is stopped (CCSR.STOP=1).</p> | RW     |
| 25:24 | OSWH     | TMR4_<t>_OWH output disable status | <p>Conditions: 1. EMB event occurs; 2. Software writes MOE=0; 3. MOE=1&amp;OEWH=0</p> <ul style="list-style-type: none"> <li>0b00: When one of the above conditions is met, the TMR4_&lt;t&gt;_OWH port outputs normally</li> <li>0b01: When one of the above conditions is met, the TMR4_&lt;t&gt;_OWH port output is Hi-Z</li> <li>0b10: When one of the above conditions is met, the TMR4_&lt;t&gt;_OWH port output is fixed at a low level</li> <li>0b11: When one of the above conditions is met, the TMR4_&lt;t&gt;_OWH port output is fixed at a high level</li> </ul> <p> <b>Note</b><br/>This bit can only be modified when Timer4 is stopped (CCSR.STOP=1).</p> | RW     |
| 23:22 | OSVL     | TMR4_<t>_OVL output disable status | <p>Conditions: 1. EMB event occurs; 2. Software writes MOE=0; 3. MOE=1&amp;OEVL=0</p> <ul style="list-style-type: none"> <li>0b00: When one of the above conditions is met, the TMR4_&lt;t&gt;_OVL port outputs normally</li> <li>0b01: When one of the above conditions is met, the TMR4_&lt;t&gt;_OVL port output is Hi-Z</li> <li>0b10: When one of the above conditions is met, the TMR4_&lt;t&gt;_OVL port output is fixed at a low level</li> </ul>                                                                                                                                                                                                                                                                                                    | RW     |

| Bit   | Name     | Function                           | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | Access |
|-------|----------|------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 23:22 | OSVL     | TMR4_<t>_OVL output disable status | <ul style="list-style-type: none"> <li>0b11: When one of the above conditions is met, the TMR4_&lt;t&gt;_OVL port output is fixed at a high level</li> </ul> <div>  <b>Note</b><br/>           This bit can only be modified when Timer4 is stopped (CCSR.STOP=1).         </div>                                                                                                                                                                                                                                                                                                                                                                                                         | RW     |
| 21:20 | OSVH     | TMR4_<t>_OVH output disable status | Conditions: 1. EMB event occurs; 2. Software writes MOE=0; 3. MOE=1&OEVL=0 <ul style="list-style-type: none"> <li>0b00: When one of the above conditions is met, the TMR4_&lt;t&gt;_OVH port outputs normally</li> <li>0b01: When one of the above conditions is met, the TMR4_&lt;t&gt;_OVH port output is Hi-Z</li> <li>0b10: When one of the above conditions is met, the TMR4_&lt;t&gt;_OVH port output is fixed at a low level</li> <li>0b11: When one of the above conditions is met, the TMR4_&lt;t&gt;_OVH port output is fixed at a high level</li> </ul> <div>  <b>Note</b><br/>           This bit can only be modified when Timer4 is stopped (CCSR.STOP=1).         </div>   | RW     |
| 19:18 | OSUL     | TMR4_<t>_OUL output disable status | Conditions: 1. EMB event occurs; 2. Software writes MOE=0; 3. MOE=1&OEVL=0 <ul style="list-style-type: none"> <li>0b00: When one of the above conditions is met, the TMR4_&lt;t&gt;_OUL port outputs normally</li> <li>0b01: When one of the above conditions is met, the TMR4_&lt;t&gt;_OUL port output is Hi-Z</li> <li>0b10: When one of the above conditions is met, the TMR4_&lt;t&gt;_OUL port output is fixed at a low level</li> <li>0b11: When one of the above conditions is met, the TMR4_&lt;t&gt;_OUL port output is fixed at a high level</li> </ul> <div>  <b>Note</b><br/>           This bit can only be modified when Timer4 is stopped (CCSR.STOP=1).         </div> | RW     |
| 17:16 | OSUH     | TMR4_<t>_OUH output disable status | Conditions: 1. EMB event occurs; 2. Software writes MOE=0; 3. MOE=1&OEVL=0 <ul style="list-style-type: none"> <li>0b00: When one of the above conditions is met, the TMR4_&lt;t&gt;_OUH port outputs normally</li> <li>0b01: When one of the above conditions is met, the TMR4_&lt;t&gt;_OUH port output is Hi-Z</li> <li>0b10: When one of the above conditions is met, the TMR4_&lt;t&gt;_OUH port output is fixed at a low level</li> <li>0b11: When one of the above conditions is met, the TMR4_&lt;t&gt;_OUH port output is fixed at a high level</li> </ul> <div>  <b>Note</b><br/>           This bit can only be modified when Timer4 is stopped (CCSR.STOP=1).         </div> | RW     |
| 15:10 | Reserved | Reserved                           | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | -      |

| Bit | Name | Function                    | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | Access |
|-----|------|-----------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 9   | AOE  | Automatic output enable     | <ul style="list-style-type: none"> <li>0b0: After the EMB event is cleared, the MOE bit can only be set to 1 by software to restore normal PWM output</li> <li>0b1: After the EMB event is cleared, the MOE bit is automatically set to 1 by hardware, and the PWM output returns to normal</li> </ul> <div>  <b>Note</b><br/>           Setting EMB_STATCLR or clearing EMB_SOE can clear the EMB event.         </div>                                                                                                   | RW     |
| 8   | MOE  | Master output enable        | <ul style="list-style-type: none"> <li>0b0: TMR4_&lt;t&gt;_O*H/L port outputs the state specified by the OS*H/L bit</li> <li>0b1: TMR4_&lt;t&gt;_O*H, TMR4_&lt;t&gt;_O*L port outputs are enabled when the OE*H, OE*L bits are set to enable</li> </ul> <div>  <b>Note</b><br/>           This bit is cleared immediately by hardware after an EMB event occurs. After the EMB event is cleared, this bit can be set to 1 by software or automatically by hardware according to the setting of the AOE bit.         </div> | RW     |
| 7:6 | ODT  | Port enable bit active time | Condition: After the value of port enable bit OE*L, OE*H is changed <ul style="list-style-type: none"> <li>0b0x: When the condition is met, it takes effect immediately</li> <li>0b10: When the condition is met, it takes effect when the counter underflows next time</li> <li>0b11: When the condition is met, it takes effect when the counter overflows next time</li> </ul>                                                                                                                                                                                                                           | RW     |
| 5   | OEWL | TMR4_<t>_OWL output enabled | <ul style="list-style-type: none"> <li>0b0: TMR4_&lt;t&gt;_OWL port output state is set by the OSWL bit</li> <li>0b1: When MOE=1, TMR4_&lt;t&gt;_OWL port outputs normal PWM waveform</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                            | RW     |
| 4   | OEWH | TMR4_<t>_OWH output enabled | <ul style="list-style-type: none"> <li>0b0: TMR4_&lt;t&gt;_OWH port output state is set by the OSWH bit</li> <li>0b1: When MOE=1, TMR4_&lt;t&gt;_OWH port outputs normal PWM waveform</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                            | RW     |
| 3   | OEVL | TMR4_<t>_OVL output enabled | <ul style="list-style-type: none"> <li>0b0: TMR4_&lt;t&gt;_OVL port output state is set by the OSVL bit</li> <li>0b1: When MOE=1, TMR4_&lt;t&gt;_OVL port outputs normal PWM waveform</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                            | RW     |
| 2   | OEVL | TMR4_<t>_OVH output enabled | <ul style="list-style-type: none"> <li>0b0: TMR4_&lt;t&gt;_OVH port output state is set by the OSVH bit</li> <li>0b1: When MOE=1, TMR4_&lt;t&gt;_OVH port outputs normal PWM waveform</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                            | RW     |

| Bit | Name | Function                    | Description                                                                                                                                                                                      | Access |
|-----|------|-----------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 1   | OEUL | TMR4_<t>_OUL output enabled | <ul style="list-style-type: none"> <li>0b0: TMR4_&lt;t&gt;_OUL port output state is set by the OSUL bit</li> <li>0b1: When MOE=1, TMR4_&lt;t&gt;_OUL port outputs normal PWM waveform</li> </ul> | RW     |
| 0   | OEUH | TMR4_<t>_OUH output enabled | <ul style="list-style-type: none"> <li>0b0: TMR4_&lt;t&gt;_OUH port output state is set by the OSUH bit</li> <li>0b1: When MOE=1, TMR4_&lt;t&gt;_OUH port outputs normal PWM waveform</li> </ul> | RW     |

The following table shows the relationship between the PWM port output status and the register setting value:

| MOE bits | OE*L/H bits | OS*L/H bits | TMR4_<t>_O*L/H output enable |
|----------|-------------|-------------|------------------------------|
| 0b1      | 0b0         | 0b00        | Normal output (Timer4 drive) |
|          |             | 0b01        | Hi-Z                         |
|          |             | 0b10        | Low Level                    |
|          |             | 0b11        | High Level                   |
|          | 0b1         | 0bXX        | Normal output (Timer4 drive) |
| 0b0      | 0bX         | 0b00        | Normal output (Timer4 drive) |
|          |             | 0b01        | Hi-Z                         |
|          |             | 0b10        | Low Level                    |
|          |             | 0b11        | High Level                   |

### 22.4.16 PWM Filter Control Register (TMR4\_PFSRn)

Offset: 0x0082 (TMR4\_PFSRu), 0x008A(TMR4\_PFSRv), 0x0092(TMR4\_PFSRw)

| Offset | Bit Position |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| -      | 15           | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x0000       |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | PFSR         |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access | RW           |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit  | Name | Function             | Description                                                                                                                                                                                                                                                                                                                                                                                                                               | Access |
|------|------|----------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 15:0 | PFSR | Filter Initial Value | Filter count initial value<br><br><div>  <b>Note</b><br/>           When the dead timer filter mode is not selected in the PWM waveform output mode, the 16-bit filter counter is used as a 16-bit reload counter, which can generate interrupt output periodically. This function is independent of the PWM waveform generator function.         </div> | RW     |

### 22.4.17 PWM Dead Zone Control Register A (TMR4\_PDARn)

Offset: 0x0084 (TMR4\_PDARu), 0x008C(TMR4\_PDARv), 0x0094(TMR4\_PDARw)

| Offset | Bit Position |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| -      | 15           | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x0000       |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | PDAR         |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access | RW           |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit  | Name | Function                | Description                   | Access |
|------|------|-------------------------|-------------------------------|--------|
| 15:0 | PDAR | Dead Zone Initial Value | Dead zone count initial value | RW     |

### 22.4.18 PWM Dead Zone Control Register B (TMR4\_PDBRn)

Offset: 0x0086 (TMR4\_PDBRu), 0x008E(TMR4\_PDBRv), 0x0096(TMR4\_PDBRw)

| Offset | Bit Position |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| -      | 15           | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x0000       |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | PDBR         |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access | RW           |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit  | Name | Function                | Description                   | Access |
|------|------|-------------------------|-------------------------------|--------|
| 15:0 | PDBR | Dead Zone Initial Value | Dead zone count initial value | RW     |

## 22.4.19 Reload Control Status Register (TMR4\_RCSR)

| Offset | Bit Position |      |       |       |      |      |       |       |      |      |       |       |          |       |       |       |
|--------|--------------|------|-------|-------|------|------|-------|-------|------|------|-------|-------|----------|-------|-------|-------|
| 0x00A4 | 15           | 14   | 13    | 12    | 11   | 10   | 9     | 8     | 7    | 6    | 5     | 4     | 3        | 2     | 1     | 0     |
| Reset  | 0x0000       |      |       |       |      |      |       |       |      |      |       |       |          |       |       |       |
| Name   | RTSW         | RTEW | RTICW | RTIFW | RTSV | RTEV | RTICV | RTIFV | RTSU | RTEU | RTICU | RTIFU | Reserved | RTIDW | RTIDV | RTIDU |
| Access | RW           | RW   | RW    | RO    | RW   | RW   | RW    | RO    | RW   | RW   | RW    | RO    |          | RW    | RW    | RW    |

| Bit | Name  | Function                   | Description                                                                                                                                                                                     | Access |
|-----|-------|----------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 15  | RTSW  | Reload counter stop W      | Writing 1 stops reloading counter W and clears RTIFW, write 0 has no effect; read always returns 0.                                                                                             | RW     |
| 14  | RTEW  | Reload counter start W     | Write 1 to start reloading counter W, write 0 has no effect.<br>Read:<br><ul style="list-style-type: none"> <li>0b0: Reload counter W stopped</li> <li>0b1: Reload counter W started</li> </ul> | RW     |
| 13  | RTICW | Clear count match status W | Write 1 to clear RTIFW flag bit, write 0 has no effect; read always returns 0.                                                                                                                  | RW     |
| 12  | RTIFW | Count match status W       | <ul style="list-style-type: none"> <li>0b0: The reload counter count value matches 0</li> <li>0b1: The reload counter count value matches 0</li> </ul>                                          | RO     |
| 11  | RTSV  | Reload counter stop V      | Writing 1 stops reloading counter V and clears RTIFV, write 0 has no effect; read always returns 0.                                                                                             | RW     |
| 10  | RTEV  | Reload counter start V     | Write 1 to start reloading counter V, write 0 has no effect.<br>Read:<br><ul style="list-style-type: none"> <li>0b0: Reload counter V stopped</li> <li>0b1: Reload counter V started</li> </ul> | RW     |
| 9   | RTICV | Clear count match status V | Write 1 to clear RTIFV flag bit, write 0 has no effect; read always returns 0.                                                                                                                  | RW     |
| 8   | RTIFV | Clear count match status V | <ul style="list-style-type: none"> <li>0b0: The reload counter count value matches 0.</li> <li>0b1: The reload counter count value matches 0.</li> </ul>                                        | RO     |
| 7   | RTSU  | Reload counter stop U      | Writing 1 stops reloading counter U and clears RTIFU, write 0 has no effect; read always returns 0.                                                                                             | RW     |
| 6   | RTEU  | Reload counter start U     | Write 1 to start reloading counter U, write 0 has no effect.<br>Read:<br><ul style="list-style-type: none"> <li>0b0: Reload counter U stopped</li> <li>0b1: Reload counter U started</li> </ul> | RW     |



| Bit | Name            | Function                   | Description                                                                                                                                                                                                             | Access |
|-----|-----------------|----------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 5   | RTICU           | Clear count match status U | Write 1 to clear RTIFU flag bit, write 0 has no effect; read always returns 0.                                                                                                                                          | RW     |
| 4   | RTIFU           | Clear count match status U | <ul style="list-style-type: none"> <li>0b0: The reload counter count value matches 0.</li> <li>0b1: The reload counter count value matches 0.</li> </ul>                                                                | RO     |
| 3   | <i>Reserved</i> | <i>Reserved</i>            | -                                                                                                                                                                                                                       | -      |
| 2   | RTIDW           | Reload interrupt mask W    | <ul style="list-style-type: none"> <li>0b0: Reload interrupt W output is enabled when the reload function is active.</li> <li>0b1: Reload interrupt W output is disabled when the reload function is active.</li> </ul> | RW     |
| 1   | RTIDV           | Reload interrupt mask V    | <ul style="list-style-type: none"> <li>0b0: Reload interrupt V output is enabled when the reload function is active.</li> <li>0b1: Reload interrupt V output is disabled when the reload function is active.</li> </ul> | RW     |
| 0   | RTIDU           | Reload interrupt mask U    | <ul style="list-style-type: none"> <li>0b0: Reload interrupt U output is enabled when the reload function is active.</li> <li>0b1: Reload interrupt U output is disabled when the reload function is active.</li> </ul> | RW     |

## 23 Emergency Brake Module (EMB)

### 23.1 Introduction

The emergency brake module is a function module that generates control events output to the timer when certain conditions are met to stop the timer or change the PWM signal output to the external motor. The following factors are used to generate control events:

- External port input level change
- Level of PWM output port occurs in phase (same high or same low)
- Voltage comparator comparison results
- System error occurred
- Write register software control

### 23.2 Functional Description

#### 23.2.1 EMB Channel Function Block Diagram

EMB is used to output a control event signal to the timer module with PWM function (Timer4, Timer6) when certain conditions are met, notifying the timer module to turn off the current PWM output. The EMB module has 7 groups, among which Group0-Group3 are used to control Timer6 and are set and selected by the Timer6 register; Group4-Group6 are used to control Timer4 and correspond to the 3 units of Timer4 respectively.

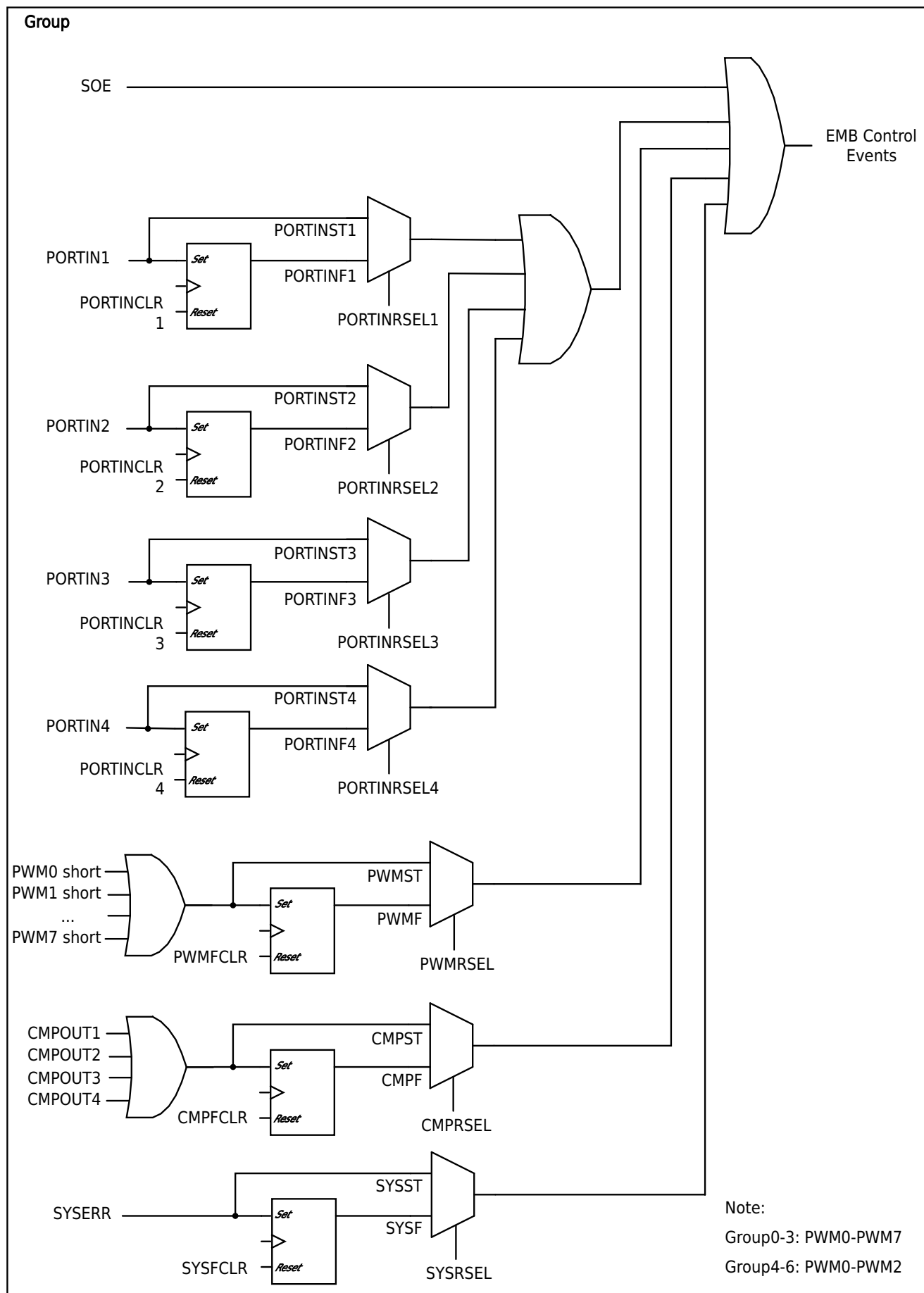


Figure 23-1 EMB Channel Function Block Diagram

23.2.2 EMB Structure Block Diagram

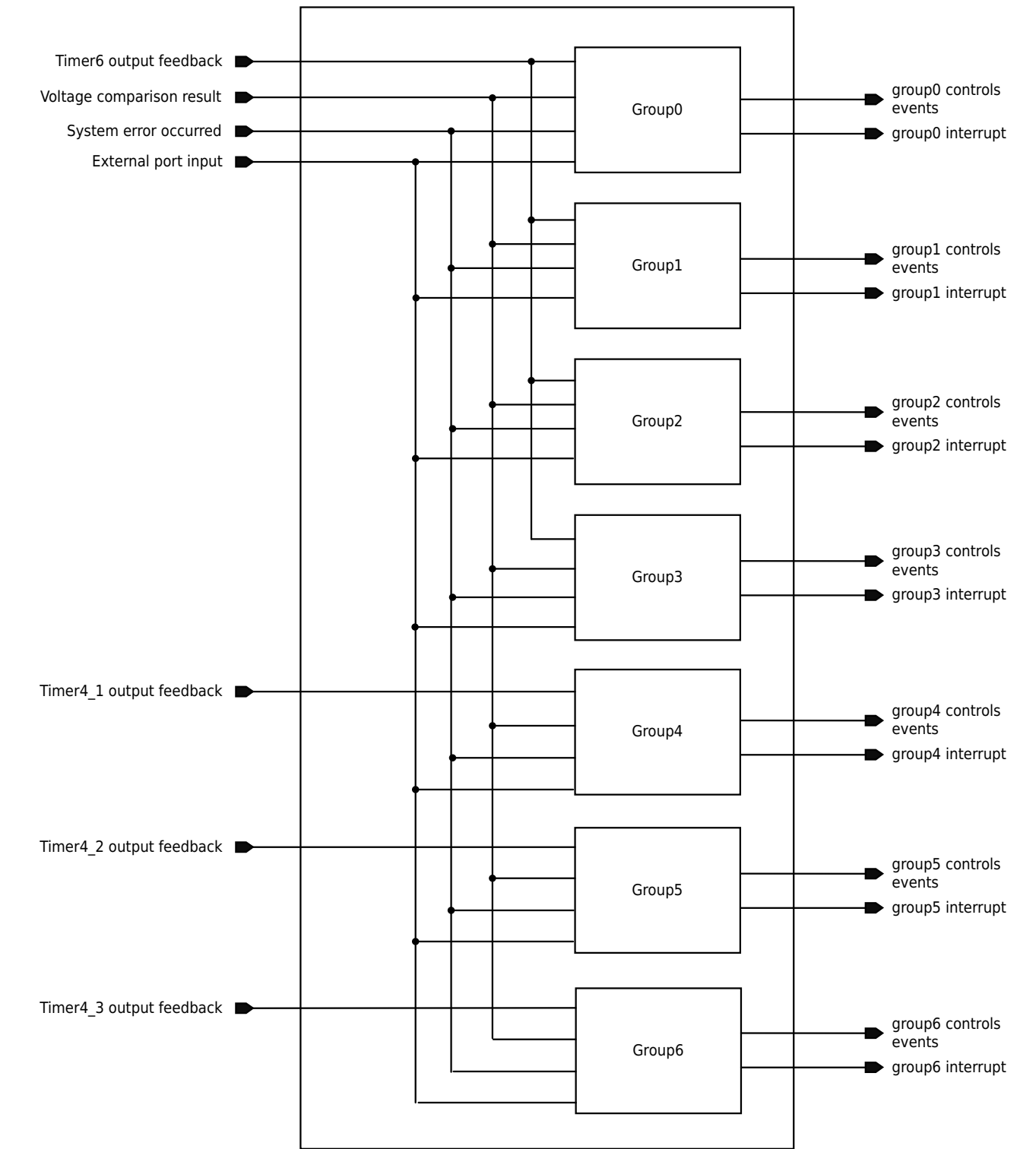


Figure 23-2 EMB Structure Block Diagram

23.2.3 Control PWM Signal Output When External Port Input Level Changes

EMB has 4 external ports to control PWM signal output when the input level changes. Each group can independently set 1 or more of the 4 external ports to be enable, and the port allocation is shown in the following table.

Table 23-1 EMB Port Allocation

| Port    | PIN Name              |
|---------|-----------------------|
| EMB_IN1 | PA11, PB2, PC5, PD7   |
| EMB_IN2 | PA6, PB11, PB12, PE15 |
| EMB_IN3 | PA7, PC2, PD9         |
| EMB_IN4 | PB15, PD4, PH2, PI4   |

When using external port input level changes to control PWM signal output, first set the enable bit, that is, the EMB control register EMB\_CTL1.PORTINENy (y=1-4) to be enable, and at the same time, set the control signal to be generated when the port level is high (INVSEL=0) or low (INVSEL=1) through EMB\_CTL1.INVSELy (y=1-4). Enable the filter function and set the filter clock as needed through EMB\_CTL2.NFENy (y=1-4) and EMB\_CTL2.NFSELy (y=1-4).

After the filter samples the input signal according to the filter clock, it filters using a method of three consistent comparisons. That is, when the filter clock samples the same level on the port three times, this level is considered as an active level and sent to the block. Levels that are consistent less than three times are considered as external interference and filtered out, not sent to the block.

When an active level that meets the conditions is generated on the port, the port input status EMB\_STAT.PORTINSTy (y=1-4) of the EMB status register is set, and at the same time, the port input control flag bit EMB\_STAT.PORTINFy (y=1-4) is set. An interrupt will be generated when EMB\_INTEN.PORTINEN=1.

The EMB control event selects registers EMB\_RLSSEL.PORTINRSELy (y=1-4) through releasing EMB. EMB\_STAT.PORTINSTy (y=1-4) or EMB\_STAT.PORTINFy (y=1-4) outputs the selection to Timer6 and Timer4. After receiving the control event, Timer6 and Timer4 can set the output port to high level, low level or high impedance state according to the register settings.

When releasing the control, if EMB\_RLSSEL.PORTINRSELy (y=1-4) =1, when the port input level becomes inactive, EMB\_STAT.PORTINSTy (y=1-4) is automatically cleared to zero, and the control event is immediately released. If EMB\_RLSSEL.PORTINRSELy (y=1-4) =0, after the port input level becomes inactive, the EMB status reset register EMB\_STATCLR.PORTINFCLR needs to be written to clear EMB\_STAT.PORTINF to 0 and release the control event.

#### 23.2.4 Stop PWM Signal Output When PWM Output Port Level Occurs In-phase (same high or same low)

When using, first set the enable bit, that is, the EMB control register EMB\_CTL1.PWMSENy (Group=0-3, y=0-7 or Group=4-6, y=0-2) to be enable, and at the same time, select the active level to be monitored through the EMB control register EMB\_CTL2.PWMLVy (Group=0-3, y=0-7 or Group=4-6, y=0-2).

After the settings are completed, EMB monitors the complementary PWM output signals of Timer6 and Timer4. When the output signals are the same high or the same low, the PWM output status EMB\_STAT.PWMST of the EMB status register is set, and at the same time, the PWM output in-phase control flag bit EMB\_STAT.PWMSF is set. An interrupt will be generated when EMB\_INTEN.PWMEN=1.

The EMB control event selects register EMB\_RLSSEL.PWMRSEL through EMB release. The selection is outputted to Timer6 and Timer4 via EMB\_STAT.PWMST or EMB\_STAT.PWMSF. After receiving the control event, Timer6 and Timer4 can set the output port to high level, low level or high impedance state according to the register settings.

When releasing control, if EMB\_RLSSEL.PWMRSEL=1, when the PWM output same high or same low state is released, EMB\_STAT.PWMST is automatically cleared to zero, and the control event is immediately released. If EMB\_RLSSEL.PWMST=0, after the PWM output same high or same low state is released, the EMB status reset register EMB\_STATCLR.PWMSFCLR needs to be written to clear EMB\_STAT.PWMSF to 0 and release the control event.

Group0-3 can be used to monitor the complementary PWM output signals of Timer6, Group4 is used to monitor the complementary PWM output signals of Timer4\_1, Group5 is used to monitor the complementary PWM output signals of Timer4\_2, Group6 is used to monitor the complementary PWM output signals of Timer4\_3.

Table 23-2 EMB Group Mapping Table

| Port Name                     | Function                                    | Corresponding Group                  | EMB_CTL1 Control Bit | EMB_CTL2 Control Bit |
|-------------------------------|---------------------------------------------|--------------------------------------|----------------------|----------------------|
| TMR6_<t>_PWMA<br>(<t> is 1-8) | Complementary PWM output signal of Timer6   | Group0<br>Group1<br>Group2<br>Group3 | PWMSEN7-<br>PWMSEN0  | PWMLV7-PWMLV0        |
| TMR6_<t>_PWMB<br>(<t> is 1-8) |                                             |                                      |                      |                      |
| TMR4_1_OUH                    | Complementary PWM output signal of Timer4_1 | Group4                               | PWMSEN2              | PWMLV2               |
| TMR4_1_OUL                    |                                             |                                      | PWMSEN1              | PWMLV1               |
| TMR4_1_OVH                    |                                             |                                      |                      |                      |
| TMR4_1_OVL                    |                                             |                                      | PWMSEN0              | PWMLV0               |
| TMR4_1_OWH                    |                                             |                                      |                      |                      |
| TMR4_1_OWL                    |                                             |                                      |                      |                      |
| TMR4_2_OUH                    | Complementary PWM output signal of Timer4_2 | Group5                               | PWMSEN2              | PWMLV2               |
| TMR4_2_OUL                    |                                             |                                      | PWMSEN1              | PWMLV1               |
| TMR4_2_OVH                    |                                             |                                      |                      |                      |
| TMR4_2_OVL                    |                                             |                                      | PWMSEN0              | PWMLV0               |
| TMR4_2_OWH                    |                                             |                                      |                      |                      |
| TMR4_2_OWL                    |                                             |                                      |                      |                      |
| TMR4_3_OUH                    | Complementary PWM output signal of Timer4_3 | Group6                               | PWMSEN2              | PWMLV2               |
| TMR4_3_OUL                    |                                             |                                      | PWMSEN1              | PWMLV1               |
| TMR4_3_OVH                    |                                             |                                      |                      |                      |
| TMR4_3_OVL                    |                                             |                                      | PWMSEN0              | PWMLV0               |
| TMR4_3_OWH                    |                                             |                                      |                      |                      |
| TMR4_3_OWL                    |                                             |                                      |                      |                      |

### 23.2.5 Stop PWM Signal Output According to Voltage Comparator Compare Result

Each group of EMB can send control event signals to Timer6 and Timer4 according to the 4 groups of comparison results of the voltage comparator. For the settings of the comparator output results, refer to the relevant descriptions in the "Voltage Comparator (CMP) " chapter.

When using, first set the enable bit, that is, the EMB control register EMB\_CTL1.CMPENy (y=1-4) to be enable. When the comparator comparison result flag is set, the comparator status EMB\_STAT.CMPST of the EMB status register is set, and at the same time, the EMB comparator control flag bit EMB\_STAT.CMPF is set. An interrupt is generated when the interrupt enable register EMB\_INTEN.CMPINT-EN=1.

The control event of EMB selects register EMB\_RLSSEL.CMPRSEL through the EMB release. The selection is output to Timer6 and Timer4 via EMB\_STAT.CMPST or EMB\_STAT.CMPF. After receiving the control event signal, Timer6 and Timer4 can set the output port to high level, low level or high impedance state according to the register settings.

When releasing control, if EMB\_RLSSEL.CMPRSEL=1, when the comparator result becomes inactive, EMB\_STAT.CMPST is automatically cleared to zero, and the control event is immediately released. If EMB\_RLSSEL.CMPRSEL=0, after the comparator result becomes inactive, the EMB status reset register EMB\_STATCLR.CMPFCLR needs to be written to clear EMB\_STAT.CMPF to 0 and release the control event.

### 23.2.6 Stop PWM Signal Output when System Error Occurs

Each Group of EMB can send notification signals to Timer6 and Timer4 when a system error occurs. System error signals come from PVD detection, Cortex-M4 lock signal, SRAM ECC check error, external oscillator stop, Flash ECC check error and FCM frequency measurement abnormality.

When using, first set the enable bit, that is, the EMB control register EMB\_CTL1.SYSEN to be enable, and set the EMB\_CTL1.PVDEN, LOCKUPEN, SRAMECCERREN, OSCSTPEN, FLASHECCERREN, FCMERREN bits to enable the system error sources to be detected. When the selected system error source flag is set, the system error status EMB\_STAT.SYSST of the EMB status register is set, and at the same time, the EMB system error control flag EMB\_STAT.SYSF is set. An interrupt will also be generated when EMB\_INTEN.SYSINTEN=1.

The EMB control events are routed to Timer6 and Timer4 through EMB\_RLSSEL.SYSRSEL, based on the status of EMB\_STAT.SYSST or EMB\_STAT.SYSF. After receiving the control event signal, Timer6 and Timer4 can set the output port to high level, low level or high impedance state according to the register settings.

When releasing control, if EMB\_RLSSEL.SYSRSEL=1, when the system error signal becomes inactive, EMB\_STAT.SYSST is automatically cleared to zero, and the control event is immediately released. If EMB\_RLSSEL.SYSRSEL=0, after the system error signal becomes inactive, the EMB status reset register EMB\_STATCLR.SYSFCLR needs to be written to clear EMB\_STAT.SYSF to 0 and release the control event.

### 23.2.7 Write Register Software Control PWM Signal Output

The EMB software output enable control register (EMB\_SOE) allows users to send control signals to Timer6 and Timer4 directly through software setting and resetting. When the software controls the PWM output, no interrupt request is generated.

## 23.3 Register Description

### 23.3.1 Register List

- EMB Group0 Base Address: 0x40017C00
- EMB Group1 Base Address: 0x40017C20
- EMB Group2 Base Address: 0x40017C40
- EMB Group3 Base Address: 0x40017C60
- EMB Group4 Base Address: 0x40017C80
- EMB Group5 Base Address: 0x40017CA0
- EMB Group6 Base Address: 0x40017CC0



#### Note

EMB contains 7 Groups, namely Group0-6.

Table 23-3 Emergency Brake Module (EMB) Register Offset

| Offset | Register                    | Description                                            | Width |
|--------|-----------------------------|--------------------------------------------------------|-------|
| 0x00   | <a href="#">EMB_CTL1</a>    | EMB Control Register 1 (Group 0-3)                     | 32    |
| 0x00   | <a href="#">EMB_CTL1</a>    | EMB Control Register 1 (Group4-6)                      | 32    |
| 0x04   | <a href="#">EMB_CTL2</a>    | EMB Control Register 2 (Group 0-3)                     | 32    |
| 0x04   | <a href="#">EMB_CTL2</a>    | EMB Control Register 2 (Group4-6)                      | 32    |
| 0x08   | <a href="#">EMB_SOE</a>     | EMB Software Output Enable Control Register            | 32    |
| 0x0C   | <a href="#">EMB_STAT</a>    | EMB Status Register                                    | 32    |
| 0x10   | <a href="#">EMB_STATCLR</a> | EMB Status Reset Register                              | 32    |
| 0x14   | <a href="#">EMB_INTEN</a>   | EMB Interrupt License Register                         | 32    |
| 0x18   | <a href="#">EMB_RLSSEL</a>  | EMB Control PWM Output Release Mode Selection Register | 32    |



### 23.3.2 EMB Control Register 1 (EMB\_CTL1) (Group0-3)

This register is a one-time write register, that is, it can only be written once after reset.

| Offset | Bit Position |          |          |              |          |               |         |         |         |         |          |           |           |           |           |          |          |         |         |         |         |         |         |         |         |       |        |        |        |        |    |    |    |
|--------|--------------|----------|----------|--------------|----------|---------------|---------|---------|---------|---------|----------|-----------|-----------|-----------|-----------|----------|----------|---------|---------|---------|---------|---------|---------|---------|---------|-------|--------|--------|--------|--------|----|----|----|
| 0x00   | 31           | 30       | 29       | 28           | 27       | 26            | 25      | 24      | 23      | 22      | 21       | 20        | 19        | 18        | 17        | 16       | 15       | 14      | 13      | 12      | 11      | 10      | 9       | 8       | 7       | 6     | 5      | 4      | 3      | 2      | 1  | 0  |    |
| Reset  | 0x00000000   |          |          |              |          |               |         |         |         |         |          |           |           |           |           |          |          |         |         |         |         |         |         |         |         |       |        |        |        |        |    |    |    |
| Name   | PVDEN        | LOCKUPEN | Reserved | SRAMECCERREN | OSCSTPEN | FLASHECCERREN | INVSEL4 | INVSEL3 | INVSEL2 | INVSEL1 | Reserved | PORTINEN4 | PORTINEN3 | PORTINEN2 | PORTINEN1 | FCMERREN | Reserved | PWMSEN7 | PWMSEN6 | PWMSEN5 | PWMSEN4 | PWMSEN3 | PWMSEN2 | PWMSEN1 | PWMSEN0 | SYSEN | CMPEN4 | CMPEN3 | CMPEN2 | CMPEN1 |    |    |    |
| Access | RW           | RW       |          | RW           | RW       | RW            | RW      | RW      | RW      | RW      |          | RW        | RW        | RW        | RW        | RW       |          | RW      | RW      | RW      | RW      | RW      | RW      | RW      | RW      | RW    | RW     | RW     | RW     | RW     | RW | RW | RW |

| Bit | Name          | Function                              | Description                                                                                                                                                                                   | Access |
|-----|---------------|---------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31  | PVDEN         | PVD detection control enabled         | <ul style="list-style-type: none"> <li>0b0: PVD detection result output control disabled</li> <li>0b1: PVD detection result output control enabled</li> </ul>                                 | RW     |
| 30  | LOCKUPEN      | Cortex-M4 Lockup control enabled      | <ul style="list-style-type: none"> <li>0b0: Output control is disabled when lockup occurs</li> <li>0b1: Output control is enabled when lockup occurs</li> </ul>                               | RW     |
| 29  | Reserved      | Reserved                              | -                                                                                                                                                                                             | -      |
| 28  | SRAMECCERREN  | SRAM ECC check error control enabled  | <ul style="list-style-type: none"> <li>0b0: Output control is disabled when SRAM ECC check error occurs</li> <li>0b1: Output control is enabled when SRAM ECC check error occurs</li> </ul>   | RW     |
| 27  | OSCSTPEN      | Oscillator stop control enabled       | <ul style="list-style-type: none"> <li>0b0: Output control is disabled when oscillator stop occurs</li> <li>0b1: Output control is enabled when oscillator stop occurs</li> </ul>             | RW     |
| 26  | FLASHECCERREN | FLASH ECC check error control enabled | <ul style="list-style-type: none"> <li>0b0: Output control is disabled when FLASH ECC check error occurs</li> <li>0b1: Output control is enabled when FLASH ECC check error occurs</li> </ul> | RW     |
| 25  | INVSEL4       | Port input 4 active level selection   | <ul style="list-style-type: none"> <li>0b0: High level enabled</li> <li>0b1: Low level enabled</li> </ul>                                                                                     | RW     |
| 24  | INVSEL3       | Port input 3 active level selection   | <ul style="list-style-type: none"> <li>0b0: High level enabled</li> <li>0b1: Low level enabled</li> </ul>                                                                                     | RW     |
| 23  | INVSEL2       | Port input 2 active level selection   | <ul style="list-style-type: none"> <li>0b0: High level enabled</li> <li>0b1: Low level enabled</li> </ul>                                                                                     | RW     |

| Bit   | Name      | Function                                              | Description                                                                                                                                                                                                     | Access |
|-------|-----------|-------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 22    | INVSEL1   | Port input 1 active level selection                   | <ul style="list-style-type: none"> <li>0b0: High level enabled</li> <li>0b1: Low level enabled</li> </ul>                                                                                                       | RW     |
| 21:20 | Reserved  | Reserved                                              | -                                                                                                                                                                                                               | -      |
| 19    | PORTINEN4 | Port input 4 control enabled                          | <ul style="list-style-type: none"> <li>0b0: Port input control disabled</li> <li>0b1: Port input control enabled</li> </ul>                                                                                     | RW     |
| 18    | PORTINEN3 | Port input 3 control enabled                          | <ul style="list-style-type: none"> <li>0b0: Port input control disabled</li> <li>0b1: Port input control enabled</li> </ul>                                                                                     | RW     |
| 17    | PORTINEN2 | Port input 2 control enabled                          | <ul style="list-style-type: none"> <li>0b0: Port input control disabled</li> <li>0b1: Port input control enabled</li> </ul>                                                                                     | RW     |
| 16    | PORTINEN1 | Port input 1 control enabled                          | <ul style="list-style-type: none"> <li>0b0: Port input control disabled</li> <li>0b1: Port input control enabled</li> </ul>                                                                                     | RW     |
| 15    | FCMERREN  | FCM frequency measurement abnormality control enabled | <ul style="list-style-type: none"> <li>0b0: Output control is disabled when FCM frequency measurement is abnormal</li> <li>0b1: Output control is enabled when FCM frequency measurement is abnormal</li> </ul> | RW     |
| 14:13 | Reserved  | Reserved                                              | -                                                                                                                                                                                                               | -      |
| 12    | PWMSEN7   | TMR6_8_PWMA/B short circuit output control enabled    | <ul style="list-style-type: none"> <li>0b0: Output control is disabled when short circuit occurs</li> <li>0b1: Output control is enabled when short circuit occurs</li> </ul>                                   | RW     |
| 11    | PWMSEN6   | TMR6_7_PWMA/B short circuit output control enabled    | <ul style="list-style-type: none"> <li>0b0: Output control is disabled when short circuit occurs</li> <li>0b1: Output control is enabled when short circuit occurs</li> </ul>                                   | RW     |
| 10    | PWMSEN5   | TMR6_6_PWMA/B short circuit output control enabled    | <ul style="list-style-type: none"> <li>0b0: Output control is disabled when short circuit occurs</li> <li>0b1: Output control is enabled when short circuit occurs</li> </ul>                                   | RW     |
| 9     | PWMSEN4   | TMR6_5_PWMA/B short circuit output control enabled    | <ul style="list-style-type: none"> <li>0b0: Output control is disabled when short circuit occurs</li> <li>0b1: Output control is enabled when short circuit occurs</li> </ul>                                   | RW     |
| 8     | PWMSEN3   | TMR6_4_PWMA/B short circuit output control enabled    | <ul style="list-style-type: none"> <li>0b0: Output control is disabled when short circuit occurs</li> <li>0b1: Output control is enabled when short circuit occurs</li> </ul>                                   | RW     |
| 7     | PWMSEN2   | TMR6_3_PWMA/B short circuit output control enabled    | <ul style="list-style-type: none"> <li>0b0: Output control is disabled when short circuit occurs</li> <li>0b1: Output control is enabled when short circuit occurs</li> </ul>                                   | RW     |

| Bit | Name    | Function                                                  | Description                                                                                                                                                                                   | Access |
|-----|---------|-----------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 6   | PWMSEN1 | TMR6_2_PWMA/B short circuit output control enabled        | <ul style="list-style-type: none"> <li>0b0: Output control is disabled when short circuit occurs</li> <li>0b1: Output control is enabled when short circuit occurs</li> </ul>                 | RW     |
| 5   | PWMSEN0 | TMR6_1_PWMA/B short circuit output control enabled        | <ul style="list-style-type: none"> <li>0b0: Output control is disabled when short circuit occurs</li> <li>0b1: Output control is enabled when short circuit occurs</li> </ul>                 | RW     |
| 4   | SYSEN   | System error control enable                               | <ul style="list-style-type: none"> <li>0b0: Output control is disabled when system error occurs</li> <li>0b1: Output control is enabled when system error occurs</li> </ul>                   | RW     |
| 3   | CMPEN4  | CMP4 voltage comparator comparison result control enabled | <ul style="list-style-type: none"> <li>0b0: Voltage comparator comparison result output control disabled</li> <li>0b1: Voltage comparator comparison result output control enabled</li> </ul> | RW     |
| 2   | CMPEN3  | CMP3 voltage comparator comparison result control enabled | <ul style="list-style-type: none"> <li>0b0: Voltage comparator comparison result output control disabled</li> <li>0b1: Voltage comparator comparison result output control enabled</li> </ul> | RW     |
| 1   | CMPEN2  | CMP2 voltage comparator comparison result control enabled | <ul style="list-style-type: none"> <li>0b0: Voltage comparator comparison result output control disabled</li> <li>0b1: Voltage comparator comparison result output control enabled</li> </ul> | RW     |
| 0   | CMPEN1  | CMP1 voltage comparator comparison result control enabled | <ul style="list-style-type: none"> <li>0b0: Voltage comparator comparison result output control disabled</li> <li>0b1: Voltage comparator comparison result output control enabled</li> </ul> | RW     |

### 23.3.3 EMB Control Register 1 (EMB\_CTL1) (Group4-6)

This register is a one-time write register, that is, it can only be written once after reset.

| Offset | Bit Position |          |          |              |          |               |         |         |         |         |          |           |           |           |           |          |          |    |    |    |    |    |   |         |         |         |       |        |        |        |        |    |    |    |
|--------|--------------|----------|----------|--------------|----------|---------------|---------|---------|---------|---------|----------|-----------|-----------|-----------|-----------|----------|----------|----|----|----|----|----|---|---------|---------|---------|-------|--------|--------|--------|--------|----|----|----|
| 0x00   | 31           | 30       | 29       | 28           | 27       | 26            | 25      | 24      | 23      | 22      | 21       | 20        | 19        | 18        | 17        | 16       | 15       | 14 | 13 | 12 | 11 | 10 | 9 | 8       | 7       | 6       | 5     | 4      | 3      | 2      | 1      | 0  |    |    |
| Reset  | 0x00000000   |          |          |              |          |               |         |         |         |         |          |           |           |           |           |          |          |    |    |    |    |    |   |         |         |         |       |        |        |        |        |    |    |    |
| Name   | PVDEN        | LOCKUPEN | Reserved | SRAMECCERREN | OSCSTPEN | FLASHECCERREN | INVSEL4 | INVSEL3 | INVSEL2 | INVSEL1 | Reserved | PORTINEN4 | PORTINEN3 | PORTINEN2 | PORTINEN1 | FCMERREN | Reserved |    |    |    |    |    |   | PWMSEN2 | PWMSEN1 | PWMSEN0 | SYSEN | CMPEN4 | CMPEN3 | CMPEN2 | CMPEN1 |    |    |    |
| Access | RW           | RW       |          | RW           | RW       | RW            | RW      | RW      | RW      | RW      |          | RW        | RW        | RW        | RW        | RW       |          |    |    |    |    |    |   |         | RW      | RW      | RW    | RW     | RW     | RW     | RW     | RW | RW | RW |

| Bit | Name          | Function                              | Description                                                                                                                                                                                   | Access |
|-----|---------------|---------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31  | PVDEN         | PVD detection control enable          | <ul style="list-style-type: none"> <li>0b0: PVD detection result output control disabled</li> <li>0b1: PVD detection result output control enabled</li> </ul>                                 | RW     |
| 30  | LOCKUPEN      | Cortex-M4 lockup control enable       | <ul style="list-style-type: none"> <li>0b0: Output control is disabled when lockup occurs</li> <li>0b1: Output control is enabled when lockup occurs</li> </ul>                               | RW     |
| 29  | Reserved      | Reserved                              | -                                                                                                                                                                                             | -      |
| 28  | SRAMECCERREN  | SRAM ECC check error control enabled  | <ul style="list-style-type: none"> <li>0b0: Output control is disabled when SRAM ECC check error occurs</li> <li>0b1: Output control is enabled when SRAM ECC check error occurs</li> </ul>   | RW     |
| 27  | OSCSTPEN      | Oscillator stop control enabled       | <ul style="list-style-type: none"> <li>0b0: Output control is disabled when oscillator stop occurs</li> <li>0b1: Output control is enabled when oscillator stop occurs</li> </ul>             | RW     |
| 26  | FLASHECCERREN | FLASH ECC check error control enabled | <ul style="list-style-type: none"> <li>0b0: Output control is disabled when FLASH ECC check error occurs</li> <li>0b1: Output control is enabled when FLASH ECC check error occurs</li> </ul> | RW     |
| 25  | INVSEL4       | Port input 4 active level selection   | <ul style="list-style-type: none"> <li>0b0: High level enabled</li> <li>0b1: Low level enabled</li> </ul>                                                                                     | RW     |
| 24  | INVSEL3       | Port input 3 active level selection   | <ul style="list-style-type: none"> <li>0b0: High level enabled</li> <li>0b1: Low level enabled</li> </ul>                                                                                     | RW     |
| 23  | INVSEL2       | Port Input 2 active level selection   | <ul style="list-style-type: none"> <li>0b0: High level enabled</li> <li>0b1: Low level enabled</li> </ul>                                                                                     | RW     |

| Bit   | Name      | Function                                                  | Description                                                                                                                                                                                                                        | Access |
|-------|-----------|-----------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 22    | INVSEL1   | Port input 1 active level selection                       | <ul style="list-style-type: none"> <li>0b0: High level enabled</li> <li>0b1: Low level enabled</li> </ul>                                                                                                                          | RW     |
| 21:20 | Reserved  | Reserved                                                  | -                                                                                                                                                                                                                                  | -      |
| 19    | PORTINEN4 | Port input 4 control enabled                              | <ul style="list-style-type: none"> <li>0b0: Port input control disabled</li> <li>0b1: Port input control enabled</li> </ul>                                                                                                        | RW     |
| 18    | PORTINEN3 | Port Input 3 control enabled                              | <ul style="list-style-type: none"> <li>0b0: Port input control disabled</li> <li>0b1: Port input control enabled</li> </ul>                                                                                                        | RW     |
| 17    | PORTINEN2 | Port Input 2 control enabled                              | <ul style="list-style-type: none"> <li>0b0: Port input control disabled</li> <li>0b1: Port input control enabled</li> </ul>                                                                                                        | RW     |
| 16    | PORTINEN1 | Port Input 1 control enabled                              | <ul style="list-style-type: none"> <li>0b0: Port input control disabled</li> <li>0b1: Port input control enabled</li> </ul>                                                                                                        | RW     |
| 15    | FCMERREN  | FCM frequency measurement abnormality control enabled     | <ul style="list-style-type: none"> <li>0b0: The output control is not enabled when the FCM frequency measurement is abnormal</li> <li>0b1: The output control is enabled when the FCM frequency measurement is abnormal</li> </ul> | RW     |
| 14:8  | Reserved  | Reserved                                                  | -                                                                                                                                                                                                                                  | -      |
| 7     | PWMSEN2   | TMR4_<t>_OUH/L short circuit output control enabled       | <ul style="list-style-type: none"> <li>0b0: Output control is disabled when short circuit occurs</li> <li>0b1: Output control is enabled when short circuit occurs</li> </ul>                                                      | RW     |
| 6     | PWMSEN1   | TMR4_<t>_OVH/L short circuit output control enabled       | <ul style="list-style-type: none"> <li>0b0: Output control is disabled when short circuit occurs</li> <li>0b1: Output control is enabled when short circuit occurs</li> </ul>                                                      | RW     |
| 5     | PWMSEN0   | TMR4_<t>_OWH/L short circuit output control enabled       | <ul style="list-style-type: none"> <li>0b0: Output control is disabled when short circuit occurs</li> <li>0b1: Output control is enabled when short circuit occurs</li> </ul>                                                      | RW     |
| 4     | SYSEN     | System error control enabled                              | <ul style="list-style-type: none"> <li>0b0: Output control is disabled when system error occurs</li> <li>0b1: Output control is enabled when system error occurs</li> </ul>                                                        | RW     |
| 3     | CMPEN4    | CMP4 voltage comparator comparison result control enabled | <ul style="list-style-type: none"> <li>0b0: Voltage comparator comparison result output control disabled</li> <li>0b1: Voltage comparator comparison result output control enabled</li> </ul>                                      | RW     |
| 2     | CMPEN3    | CMP3 voltage comparator comparison result control enabled | <ul style="list-style-type: none"> <li>0b0: Voltage comparator comparison result output control disabled</li> <li>0b1: Voltage comparator comparison result output control enabled</li> </ul>                                      | RW     |

| Bit | Name   | Function                                                  | Description                                                                                                                                                                                   | Access |
|-----|--------|-----------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 1   | CMPEN2 | CMP2 voltage comparator comparison result control enabled | <ul style="list-style-type: none"> <li>0b0: Voltage comparator comparison result output control disabled</li> <li>0b1: Voltage comparator comparison result output control enabled</li> </ul> | RW     |
| 0   | CMPEN1 | CMP1 voltage comparator comparison result control enabled | <ul style="list-style-type: none"> <li>0b0: Voltage comparator comparison result output control disabled</li> <li>0b1: Voltage comparator comparison result output control enabled</li> </ul> | RW     |

**Note**

- Group4: <t> is 1
- Group5: <t> is 2
- Group6: <t> is 3

### 23.3.4 EMB Control Register 2 (EMB\_CTL2) (Group0-3)

This register is a one-time write register, that is, it can only be written once after reset.

| Offset | Bit Position |    |       |        |    |       |        |    |       |        |    |       |        |    |          |    |    |    |    |    |    |    |        |        |        |        |        |        |        |        |    |   |
|--------|--------------|----|-------|--------|----|-------|--------|----|-------|--------|----|-------|--------|----|----------|----|----|----|----|----|----|----|--------|--------|--------|--------|--------|--------|--------|--------|----|---|
| 0x04   | 31           | 30 | 29    | 28     | 27 | 26    | 25     | 24 | 23    | 22     | 21 | 20    | 19     | 18 | 17       | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9      | 8      | 7      | 6      | 5      | 4      | 3      | 2      | 1  | 0 |
| Reset  | 0x00000000   |    |       |        |    |       |        |    |       |        |    |       |        |    |          |    |    |    |    |    |    |    |        |        |        |        |        |        |        |        |    |   |
| Name   | Reserved     |    | NFEN4 | NFSEL4 |    | NFEN3 | NFSEL3 |    | NFEN2 | NFSEL2 |    | NFEN1 | NFSEL1 |    | Reserved |    |    |    |    |    |    |    | PWMLV7 | PWMLV6 | PWMLV5 | PWMLV4 | PWMLV3 | PWMLV2 | PWMLV1 | PWMLV0 |    |   |
|        |              |    |       |        |    |       |        |    |       |        |    |       |        |    |          |    |    |    |    |    |    |    |        |        |        |        |        |        |        |        |    |   |
| Access |              |    | RW    | RW     |    | RW    | RW     |    | RW    | RW     |    | RW    | RW     |    |          |    |    |    |    |    |    |    | RW     | RW     | RW     | RW     | RW     | RW     | RW     | RW     | RW |   |

| Bit   | Name     | Function                                    | Description                                                                                                                                                                                                                          | Access |
|-------|----------|---------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:28 | Reserved | Reserved                                    | -                                                                                                                                                                                                                                    | -      |
| 27    | NFEN4    | Port Input 4 Digital Filter Enabled         | <ul style="list-style-type: none"> <li>0b0: Filter disabled</li> <li>0b1: Filter enabled</li> </ul>                                                                                                                                  | RW     |
| 26:25 | NFSEL4   | Port Input 4 Digital Filter Clock Selection | <ul style="list-style-type: none"> <li>0b00: Use PCLK1 filtering</li> <li>0b01: Filtering with 8-division of PCLK1</li> <li>0b10: Filtering with 32-division of PCLK1</li> <li>0b11: Filtering with 128-division of PCLK1</li> </ul> | RW     |
| 24    | NFEN3    | Port Input 3 Digital Filter Enabled         | <ul style="list-style-type: none"> <li>0b0: Filter disabled</li> <li>0b1: Filter enabled</li> </ul>                                                                                                                                  | RW     |
| 23:22 | NFSEL3   | Port Input 3 Digital Filter Clock Selection | <ul style="list-style-type: none"> <li>0b00: Use PCLK1 filtering</li> <li>0b01: Filtering with 8-division of PCLK1</li> <li>0b10: Filtering with 32-division of PCLK1</li> <li>0b11: Filtering with 128-division of PCLK1</li> </ul> | RW     |
| 21    | NFEN2    | Port Input 2 Digital Filter Enabled         | <ul style="list-style-type: none"> <li>0b0: Filter disabled</li> <li>0b1: Filter enabled</li> </ul>                                                                                                                                  | RW     |
| 20:19 | NFSEL2   | Port Input 2 Digital Filter Clock Selection | <ul style="list-style-type: none"> <li>0b00: Use PCLK1 filtering</li> <li>0b01: Filtering with 8-division of PCLK1</li> <li>0b10: Filtering with 32-division of PCLK1</li> <li>0b11: Filtering with 128-division of PCLK1</li> </ul> | RW     |
| 18    | NFEN1    | Port Input 1 Digital Filter Enabled         | <ul style="list-style-type: none"> <li>0b0: Filter disabled</li> <li>0b1: Filter enabled</li> </ul>                                                                                                                                  | RW     |
| 17:16 | NFSEL1   | Port Input 1 Digital Filter Clock Selection | <ul style="list-style-type: none"> <li>0b00: Use PCLK1 filtering</li> <li>0b01: Filtering with 8-division of PCLK1</li> <li>0b10: Filtering with 32-division of PCLK1</li> <li>0b11: Filtering with 128-division of PCLK1</li> </ul> | RW     |
| 15:8  | Reserved | Reserved                                    | -                                                                                                                                                                                                                                    | -      |
| 7     | PWMLV7   | TMR6_8_PWMA/B Output Enable Level Selection | <ul style="list-style-type: none"> <li>0b0: Low level is the enable level</li> <li>0b1: High level is the enable level</li> </ul>                                                                                                    | RW     |

| Bit | Name   | Function                                          | Description                                                                                                                           | Access |
|-----|--------|---------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------|--------|
| 6   | PWMLV6 | TMR6_7_PWMA/B<br>Output Enable<br>Level Selection | <ul style="list-style-type: none"> <li>● 0b0: Low level is the enable level</li> <li>● 0b1: High level is the enable level</li> </ul> | RW     |
| 5   | PWMLV5 | TMR6_6_PWMA/B<br>Output Enable<br>Level Selection | <ul style="list-style-type: none"> <li>● 0b0: Low level is the enable level</li> <li>● 0b1: High level is the enable level</li> </ul> | RW     |
| 4   | PWMLV4 | TMR6_5_PWMA/B<br>Output Enable<br>Level Selection | <ul style="list-style-type: none"> <li>● 0b0: Low level is the enable level</li> <li>● 0b1: High level is the enable level</li> </ul> | RW     |
| 3   | PWMLV3 | TMR6_4_PWMA/B<br>Output Enable<br>Level Selection | <ul style="list-style-type: none"> <li>● 0b0: Low level is the enable level</li> <li>● 0b1: High level is the enable level</li> </ul> | RW     |
| 2   | PWMLV2 | TMR6_3_PWMA/B<br>Output Enable<br>Level Selection | <ul style="list-style-type: none"> <li>● 0b0: Low level is the enable level</li> <li>● 0b1: High level is the enable level</li> </ul> | RW     |
| 1   | PWMLV1 | TMR6_2_PWMA/B<br>Output Enable<br>Level Selection | <ul style="list-style-type: none"> <li>● 0b0: Low level is the enable level</li> <li>● 0b1: High level is the enable level</li> </ul> | RW     |
| 0   | PWMLV0 | TMR6_1_PWMA/B<br>Output Enable<br>Level Selection | <ul style="list-style-type: none"> <li>● 0b0: Low level is the enable level</li> <li>● 0b1: High level is the enable level</li> </ul> | RW     |



### 23.3.5 EMB Control Register 2 (EMB\_CTL2) (Group4-6)

This register is a one-time write register, that is, it can only be written once after reset.

| Offset | Bit Position |    |       |        |    |       |        |    |       |        |    |       |        |    |          |    |    |    |    |    |    |    |        |        |        |   |   |   |   |   |   |   |
|--------|--------------|----|-------|--------|----|-------|--------|----|-------|--------|----|-------|--------|----|----------|----|----|----|----|----|----|----|--------|--------|--------|---|---|---|---|---|---|---|
| 0x04   | 31           | 30 | 29    | 28     | 27 | 26    | 25     | 24 | 23    | 22     | 21 | 20    | 19     | 18 | 17       | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9      | 8      | 7      | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |       |        |    |       |        |    |       |        |    |       |        |    |          |    |    |    |    |    |    |    |        |        |        |   |   |   |   |   |   |   |
| Name   | Reserved     |    | NFEN4 | NFSEL4 |    | NFEN3 | NFSEL3 |    | NFEN2 | NFSEL2 |    | NFEN1 | NFSEL1 |    | Reserved |    |    |    |    |    |    |    | PWMLV2 | PWMLV1 | PWMLV0 |   |   |   |   |   |   |   |
| Access |              |    | RW    | RW     |    | RW    | RW     |    | RW    | RW     |    | RW    | RW     |    |          |    |    |    |    |    |    |    | RW     | RW     | RW     |   |   |   |   |   |   |   |

| Bit   | Name     | Function                                     | Description                                                                                                                                                                                                                          | Access |
|-------|----------|----------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:28 | Reserved | Reserved                                     | -                                                                                                                                                                                                                                    | -      |
| 27    | NFEN4    | Port input 4 digital filter enable           | <ul style="list-style-type: none"> <li>0b0: Filter disabled</li> <li>0b1: Filter enabled</li> </ul>                                                                                                                                  | RW     |
| 26:25 | NFSEL4   | Port input 4 digital filter clock selection  | <ul style="list-style-type: none"> <li>0b00: Use PCLK1 filtering</li> <li>0b01: Filtering with 8-division of PCLK1</li> <li>0b10: Filtering with 32-division of PCLK1</li> <li>0b11: Filtering with 128-division of PCLK1</li> </ul> | RW     |
| 24    | NFEN3    | Port input 3 digital filter enabled          | <ul style="list-style-type: none"> <li>0b0: Filter disabled</li> <li>0b1: Filter enabled</li> </ul>                                                                                                                                  | RW     |
| 23:22 | NFSEL3   | Port input 3 digital filter clock selection  | <ul style="list-style-type: none"> <li>0b00: Use PCLK1 filtering</li> <li>0b01: Filtering with 8-division of PCLK1</li> <li>0b10: Filtering with 32-division of PCLK1</li> <li>0b11: Filtering with 128-division of PCLK1</li> </ul> | RW     |
| 21    | NFEN2    | Port input 2 digital filter enabled          | <ul style="list-style-type: none"> <li>0b0: Filter disabled</li> <li>0b1: Filter enabled</li> </ul>                                                                                                                                  | RW     |
| 20:19 | NFSEL2   | Port input 2 digital filter clock selection  | <ul style="list-style-type: none"> <li>0b00: Use PCLK1 filtering</li> <li>0b01: Filtering with 8-division of PCLK1</li> <li>0b10: Filtering with 32-division of PCLK1</li> <li>0b11: Filtering with 128-division of PCLK1</li> </ul> | RW     |
| 18    | NFEN1    | Port input 1 digital filter enabled          | <ul style="list-style-type: none"> <li>0b0: Filter disabled</li> <li>0b1: Filter enabled</li> </ul>                                                                                                                                  | RW     |
| 17:16 | NFSEL1   | Port input 1 digital filter clock selection  | <ul style="list-style-type: none"> <li>0b00: Use PCLK1 filtering</li> <li>0b01: Filtering with 8-division of PCLK1</li> <li>0b10: Filtering with 32-division of PCLK1</li> <li>0b11: Filtering with 128-division of PCLK1</li> </ul> | RW     |
| 15:3  | Reserved | Reserved                                     | -                                                                                                                                                                                                                                    | -      |
| 2     | PWMLV2   | TMR4_<t>_OUH/L output enable level selection | <ul style="list-style-type: none"> <li>0b0: Low level is the enable level</li> <li>0b1: High level is the enable level</li> </ul>                                                                                                    | RW     |

| Bit | Name   | Function                                     | Description                                                                                                                       | Access |
|-----|--------|----------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------|--------|
| 1   | PWMLV1 | TMR4_<t>_OVH/L output enable level selection | <ul style="list-style-type: none"> <li>0b0: Low level is the enable level</li> <li>0b1: High level is the enable level</li> </ul> | RW     |
| 0   | PWMLV0 | TMR4_<t>_OWH/L output enable level selection | <ul style="list-style-type: none"> <li>0b0: Low level is the enable level</li> <li>0b1: High level is the enable level</li> </ul> | RW     |

**Note**

- Group4: <t> is 1
- Group5: <t> is 2
- Group6: <t> is 3

### 23.3.6 EMB Software Output Enable Control Register (EMB\_SOE)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |     |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|-----|
| 0x08   | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |     |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |     |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   | SOE |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   | RW  |

| Bit  | Name     | Function                | Description                                                                                             | Access |
|------|----------|-------------------------|---------------------------------------------------------------------------------------------------------|--------|
| 31:1 | Reserved | Reserved                | -                                                                                                       | -      |
| 0    | SOE      | Software Control Output | <ul style="list-style-type: none"> <li>0b0: Normal PWM output</li> <li>0b1: PWM output stops</li> </ul> | RW     |

## 23.3.7 EMB Status Register (EMB\_STAT)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |           |           |           |           |          |          |          |          |          |       |       |       |          |      |      |       |          |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|-----------|-----------|-----------|-----------|----------|----------|----------|----------|----------|-------|-------|-------|----------|------|------|-------|----------|---|
| 0x0C   | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17        | 16        | 15        | 14        | 13       | 12       | 11       | 10       | 9        | 8     | 7     | 6     | 5        | 4    | 3    | 2     | 1        | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |           |           |           |           |          |          |          |          |          |       |       |       |          |      |      |       |          |   |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    | PORTINST4 | PORTINST3 | PORTINST2 | PORTINST1 | Reserved | PORTINF4 | PORTINF3 | PORTINF2 | PORTINF1 | SYSST | CMPST | PWMST | Reserved | SYSF | CMPF | PWMSF | Reserved |   |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    | RO        | RO        | RO        | RO        |          | RO       | RO       | RO       | RO       | RO    | RO    | RO    |          | RO   | RO   | RO    |          |   |

| Bit   | Name      | Function                      | Description                                                                                                                                                           | Access |
|-------|-----------|-------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:18 | Reserved  | Reserved                      | -                                                                                                                                                                     | -      |
| 17    | PORTINST4 | Port Input 4 Control Status   | <ul style="list-style-type: none"> <li>0b0: Port input control is in the disable level state</li> <li>0b1: Port input control is in the enable level state</li> </ul> | RO     |
| 16    | PORTINST3 | Port Input 3 Control Status   | <ul style="list-style-type: none"> <li>0b0: Port input control is in the disable level state</li> <li>0b1: Port input control is in the enable level state</li> </ul> | RO     |
| 15    | PORTINST2 | Port Input 2 Control Status   | <ul style="list-style-type: none"> <li>0b0: Port input control is in the disable level state</li> <li>0b1: Port input control is in the enable level state</li> </ul> | RO     |
| 14    | PORTINST1 | Port Input 1 Control Status   | <ul style="list-style-type: none"> <li>0b0: Port input control is in the disable level state</li> <li>0b1: Port input control is in the enable level state</li> </ul> | RO     |
| 13:12 | Reserved  | Reserved                      | -                                                                                                                                                                     | -      |
| 11    | PORTINF4  | Port Input 4 Control Flag Bit | Set when EMB_STAT.PORTINST4=1.<br>Clear 0 when EMB_STATCLR.PORTINFCLR4 writes 1                                                                                       | RO     |
| 10    | PORTINF3  | Port Input 3 Control Flag Bit | Set when EMB_STAT.PORTINST3=1.<br>Clear 0 when EMB_STATCLR.PORTINFCLR3 writes 1                                                                                       | RO     |
| 9     | PORTINF2  | Port Input 2 Control Flag Bit | Set when EMB_STAT.PORTINST2=1.<br>Clear 0 when EMB_STATCLR.PORTINFCLR2 writes 1                                                                                       | RO     |
| 8     | PORTINF1  | Port Input 1 Control Flag Bit | Set when EMB_STAT.PORTINST1 is 1<br>Clear 0 when EMB_STATCLR.PORTINFCLR1 writes 1                                                                                     | RO     |
| 7     | SYSST     | System Error Status           | <ul style="list-style-type: none"> <li>0b0: No system error</li> <li>0b1: System error occurs</li> </ul>                                                              | RO     |

| Bit | Name            | Function                                    | Description                                                                                                                                                                                    | Access |
|-----|-----------------|---------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 6   | CMPST           | Voltage Com-<br>parator Status              | <ul style="list-style-type: none"> <li>● 0b0: The voltage comparator comparison result flag bit is not set</li> <li>● 0b1: The voltage comparator comparison result flag bit is set</li> </ul> | RO     |
| 5   | PWMST           | PWM Output<br>Status                        | <ul style="list-style-type: none"> <li>● 0b0: No PWM output in-phase occurs</li> <li>● 0b1: PWM output in-phase occurs</li> </ul>                                                              | RO     |
| 4   | <i>Reserved</i> | <i>Reserved</i>                             | -                                                                                                                                                                                              | -      |
| 3   | SYSF            | System Error<br>Flag Bit                    | Set when EMB_STAT.SYSST=1.<br>Clear 0 when EMB_STATCLR.SYSFCLR writes 1                                                                                                                        | RO     |
| 2   | CMPF            | Voltage Com-<br>parator Control<br>Flag Bit | Set when EMB_STAT.CMPST is 1<br>Clear 0 when EMB_STATCLR.CMPFCLR writes 1                                                                                                                      | RO     |
| 1   | PWMSF           | PWM Output In-<br>phase Control<br>Flag Bit | Set when EMB_STAT.PWMST=1.<br>Clear 0 when EMB_STATCLR.PWMSCLR writes 1                                                                                                                        | RO     |
| 0   | <i>Reserved</i> | <i>Reserved</i>                             | -                                                                                                                                                                                              | -      |

## 23.3.8 EMB Status Reset Register (EMB\_STATCLR)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |             |             |             |             |          |   |   |   |         |         |          |          |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-------------|-------------|-------------|-------------|----------|---|---|---|---------|---------|----------|----------|
| 0x10   | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11          | 10          | 9           | 8           | 7        | 6 | 5 | 4 | 3       | 2       | 1        | 0        |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |             |             |             |             |          |   |   |   |         |         |          |          |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | PORTINFCLR4 | PORTINFCLR3 | PORTINFCLR2 | PORTINFCLR1 | Reserved |   |   |   | SYSFCLR | CMPFCLR | PWMSFCLR | Reserved |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | WO          | WO          | WO          | WO          |          |   |   |   | WO      | WO      | WO       |          |

| Bit   | Name        | Function                | Description                                                                                | Access |
|-------|-------------|-------------------------|--------------------------------------------------------------------------------------------|--------|
| 31:12 | Reserved    | Reserved                | -                                                                                          | -      |
| 11    | PORTINFCLR4 | Reset EMB_STAT.PORTINF4 | Write 1 to clear EMB_STAT.PORTINF4 to 0 (when EMB_STAT.PORTINST4=0), write 0 has no effect | WO     |
| 10    | PORTINFCLR3 | Reset EMB_STAT.PORTINF3 | Write 1 to clear EMB_STAT.PORTINF3 to 0 (when EMB_STAT.PORTINST3=0), write 0 has no effect | WO     |
| 9     | PORTINFCLR2 | Reset EMB_STAT.PORTINF2 | Write 1 to clear EMB_STAT.PORTINF2 to 0 (when EMB_STAT.PORTINST2=0), write 0 has no effect | WO     |
| 8     | PORTINFCLR1 | Reset EMB_STAT.PORTINF1 | Write 1 to clear EMB_STAT.PORTINF1 to 0 (when EMB_STAT.PORTINST1=0), write 0 has no effect | WO     |
| 7:4   | Reserved    | Reserved                | -                                                                                          | -      |
| 3     | SYSFCLR     | Reset EMB_STAT.SYSF     | Write 1 to clear EMB_STAT.SYSF to 0 (when EMB_STAT.SYSST=0), write 0 has no effect         | WO     |
| 2     | CMPFCLR     | Reset EMB_STAT.CMPF     | Write 1 to clear EMB_STAT.CMPF to 0 (when EMB_STAT.CMPST=0), write 0 has no effect         | WO     |
| 1     | PWMSFCLR    | Reset EMB_STAT.PWMSF    | Write 1 to clear EMB_STAT.PWMSF to 0 (when EMB_STAT.PWMST=0), write 0 has no effect        | WO     |
| 0     | Reserved    | Reserved                | -                                                                                          | -      |

### 23.3.9 EMB Interrupt License Register (EMB\_INTEN)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |        |      |        |      |        |      |        |      |          |   |   |   |          |        |           |          |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|--------|------|--------|------|--------|------|--------|------|----------|---|---|---|----------|--------|-----------|----------|
| 0x14   | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11     | 10   | 9      | 8    | 7      | 6    | 5      | 4    | 3        | 2 | 1 | 0 |          |        |           |          |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |        |      |        |      |        |      |        |      |          |   |   |   |          |        |           |          |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | PORTIN | TEN4 | PORTIN | TEN3 | PORTIN | TEN2 | PORTIN | TEN1 | Reserved |   |   |   | SYSINTEN | CMPTEN | PWMSINTEN | Reserved |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RW     | RW   | RW     | RW   |        |      |        |      |          |   |   |   | RW       | RW     | RW        |          |

| Bit   | Name         | Function                                                       | Description                                                                                                                                                                                                    | Access |
|-------|--------------|----------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:12 | Reserved     | Reserved                                                       | -                                                                                                                                                                                                              | -      |
| 11    | PORTININTEN4 | Port input 4 control interrupt enabled                         | <ul style="list-style-type: none"> <li>0b0: Port input control does not generate interrupt</li> <li>0b1: Port input control generates interrupt</li> </ul>                                                     | RW     |
| 10    | PORTININTEN3 | Port input 3 control interrupt enabled                         | <ul style="list-style-type: none"> <li>0b0: Port input control does not generate interrupt</li> <li>0b1: Port input control generates interrupt</li> </ul>                                                     | RW     |
| 9     | PORTININTEN2 | Port input 2 control interrupt enabled                         | <ul style="list-style-type: none"> <li>0b0: Port input control does not generate interrupt</li> <li>0b1: Port input control generates interrupt</li> </ul>                                                     | RW     |
| 8     | PORTININTEN1 | Port input 1 control interrupt enabled                         | <ul style="list-style-type: none"> <li>0b0: Port input control does not generate interrupt</li> <li>0b1: Port input control generates interrupt</li> </ul>                                                     | RW     |
| 7:4   | Reserved     | Reserved                                                       | -                                                                                                                                                                                                              | -      |
| 3     | SYSINTEN     | System error control interrupt enabled                         | <ul style="list-style-type: none"> <li>0b0: System error control does not generate interrupt</li> <li>0b1: System error control generates interrupt</li> </ul>                                                 | RW     |
| 2     | CMPINTEN     | Voltage comparator comparison result control interrupt enabled | <ul style="list-style-type: none"> <li>0b0: Voltage comparator comparison result control does not generate interrupt</li> <li>0b1: Voltage comparator comparison result control generates interrupt</li> </ul> | RW     |
| 1     | PWMSINTEN    | PWM output in-phase control interrupt enabled                  | <ul style="list-style-type: none"> <li>0b0: PWM output in-phase control does not generate interrupt</li> <li>0b1: PWM output in-phase control generates interrupt</li> </ul>                                   | RW     |
| 0     | Reserved     | Reserved                                                       | -                                                                                                                                                                                                              | -      |

## 23.3.10 EMB Control PWM Output Release Mode Selection Register (EMB\_RLSSEL)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |             |    |             |    |             |   |             |   |          |   |   |   |         |    |        |  |         |  |          |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-------------|----|-------------|----|-------------|---|-------------|---|----------|---|---|---|---------|----|--------|--|---------|--|----------|
| 0x18   | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11          | 10 | 9           | 8  | 7           | 6 | 5           | 4 | 3        | 2 | 1 | 0 |         |    |        |  |         |  |          |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |             |    |             |    |             |   |             |   |          |   |   |   |         |    |        |  |         |  |          |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | PORTINRSEL4 |    | PORTINRSEL3 |    | PORTINRSEL2 |   | PORTINRSEL1 |   | Reserved |   |   |   | SYSRSEL |    | CMRSEL |  | PWMRSEL |  | Reserved |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RW          | RW | RW          | RW |             |   |             |   |          |   |   |   | RW      | RW | RW     |  |         |  |          |

| Bit   | Name        | Function                                                       | Description                                                                                                                                                                    | Access |
|-------|-------------|----------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:12 | Reserved    | Reserved                                                       | -                                                                                                                                                                              | -      |
| 11    | PORTINRSEL4 | Port input 4 control release selection                         | <ul style="list-style-type: none"> <li>0b0: Release PWM output control when EMB_STAT.PORTINF4=0.</li> <li>0b1: Release PWM output control when EMB_STAT.PORTINST4=0</li> </ul> | RW     |
| 10    | PORTINRSEL3 | Port input 3 control release selection                         | <ul style="list-style-type: none"> <li>0b0: Release PWM output control when EMB_STAT.PORTINF3=0</li> <li>0b1: Release PWM output control when EMB_STAT.PORTINST3=0</li> </ul>  | RW     |
| 9     | PORTINRSEL2 | Port input 2 control release selection                         | <ul style="list-style-type: none"> <li>0b0: Release PWM output control when EMB_STAT.PORTINF2=0</li> <li>0b1: Release PWM output control when EMB_STAT.PORTINST2=0</li> </ul>  | RW     |
| 8     | PORTINRSEL1 | Port input 1 control release selection                         | <ul style="list-style-type: none"> <li>0b0: Release PWM output control when EMB_STAT.PORTINF1=0</li> <li>0b1: Release PWM output control when EMB_STAT.PORTINST1=0</li> </ul>  | RW     |
| 7:4   | Reserved    | Reserved                                                       | -                                                                                                                                                                              | -      |
| 3     | SYSRSEL     | System error control release selection                         | <ul style="list-style-type: none"> <li>0b0: Release PWM output control when EMB_STAT.SYSF=0.</li> <li>0b1: Release PWM output control when EMB_STAT.SYSST=0</li> </ul>         | RW     |
| 2     | CMRSEL      | Voltage comparator comparison result control release selection | <ul style="list-style-type: none"> <li>0b0: Release PWM output control when EMB_STAT.CMPF=0</li> <li>0b1: Release PWM output control when EMB_STAT.CMPST=0</li> </ul>          | RW     |
| 1     | PWMRSEL     | PWM output control release selection                           | <ul style="list-style-type: none"> <li>0b0: Release PWM output control when EMB_STAT.PWMSF=0.</li> <li>0b1: Release PWM output control when EMB_STAT.PWMST=0</li> </ul>        | RW     |
| 0     | Reserved    | Reserved                                                       | -                                                                                                                                                                              | -      |

## 24 General Timer (TimerA)

### 24.1 Introduction

The General Timer A (TimerA) is a timer with a 16-bit counting width and 4-channel PWM output. The timer supports two waveform modes, triangle wave and sawtooth wave, and can generate various PWM waveforms (single-side aligned PWM, double-side symmetrical PWM); supports synchronous start of counter; the compare reference value register supports buffer function; supports cascading between units to achieve 32-bit counting; supports 2-phase quadrature encoding count and 3-phase quadrature encoding count. This product series is equipped with 12 TimerA units, supporting up to 48 PWM outputs.

### 24.2 Features

The basic functions and features of TimerA are shown in the table below:

Table 24-1 Basic Functions and Features of TimerA

|                |                                 |
|----------------|---------------------------------|
| Waveform mode  | Sawtooth wave, triangular wave  |
| Basic function | Up/Down-counting direction      |
|                | Synchronous startup counter     |
|                | Reference value buffer function |
|                | 32-bit cascading count          |
|                | Quadrature encoding count       |
|                | 4 PWM output                    |
|                | Compare match event output      |
| Interrupt type | Compare match interrupt         |
|                | Period match interrupt          |



## 24.3 Functional Description

### 24.3.1 TimerA Block Diagram

The basic block diagram of TimerA is shown in the figure. In the figure, "<t>" is the unit number, that is, "<t>" is 1-12. In the rest of this section, "<t>" refers to the unit number and will not be repeated here.

This series of products carries 12 units of TimerA, among which the bus clock and count clock of units 1 to 4 are both PCLK0 (in this case, the PCLK shown in the block diagram refers to PCLK0); the bus clock and count clock of units 5 to 12 are PCLK1 (in this case, the PCLK shown in the block diagram refers to PCLK1). The counting frequency division clock source of each unit and the sampling reference clock source of the port digital filter are the same as its bus clock and counting clock.

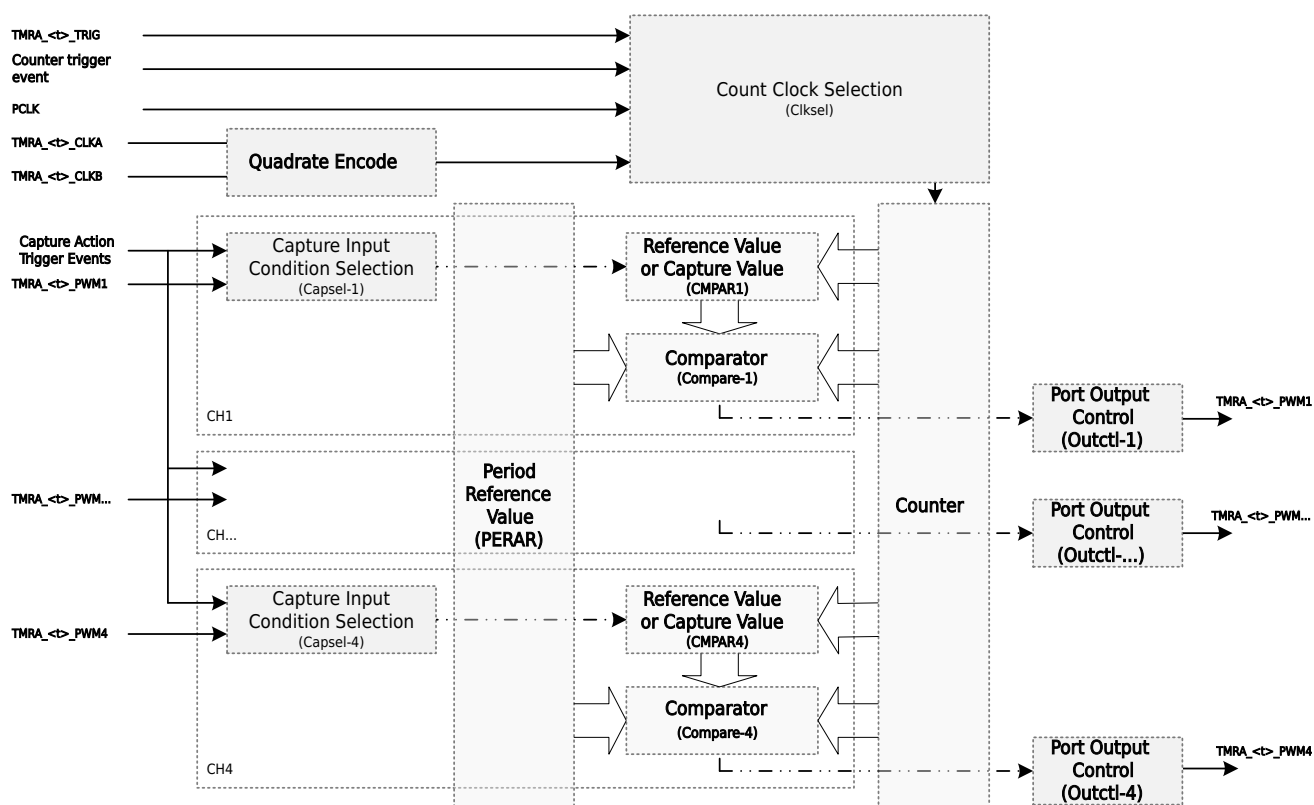


Figure 24-1 TimerA Basic Block Diagram

The following table lists the input and output ports of TimerA.

Table 24-2 TimerA Port List

| Port Name     | Direction    | Function                                                   |
|---------------|--------------|------------------------------------------------------------|
| TMRA_<t>_PWMm | Input/Output | Capture input event port or PWM output port (m=1-4)        |
| TMRA_<t>_CLKA | Input        | Quadrature encoder counting event input port               |
| TMRA_<t>_CLKB |              |                                                            |
| TMRA_<t>_TRIG | Input        | Hardware-triggered start, stop, and clear event input port |

### 24.3.2 Symmetric Units

TimerA unit 1 and unit 2 are symmetric units, as are unit 3 and unit 4, and so on. Symmetric units can be linked to achieve cascaded counting, synchronized start, and other functions, as described in the following sections.

### 24.3.3 Basic Actions

#### 24.3.3.1 Waveform Mode

TimerA has two basic counting wave modes: sawtooth wave mode and triangle wave mode. The basic waveforms of the two waveform modes are as follows:

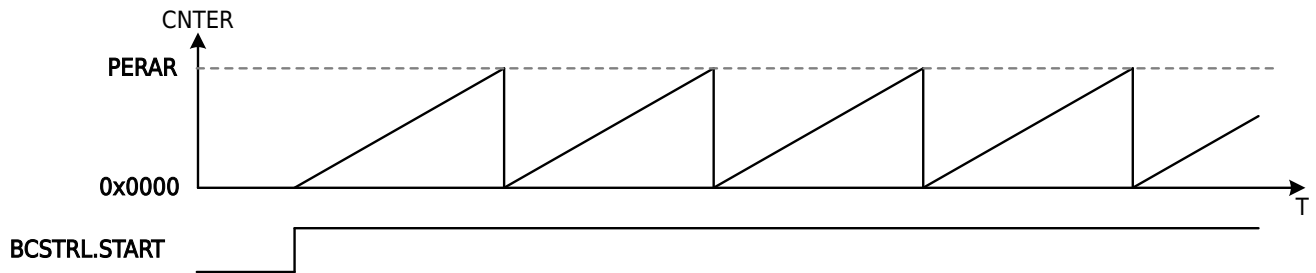


Figure 24-2 Sawtooth Waveform (Counting Up)

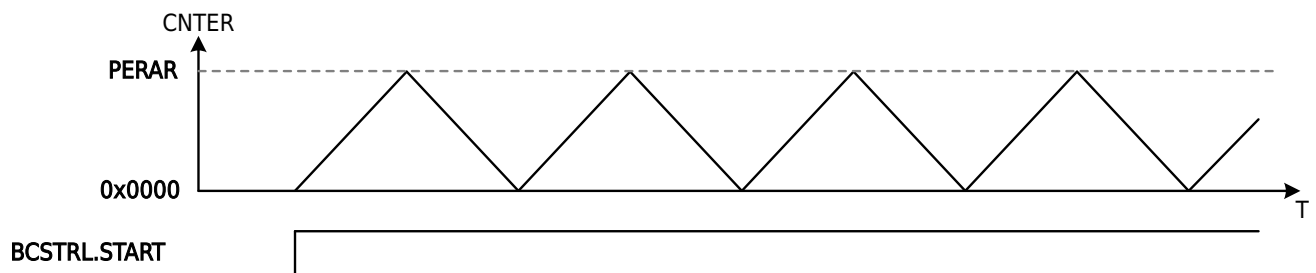


Figure 24-3 Triangle Waveform

#### 24.3.3.2 Compare Output

Each TimerA unit contains 4 channels compare output (TMRA\_<t>\_PWMm), which can output a specified level when the count value matches the compare reference value. The TMRA\_CMPARm registers correspond to the count compare reference value for the TMRA\_<t>\_PWMm output ports. When the timer count value equals TMRA\_CMPARm, the TMRA\_<t>\_PWMm port outputs the specified level (m=1-4).

The output level of the TMRA\_<t>\_PWMm port at various timing events -- such as the start of counting, stop of counting, count compare match, and count period match -- can be configured using the STAC, STPC, CMPC, PERC, and FORC bits in PCONRm register (m=1-4). The diagram below illustrates an example of compare output action for Unit 1.

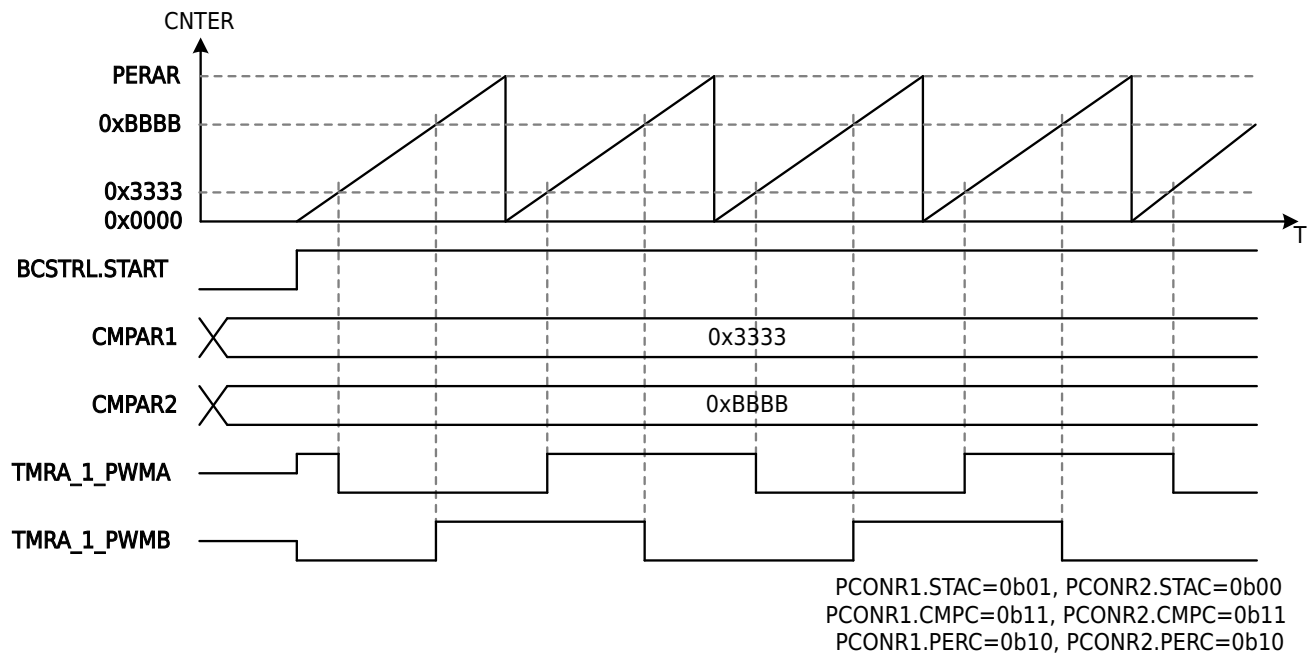


Figure 24-4 Compare Output Action

24.3.3.3 Capture Input

Each PWM output channel of every TimerA unit is equipped with a capture input function that stores the captured count value. When the CCONR.CAPMD bit in the Capture Control Register (CCONRm) is set to 1, the capture input function becomes active. At this point, when the corresponding capture input condition is selected and this condition is valid, the current count value is stored in the corresponding register (CMPARm) (m=1-4).

The capture input condition can select the internal capture action trigger event (selected by the TMRA\_TRGSELx register, for the specific register content, see the "TimerA Hardware Trigger Event Selection Register (TMRA\_TRGSELx) (x=0-3) " in the "Automatic Operation System (AOS) " chapter), TMRA\_<t>\_PWMm port input, etc. The specific condition selection can be set through the CCONRm.HICP bit (m=1-4).

The internal trigger event selection register is shared by all 12 units of TimerA, and the specific allocation is shown in the following table.

Table 24-3 TRGSEL Selection Relationship Table for Internal Trigger Events

| TimerA Unit | Event Classification                                                                 | Corresponding Selection Register |
|-------------|--------------------------------------------------------------------------------------|----------------------------------|
| TimerA_1    | Counter Trigger Events: TRGSEL selection events specified by HCONR, HCUPR, and HCDOR | TMRA_TRGSEL0                     |
| TimerA_2    |                                                                                      | TMRA_TRGSEL1                     |
| TimerA_3    |                                                                                      | TMRA_TRGSEL2                     |
| TimerA_4    |                                                                                      | TMRA_TRGSEL3                     |
| TimerA_5    |                                                                                      | TMRA_TRGSEL0                     |
| TimerA_6    |                                                                                      | TMRA_TRGSEL1                     |
| TimerA_7    |                                                                                      | TMRA_TRGSEL2                     |
| TimerA_8    |                                                                                      | TMRA_TRGSEL3                     |
| TimerA_9    |                                                                                      | TMRA_TRGSEL0                     |
| TimerA_10   |                                                                                      | TMRA_TRGSEL1                     |
| TimerA_11   |                                                                                      | TMRA_TRGSEL2                     |
| TimerA_12   |                                                                                      | TMRA_TRGSEL3                     |
| TimerA_1    | Capture Action Trigger Events: TRGSEL selection events specified by CCONR            | TMRA_TRGSEL1                     |
| TimerA_2    |                                                                                      | TMRA_TRGSEL0                     |
| TimerA_3    |                                                                                      | TMRA_TRGSEL3                     |
| TimerA_4    |                                                                                      | TMRA_TRGSEL2                     |
| TimerA_5    |                                                                                      | TMRA_TRGSEL1                     |
| TimerA_6    |                                                                                      | TMRA_TRGSEL0                     |
| TimerA_7    |                                                                                      | TMRA_TRGSEL3                     |
| TimerA_8    |                                                                                      | TMRA_TRGSEL2                     |
| TimerA_9    |                                                                                      | TMRA_TRGSEL1                     |
| TimerA_10   |                                                                                      | TMRA_TRGSEL0                     |
| TimerA_11   |                                                                                      | TMRA_TRGSEL3                     |
| TimerA_12   |                                                                                      | TMRA_TRGSEL2                     |

The following figure shows an example of the capture input action.

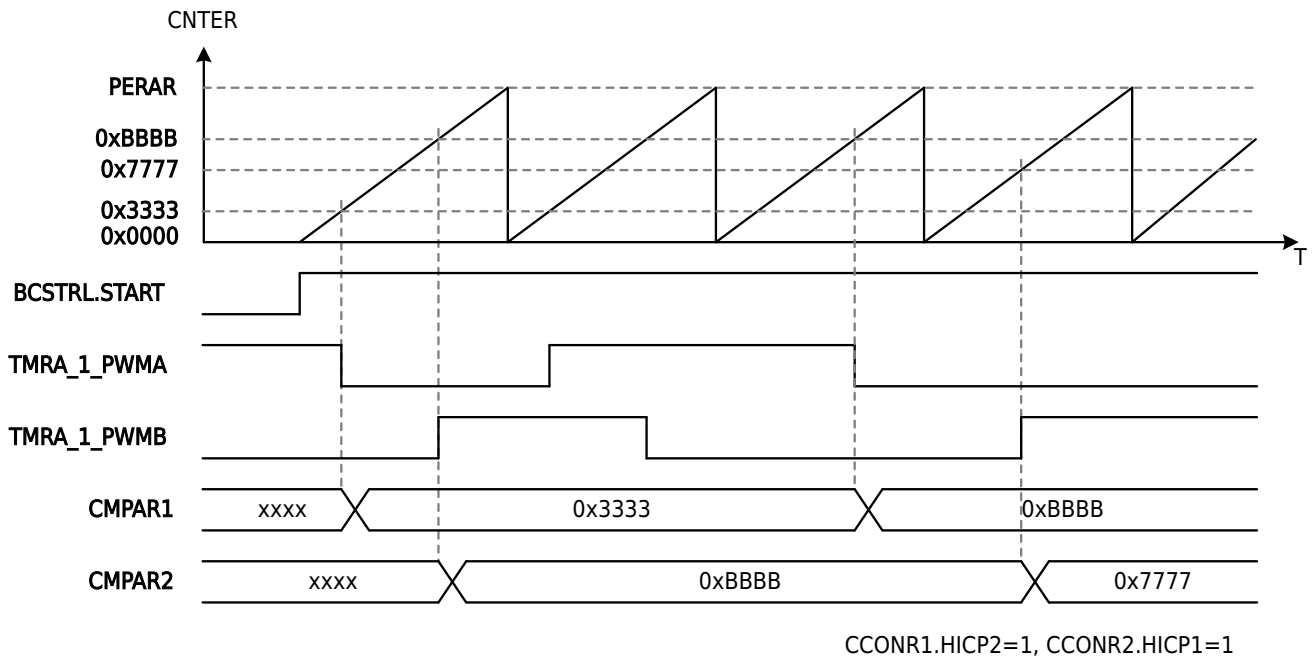


Figure 24-5 Capture Input Action

### 24.3.4 Clock Source Selection

The TimerA count clock can be selected from the following sources:

1. 1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024 division of PCLK (configured by BCSTR.L.CKDIV)
2. TMRA\_<t>\_TRIG port event input (configured by HCUPR.HCUPn or HCDOR.HCDO n) (n=8-9)
3. Internal counter trigger event input (configured by HCUPR.HCUP10 or HCDOR.HCDO10)
4. Count overflow or underflow event input for symmetric units (configured by HCUPR.HCUPn or HCDOR.HCDO n) (n=11-12)
5. TMRA\_<t>\_CLKA and TMRA\_<t>\_CLKB port quadrature encoder input (configured by HCUPR.HCUPn or HCDOR.HCDO n) (n=0-7)

When the clock source selection is set to 1, the timer operates in software counting mode. When set to others, the timer operates in hardware counting mode. In hardware counting mode, ensure that BCSTR.L.CKDIV is 0. Each count clock source is independent and can be enabled or disabled individually. When the hardware clocks are selected, the software clock is automatically disabled.

### 24.3.5 Count Direction

The counting direction of TimerA can be changed via software. The method of changing the counting direction may vary slightly depending on the selected waveform mode.

#### 24.3.5.1 Sawtooth Wave Count Direction

In sawtooth wave counting mode, the counting direction can be set either during the timer counting or when the counting is stopped. Once set, the direction changes immediately.

#### 24.3.5.2 Triangle Wave Count Direction

In the triangle wave counting mode, the counting direction is set when counting stops. The direction setting is disabled when counting is in progress.

### 24.3.6 Digital Filtering

Each unit's TMRA\_<t>\_CLKA, TMRA\_<t>\_CLKB, TMRA\_<t>\_TRIG, TMRA\_<t>PWMm (when capturing input function) port inputs have digital filtering functions. The filter function of each port can be enabled and the filter clock can be selected by setting the corresponding bits of the FCONR register and the CCONRm register (m=1-4).

When the port level sampled by the filter sampling reference clock remains consistent for 3 consecutive times, that level is transmitted as a active level and transmitted to the internal block. Levels with fewer than 3 consistent samples are filtered as external noise and not transmitted to the internal block. The figure below shows an example of TMRA\_1\_CLKA port filtering action.

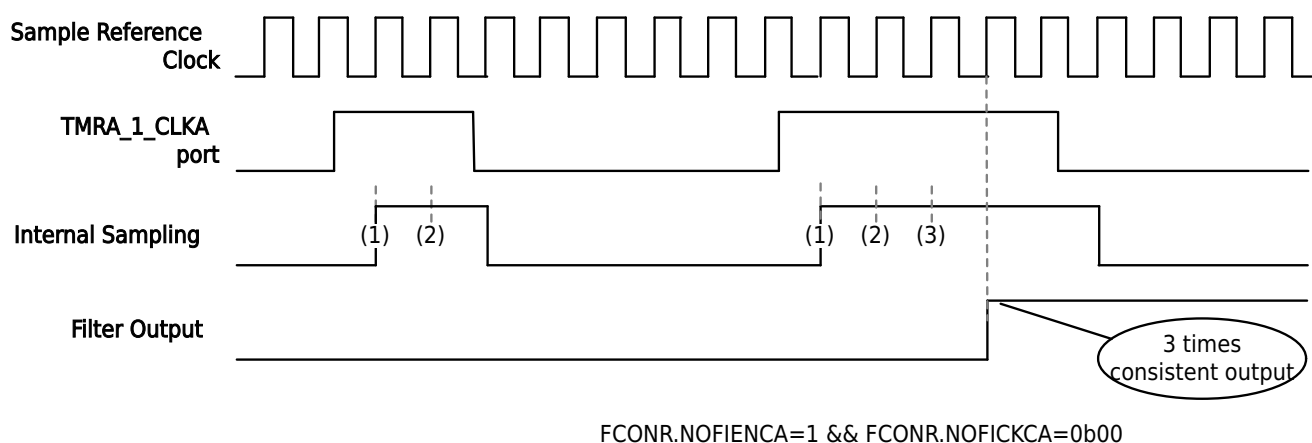


Figure 24-6 Filtering Function of Clock Input Port

### 24.3.7 Synchronous Starting

The symmetric units of TimerA can realize software synchronous start or hardware synchronous start, and the odd unit synchronously starts the even unit.

When the BCSTRL.SYNST bit of the even unit in the symmetric unit is set to 1, the synchronous start function of the even unit and the odd unit is enable. At this time, if the software sets the BCSTRL.START bit of the odd unit to 1, the counter of the synchronized even unit starts software synchronous counting; if the hardware sets any bit in HCONR.HSTA1-0 of the even unit to 1, and the corresponding hardware event occurs, the counter of the synchronized even unit starts hardware synchronous counting. When the hardware synchronous count start function is selected, the corresponding bits of HCONR.HSTA1-0 of the synchronized even unit must also be set to active.

The following figure shows an example of software synchronization startup when BCSTRL.SYNST=1 of unit 2 is set.

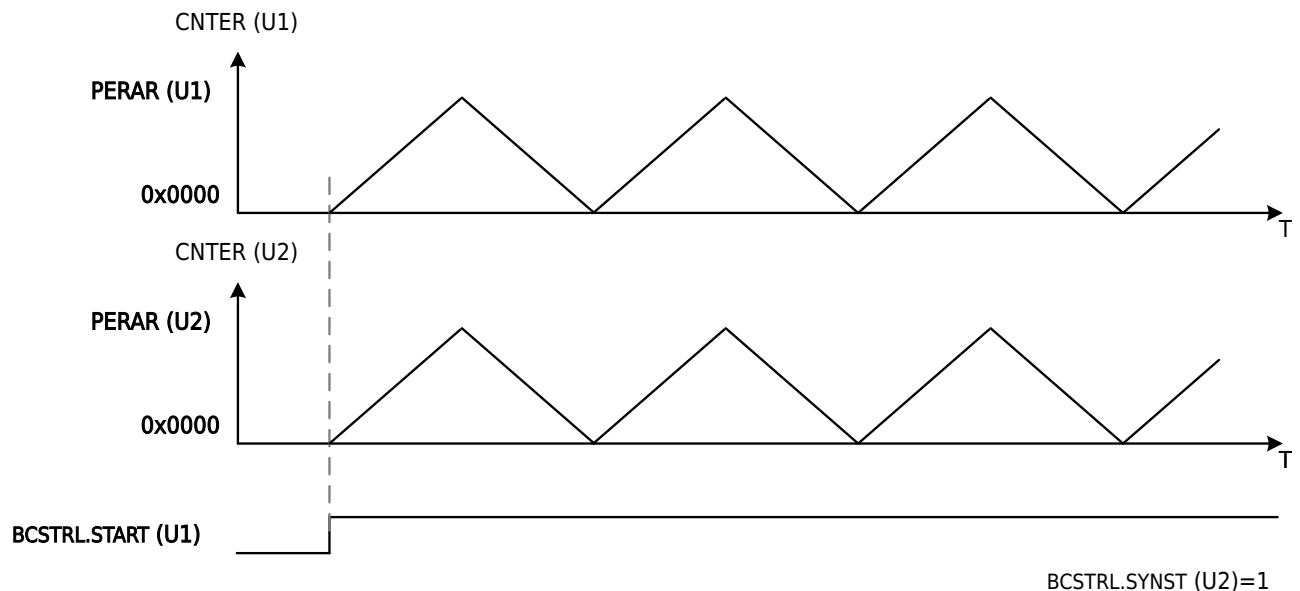


Figure 24-7 Software Synchronization Action

In addition, the product is equipped with a Counter Synchronization Enable Register TMR\_SYNENRx ( $x=1-2$ ) to synchronously start all the timers (including Timer0, Timer2, Timer4, Timer6, TimerA) carried by the product. For usage details and register descriptions, refer to the "General Timer (Timer0)" chapter.

### 24.3.8 Buffer Function

A total of 4 compare reference registers (CMPARm) ( $m=1-4$ ) of TimerA can be used in pairs to implement the buffer function. That is, CMPAR2 is used as the buffer value of CMPAR1, CMPAR4 is used as the buffer value of CMPAR3. The buffer control register (BCONRm) ( $m=1-2$ ) controls the two groups of buffer functions respectively.

When the BCONRm.BEN ( $m=1-2$ ) bit is set, the buffer function becomes active. When the counter reaches a certain time point, a buffer transmission occurs (CMPAR2/4- $\rightarrow$  CMPAR1/3). The "Certain Time Point" includes the following situations:

1. In hardware counting mode, the counter reaches the overflow point (when HCUPR!=0), the underflow point (when HCDOR!=0), or a hardware clear action occurs for the counter.
2. In sawtooth counting mode (BCSTRL.MODE=0), the counter counts to the overflow point (BCSTRL.DIR=1) or underflow point (BCSTRL.DIR=0) or the hardware clears the counter.
3. In triangle wave counting mode (BCSTRL.MODE=1), count to peak point (BCSTRL.DIR=1 && BCONRm.BSE0=1) ( $m=1-2$ )
4. In triangle wave counting mode (BCSTRL.MODE=1), count to valley point (BCSTRL.DIR=0 && BCONRm.BSE1=1) ( $m=1-2$ )

When the BCONRm.BSEN bit is set, the buffer special function becomes active ( $m=1-2$ ). When the CNTNR of the symmetric unit is read, a buffer transmission occurs in the CMPAR of this unit (CMPAR2/4- $\rightarrow$  CMPAR1/3).

The figure below is a schematic diagram of buffer transmission in sawtooth mode.

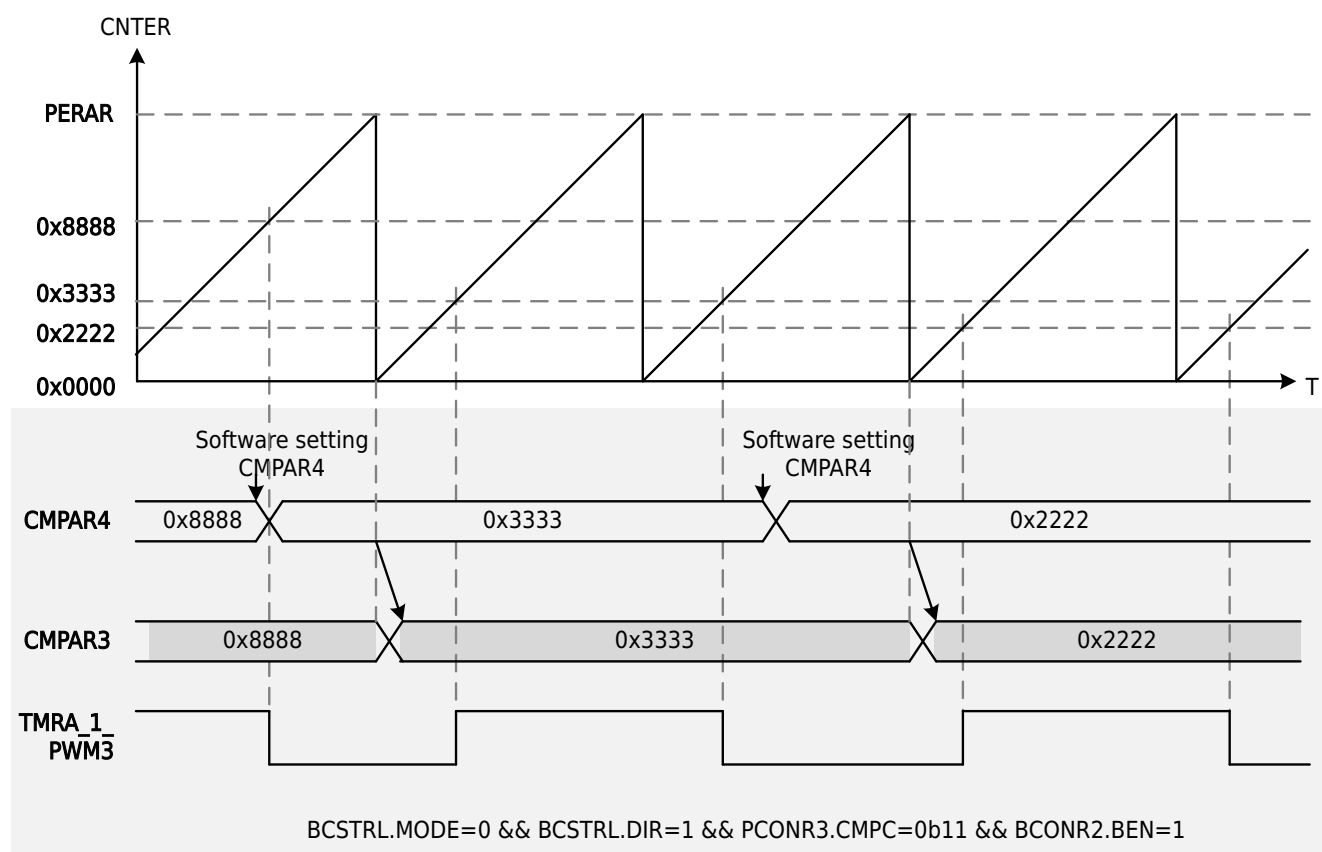


Figure 24-8 Buffer Action in Sawtooth Mode

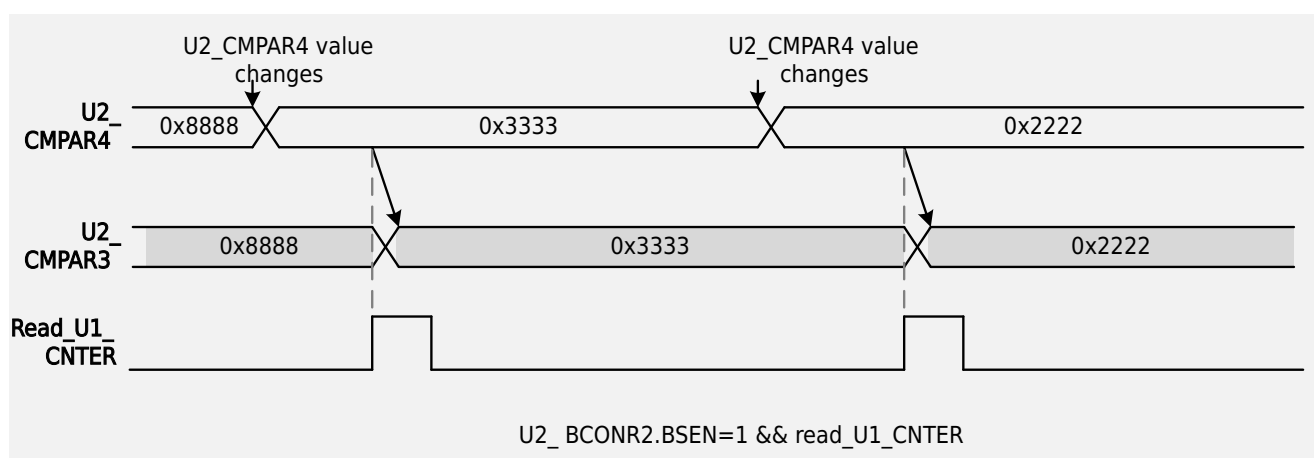


Figure 24-9 Buffer and Special Buffer Actions in Sawtooth Mode

### 24.3.9 Cascade Counting

As described in the "[Clock Source Selection](#)" section, when the clock source is set to 4, the counting clock of this unit is selected as the count overflow (count up or count down) event of the symmetric unit. This allows the two units to be cascade combined, creating a 32-bit counter. In cascaded counting, the CNTER register of the symmetric unit serves as the lower 16-bit counter, while the CNTER register of this unit serves as the upper 16-bit counter.

For example, in triangular up-counting mode (BCSTR.L.MODE=0, BCSTR.L.DIR=1), if unit 1 is configured to use PCLK as the counting clock and unit 2 is configured to use the overflow event of unit 1 as its counting clock source (TMRA\_HCUPR.HCUP11=1 in unit 2), then starting the counters of unit 2 and unit 1 (start unit 2 first, then unit 1) enables cascaded counting. In this configuration, unit 1's CNTER serves



as the lower 16-bit counter, and unit 2's CNTER serves as the upper 16-bit counter. The figure below is a schematic diagram of the cascade counting of units 1 and 2.

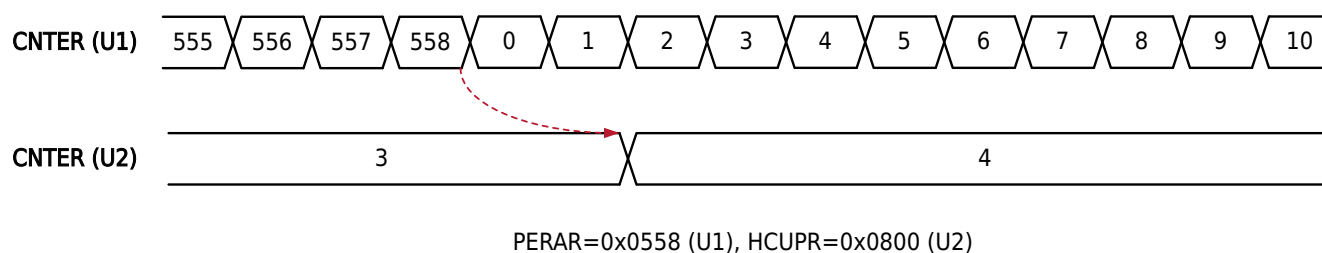


Figure 24-10 32-bit Cascade Counting Action

## 24.3.10 General PWM Output

### 24.3.10.1 Single-Side Aligned PWM Output

In sawtooth counting mode, each channel within a unit can be configured via various port control settings to achieve single-side aligned PWM output (aligned at the period match point). For example, by setting the output to toggle on compare match (PCONR.CMPC=0b11) and also to toggle on period match (PCONR.PERC=0b11), an edge-aligned PWM waveform can be generated within one counting period.

The figure below shows an example of single-side aligned PWM output waveform in sawtooth mode.

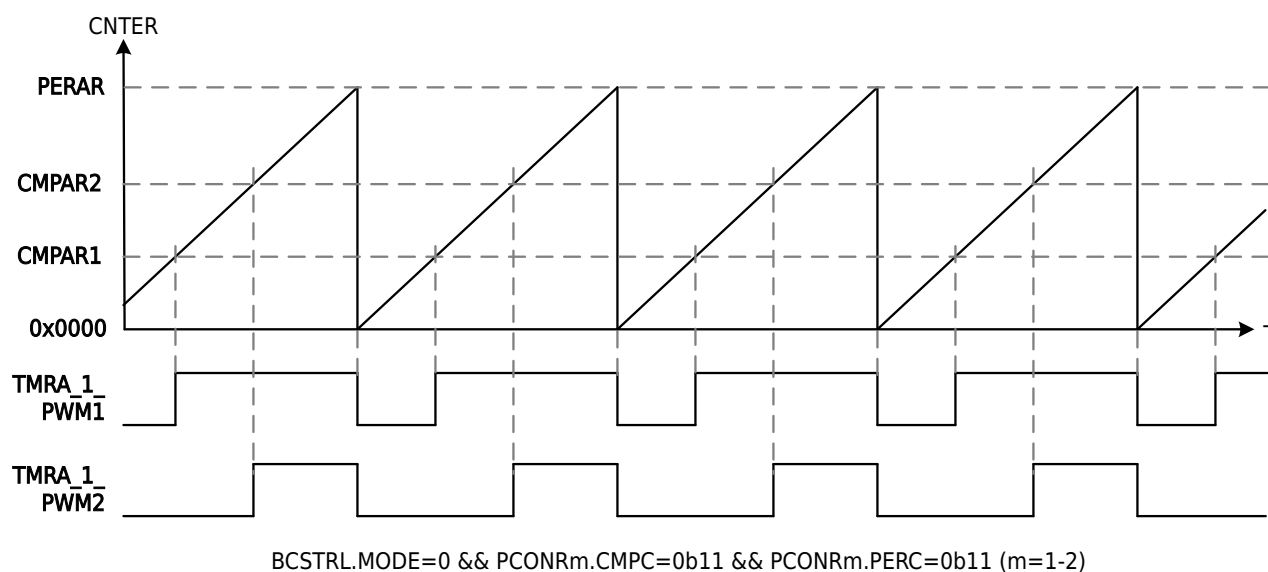


Figure 24-11 Example of Single-Side Aligned PWM Output

### 24.3.10.2 Double-Side Symmetric PWM Output

In triangle counting mode, each channel within a unit can be configured via various port control settings to achieve double-side symmetric PWM output, where the waveform is symmetrical around the counting peak point. Depending on the output relationship between channels, either independent PWM outputs or complementary PWM outputs can be realized.

The figure below shows an example of double-side symmetric PWM output waveforms for channels 1, 2, 3, and 4 in triangle mode. Among them, channels 1 and 2 can be used as a pair of complementary PWM outputs.

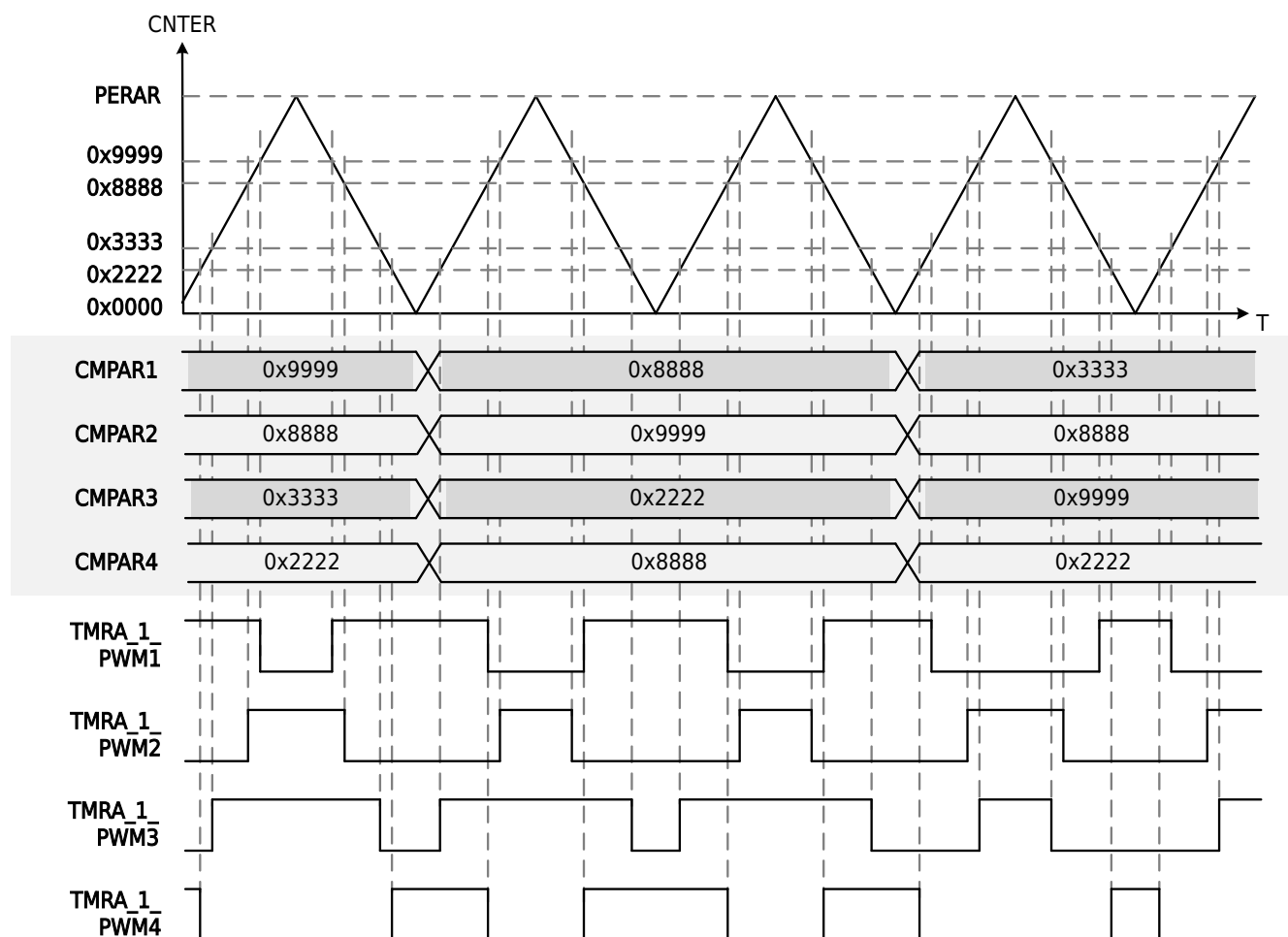


Figure 24-12 Example of Double-Side Symmetric PWM Output

### 24.3.11 Quadrature Encoding Count

By considering the TMRA\_<t>\_CLKA input as AIN input, the TMRA\_<t>\_CLKB input as BIN input, and the TMRA\_<t>\_TRIG input as ZIN input, TimerA can implement quadrature encoding counting of three inputs.

The individual actions of AIN and BIN of each unit can realize the position counting mode; the combined actions of AIN, BIN and ZIN of two units can realize the revolution counting mode. The unit used for position counting is called the position counting unit, and the unit used for revolution counting is called the revolution counting unit. In the revolution counting mode, every two units are combined with each other (unit 1 and 2 are combined; unit 3 and 4 are combined; unit 5 and 6 are combined; unit 7 and 8 are combined; unit 9 and 10 are combined; unit 11 and 12 are combined), and the position counting unit and the revolution counting unit in the combination can be specified arbitrarily.

The counting condition enable of AIN and BIN is realized by setting the quadrature relationship between TMRA\_<t>\_CLKA and TMRA\_<t>\_CLKB in the HCUPR register and the HCDOR register; the input action of ZIN is realized by setting the clear enable bit of the HCONR register of the position counting unit to clear the position timer, and by setting HCUPR register of the revolution unit to count the revolution timer.

#### 24.3.11.1 Position Counting Mode

The quadrature encoder position counting mode enables the implementation of basic counting, phase difference counting, and direction counting based on the input signals from AIN and BIN.

## Basic Counting

Basic counting uses the input clock from either the AIN or BIN port to count.

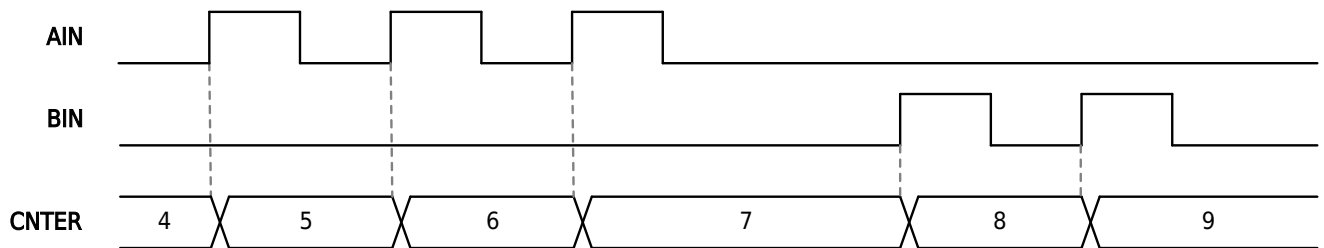
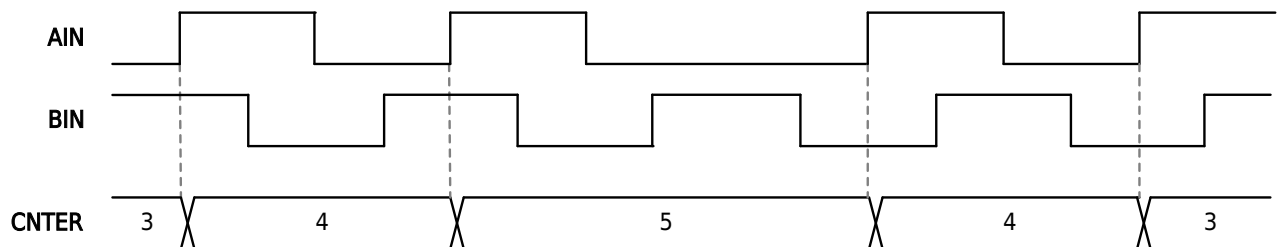


Figure 24-13 Position Mode-Basic Counting

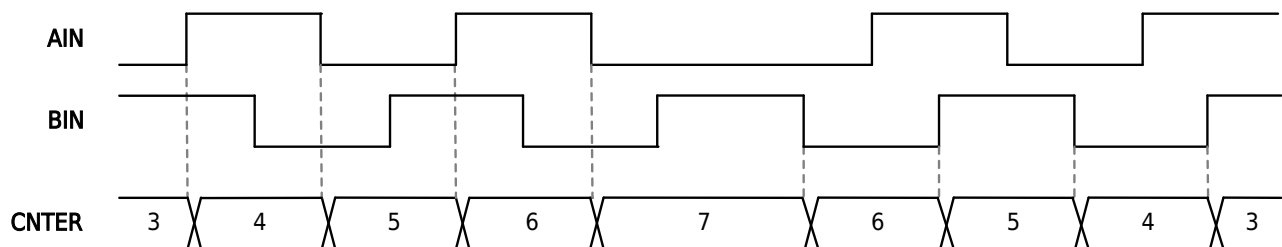
## Phase Difference Counting

Phase difference counting operates based on the phase relationship between AIN and BIN. Depending on the configuration, the system can perform 1x counting, 2x counting, or 4x counting.



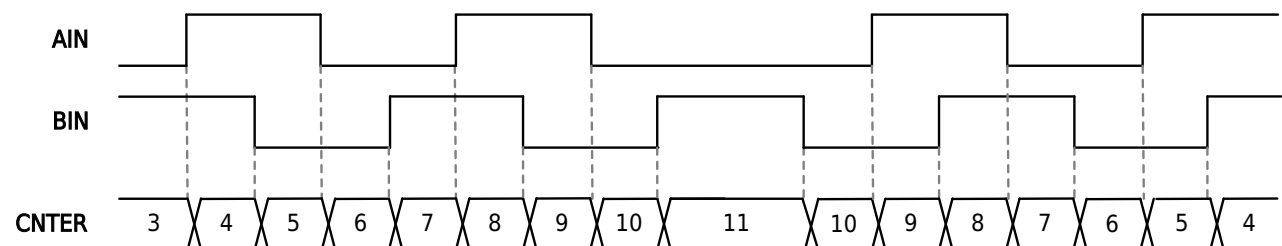
HCUPR.HCUP6=0b1 && HCDOR.HCDO4=0b1

Figure 24-14 Position Counting Mode-Phase Difference Counting (1x Counting)



HCUPR.HCUP6=HCUPR.HCUP5=0b1 && HCDOR.HCDO2=HCDOR.HCDO1=0b1

Figure 24-15 Position Counting Mode-Phase Difference Counting (2x Counting)



HCUPR.HCUP6=HCUPR.HCUP5=HCUPR.HCUP3=HCUPR.HCUP0=0b1  
HCDOR.HCDO7=HCDOR.HCDO4=HCDOR.HCDO2=HCDOR.HCDO1=0b1

Figure 24-16 Position Counting Mode-Phase Difference Counting (4x Counting)

### Direction Counting

Direction counting sets the AIN input as the direction control, and uses the BIN input as the counting clock.

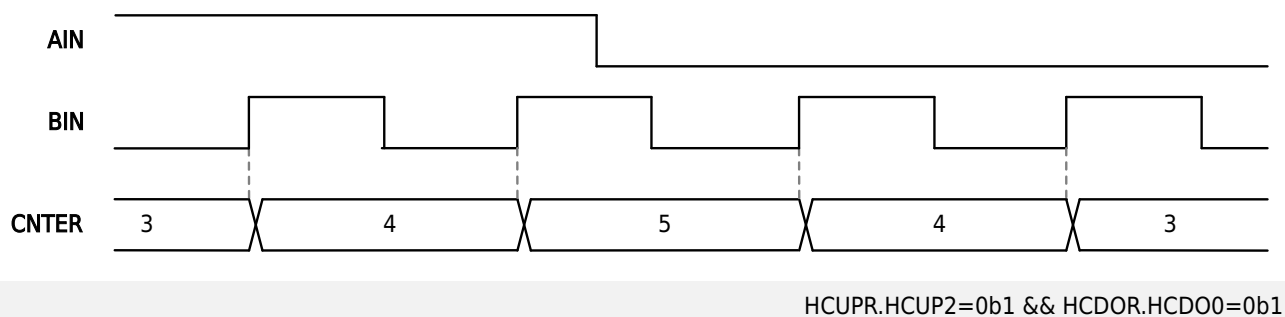


Figure 24-17 Position Counting Mode-Direction Counting

### 24.3.11.2 Revolution Counting Mode

Quadrature encoding revolution mode refers to determining the revolution count by incorporating ZIN input events on the basis of AIN and BIN counting. In revolution counting mode, the Z phase count function, position overflow counting function, and mixed counting function can be implemented based on the counting method of the revolution timer.

#### Z-phase Counting

Z-phase counting refers to a counting operation in which the revolution counting unit performs counting based on the ZIN input, while the position counting unit is reset to zero.

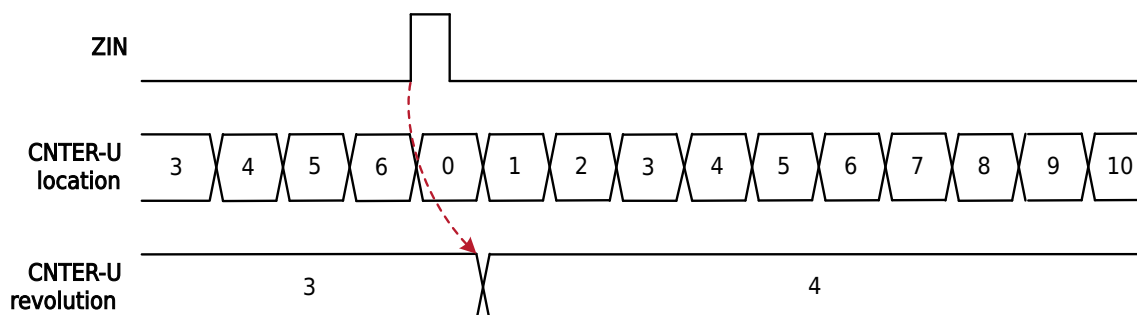


Figure 24-18 Revolution Counting Mode-Z-phase Counting

### Position Overflow Counting

Position overflow counting refers to the situation where an overflow event is generated when the position counting unit's count overflows or underflows, thereby triggering the timer of the revolution counting unit to perform one count. (In this counting mode, the ZIN input does not trigger counting in the revolution counting unit or clear the position counting unit.)

The count up (count down) event Bits 12-11 of the HCUPR or HCDOR register of the revolution counting unit are enabled, and the overflow event of the position counting unit can trigger the revolution counting unit to count once.

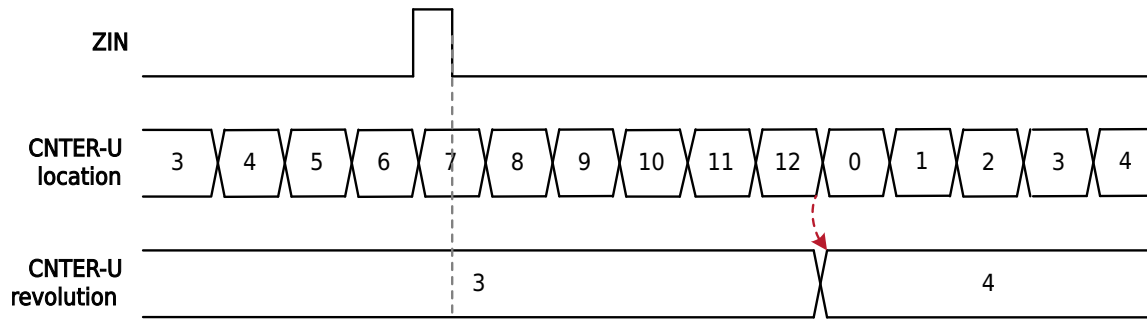


Figure 24-19 Revolution Counting Mode-Position Overflow Counting

### Mixed Counting

Mixed counting refers to a combined counting method that integrates both Z-phase counting and position overflow counting. Its implementation method is also a combination of the above two counting methods.

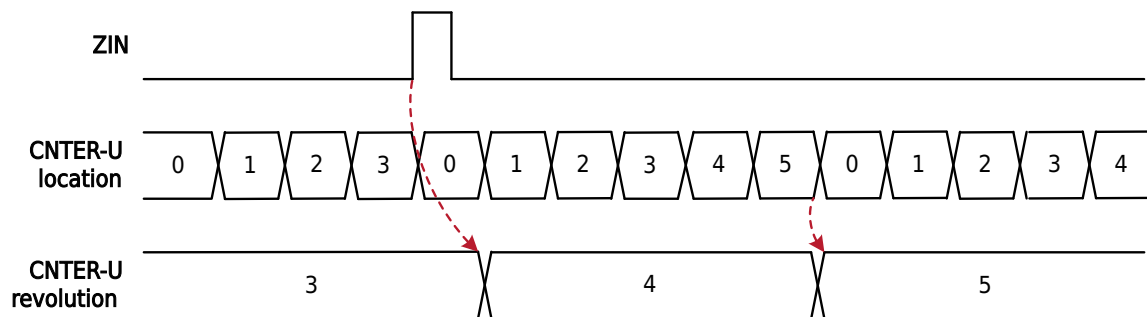


Figure 24-20 Revolution Counting Mode-Mixed Counting

## 24.3.12 Interrupt and Events

TimerA contains 3 interrupt outputs and 3 event outputs, which are 1 comparison match interrupt and event, and 2 period match interrupts and events.

### 24.3.12.1 Compare Match Interrupt and Event

When a comparison match occurs between the CMPARM register and the count value, the STFLR.CMPFm bit will be set to 1. At this time, if the ICONR.ITENm bit is set to 1, the corresponding interrupt request (TMRA\_<t>\_CMP) will be triggered; if the ECONR.ETENm bit is set to 1, the corresponding event request (TMRA\_<t>\_CMP) will be triggered (m=1-4).

When the capture input active condition selected by the CCONRM register occurs, the capture input action occurs. At this time, if the ICONR.ITENm bit is set to 1, the corresponding interrupt request (TMRA\_<t>\_CMP) will be triggered; if the ECONR.ETENm bit is set to 1, the corresponding event request (TMRA\_<t>\_CMP) will be triggered (m=1-4).

The comparison match interrupts and comparison match events of the 4 reference value within each unit are not output independently. The comparison match interrupts are summarized into an interrupt output to the interrupt module through "OR logic" (see the "Interrupt Controller (INTC)" chapter), and the comparison match events are summarized into an event output through "OR logic" to select trigger other modules.

### 24.3.12.2 Period Match Interrupt and Event

The sawtooth wave mode counts up to the overflow point, the sawtooth wave mode counts down to the underflow point, and the triangle wave mode counts to the valley point or peak point. The BCSTRH.OVFF or BCSTRH.UDFF bit will be set to 1. At this time, if the BCSTRH.ITENOVF or BCSTRH.ITENUDF bit is

set to 1 to enable interrupts, the period match interrupt (TMRA\_<t>\_OVF and TMRA\_<t>\_UDF) can be triggered at the corresponding period point to be output to the interrupt module (INTC); the period match event has no corresponding enable bit control, and the period match event (TMRA\_<t>\_OVF and TMRA\_<t>\_UDF) output is triggered at the corresponding counting period point to select triggering other modules.

## 24.4 Register Description

### 24.4.1 Register List

- TimerA\_1 base address: 0x4003A000
- TimerA\_2 base address: 0x4003A400
- TimerA\_3 base address: 0x4003A800
- TimerA\_4 base address: 0x4003AC00
- TimerA\_5 base address: 0x40026000
- TimerA\_6 base address: 0x40026400
- TimerA\_7 base address: 0x40026800
- TimerA\_8 base address: 0x40026C00
- TimerA\_9 base address: 0x40027000
- TimerA\_10 base address: 0x40027400
- TimerA\_11 base address: 0x40027800
- TimerA\_12 base address: 0x40027C00



#### Note

TimerA is the name of the timer, and TimerA\_<t> (<t>: 1-12) is the timer unit.

Table 24-4 General Timer (TimerA) Register Offset

| Offset       | Register    | Description                                | Width |
|--------------|-------------|--------------------------------------------|-------|
| 0x0000       | TMRA_CNTER  | General Count Value Register               | 16    |
| 0x0004       | TMRA_PERAR  | Period Reference Value Register            | 16    |
| 0x003C+0x4*m | TMRA_CMPARm | Compare Reference Value Register m (m=1-4) | 16    |
| 0x0080       | TMRA_BCSTRL | Control Status Register L                  | 8     |
| 0x0081       | TMRA_BCSTRH | Control Status Register H                  | 8     |
| 0x0090       | TMRA_ICONR  | Interrupt Control Register                 | 16    |
| 0x0094       | TMRA_ECONR  | Event Control Register                     | 16    |
| 0x0098       | TMRA_FCONR  | Filter Control Register                    | 16    |
| 0x009C       | TMRA_STFLR  | Status Flag Register                       | 16    |
| 0x00B8+0x8*m | TMRA_BCONRm | Buffer Control Register m (m=1-2)          | 16    |
| 0x00FC+0x4*m | TMRA_CCONRm | Capture Control Register m (m=1-4)         | 16    |
| 0x013C+0x4*m | TMRA_PCONRm | Port Control Register m (m=1-4)            | 16    |
| 0x0084       | TMRA_HCONR  | Hardware Trigger Event Selection Register  | 16    |
| 0x0088       | TMRA_HCUPR  | Hardware Count Up Event Selection Register | 16    |

| Offset | Register   | Description                                  | Width |
|--------|------------|----------------------------------------------|-------|
| 0x008C | TMRA_HCDOR | Hardware Count Down Event Selection Register | 16    |



24.4.2 General Count Value Register (TMRA\_CNTER)

| Offset | Bit Position |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0x0000 | 15           | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x0000       |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | CNT          |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access | RW           |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit  | Name | Function    | Description                                                                                                                                                                                                                                                                                                                                                                           | Access |
|------|------|-------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 15:0 | CNT  | Count Value | <div>Current timer count value</div> <div> <b>Note</b><br/>In the sawtooth up-counting mode, the initial value of CNT is set to 0. In the sawtooth down-counting mode, the initial value of CNT is set to the period value. In the triangular wave mode, the initial value of CNT is set to 1.</div> | RW     |

24.4.3 Period Reference Value Register (TMRA\_PERAR)

| Offset | Bit Position |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0x0004 | 15           | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0xFFFF       |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | PER          |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access | RW           |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit  | Name | Function              | Description                                            | Access |
|------|------|-----------------------|--------------------------------------------------------|--------|
| 15:0 | PER  | Counting Period Value | Set the counting period value for each counting period | RW     |

24.4.4 Compare Reference Value Register (TMRA\_CMPARm) (m=1-4)

Offset: 0x003C+0x4\*m

| Offset | Bit Position |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| -      | 15           | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0xFFFF       |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | CMP          |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access | RW           |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit  | Name | Function                      | Description                     | Access |
|------|------|-------------------------------|---------------------------------|--------|
| 15:0 | CMP  | Count Compare Reference Value | Set the compare reference value | RW     |

## 24.4.5 Control Status Register L (TMRA\_BCSTRL)

| Offset | Bit Position |   |   |   |       |      |     |       |
|--------|--------------|---|---|---|-------|------|-----|-------|
| 0x0080 | 7            | 6 | 5 | 4 | 3     | 2    | 1   | 0     |
| Reset  | 0x02         |   |   |   |       |      |     |       |
| Name   | CKDIV        |   |   |   | SYNST | MODE | DIR | START |
| Access | RW           |   |   |   | RW    | RW   | RW  | RW    |

| Bit | Name  | Function                  | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | Access |
|-----|-------|---------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 7:4 | CKDIV | Count Clock Selection     | <ul style="list-style-type: none"> <li>0b0000: PCLK</li> <li>0b0001: PCLK/2</li> <li>0b0010: PCLK/4</li> <li>0b0011: PCLK/8</li> <li>0b0100: PCLK/16</li> <li>0b0101: PCLK/32</li> <li>0b0110: PCLK/64</li> <li>0b0111: PCLK/128</li> <li>0b1000: PCLK/256</li> <li>0b1001: PCLK/512</li> <li>0b1010: PCLK/1024</li> <li>Others: Setting prohibited</li> </ul>                                                                                                                                                                                                   | RW     |
| 3   | SYNST | Synchronous Start Enabled | <ul style="list-style-type: none"> <li>0b0: Synchronous start between the even unit and the counting unit in the symmetric unit is disabled</li> <li>0b1: Synchronous start between the even unit and the counting unit in the symmetric unit is enabled</li> </ul> <div>  <b>Note</b><br/>           This setting has no effect for odd unit TimerA units in a symmetric unit or TimerA units without a symmetric unit; the bit will read as 0 in such cases.         </div> | RW     |
| 2   | MODE  | Counting Mode             | <ul style="list-style-type: none"> <li>0b0: Sawtooth wave mode</li> <li>0b1: Triangle wave mode</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                       | RW     |
| 1   | DIR   | Count Direction           | <ul style="list-style-type: none"> <li>0b0: Counter count down</li> <li>0b1: Counter count up</li> </ul> <div>  <b>Note</b><br/>           In triangle wave mode, this bit indicates the direction status of the counter.         </div>                                                                                                                                                                                                                                      | RW     |

| Bit | Name  | Function    | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | Access |
|-----|-------|-------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 0   | START | Timer Start | <div><ul style="list-style-type: none"><li>0b0: Timer disabled</li><li>0b1: Timer enabled</li></ul><div><div></div><div><b>Note</b><ul style="list-style-type: none"><li>This bit will automatically be cleared to 0 when the hardware stop condition is active.</li><li>When the synchronous start function of the even unit in the symmetric unit is effective, the bit of the even unit will also be set after the software of the odd unit is started.</li></ul></div></div></div> | RW     |



**Notice**

- For TimerA Units 1 to 4, the PCLK source is PCLK0; for TimerA Units 5 to 12, the PCLK source is PCLK1.
- In triangle wave mode, do not write to this register during timer operation.

## 24.4.6 Control Status Register H (TMRA\_BCSTRH)

| Offset | Bit Position |      |         |         |          |   |   |       |
|--------|--------------|------|---------|---------|----------|---|---|-------|
| 0x0081 | 7            | 6    | 5       | 4       | 3        | 2 | 1 | 0     |
| Reset  | 0x00         |      |         |         |          |   |   |       |
| Name   | UDFF         | OVFF | ITENUDF | ITENOVF | Reserved |   |   | OVSTP |
| Access | RW           | RW   | RW      | RW      |          |   |   | RW    |

| Bit | Name            | Function                    | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | Access |
|-----|-----------------|-----------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 7   | UDFF            | Underflow Flag              | <ul style="list-style-type: none"> <li>0b0: No counter underflow occurred during down counting</li> <li>0b1: Counter underflow occurred during down counting</li> </ul>                                                                                                                                                                                                                                                                                                                             | RW     |
| 6   | OVFF            | Overflow Flag               | <ul style="list-style-type: none"> <li>0b0: No counter overflow occurred during up counting</li> <li>0b1: Counter overflow occurred during up counting</li> </ul>                                                                                                                                                                                                                                                                                                                                   | RW     |
| 5   | ITENUDF         | Underflow Interrupt Enabled | <ul style="list-style-type: none"> <li>0b0: Disable</li> <li>0b1: Enable</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                 | RW     |
| 4   | ITENOVF         | Overflow Interrupt Enabled  | <ul style="list-style-type: none"> <li>0b0: Disable</li> <li>0b1: Enable</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                 | RW     |
| 3:1 | <i>Reserved</i> | <i>Reserved</i>             | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | -      |
| 0   | OVSTP           | Overflow Stop Control       | <ul style="list-style-type: none"> <li>0b0: The counter continues counting after an overflow or underflow.</li> <li>0b1: The counter stops counting after an overflow or underflow.</li> </ul> <div style="border: 1px solid black; padding: 5px; margin-top: 10px;">  <b>Note</b><br/>           The TMRA_CNTER register must be initialized before restarting the timer after an overflow stop.         </div> | RW     |

## 24.4.7 Interrupt Control Register (TMRA\_ICONR)

| Offset | Bit Position |    |    |    |    |    |   |   |   |   |   |   |       |       |       |       |
|--------|--------------|----|----|----|----|----|---|---|---|---|---|---|-------|-------|-------|-------|
| 0x0090 | 15           | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3     | 2     | 1     | 0     |
| Reset  | 0x0000       |    |    |    |    |    |   |   |   |   |   |   |       |       |       |       |
| Name   | Reserved     |    |    |    |    |    |   |   |   |   |   |   | ITEN4 | ITEN3 | ITEN2 | ITEN1 |
| Access |              |    |    |    |    |    |   |   |   |   |   |   | RW    | RW    | RW    | RW    |

| Bit  | Name     | Function                       | Description                                                                                                                                                                                                                                                        | Access |
|------|----------|--------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 15:4 | Reserved | Reserved                       | -                                                                                                                                                                                                                                                                  | -      |
| 3    | ITEN4    | Count Match Interrupt Enable 4 | Interrupt Condition: Triggered when the value of the CMPAR4 register matches the counter value (CNTENR), or when an input capture occurs.<br><ul style="list-style-type: none"> <li>0b0: Disable interrupt output</li> <li>0b1: Enable interrupt output</li> </ul> | RW     |
| 2    | ITEN3    | Count Match Interrupt Enable 3 | Interrupt Condition: Triggered when the value of the CMPAR3 register matches the counter value (CNTENR), or when an input capture occurs.<br><ul style="list-style-type: none"> <li>0b0: Disable interrupt output</li> <li>0b1: Enable interrupt output</li> </ul> | RW     |
| 1    | ITEN2    | Count Match Interrupt Enable 2 | Interrupt Condition: Triggered when the value of the CMPAR2 register matches the counter value (CNTENR), or when an input capture occurs.<br><ul style="list-style-type: none"> <li>0b0: Disable interrupt output</li> <li>0b1: Enable interrupt output</li> </ul> | RW     |
| 0    | ITEN1    | Count Match Interrupt Enable 1 | Interrupt Condition: Triggered when the value of the CMPAR1 register matches the counter value (CNTENR), or when an input capture occurs.<br><ul style="list-style-type: none"> <li>0b0: Disable interrupt output</li> <li>0b1: Enable interrupt output</li> </ul> | RW     |

## 24.4.8 Event Control Register (TMRA\_ECONR)

| Offset | Bit Position |    |    |    |    |    |   |   |   |   |   |   |       |       |       |       |
|--------|--------------|----|----|----|----|----|---|---|---|---|---|---|-------|-------|-------|-------|
| 0x0094 | 15           | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3     | 2     | 1     | 0     |
| Reset  | 0x0000       |    |    |    |    |    |   |   |   |   |   |   |       |       |       |       |
| Name   | Reserved     |    |    |    |    |    |   |   |   |   |   |   | ETEN4 | ETEN3 | ETEN2 | ETEN1 |
| Access |              |    |    |    |    |    |   |   |   |   |   |   | RW    | RW    | RW    | RW    |

| Bit  | Name     | Function                    | Description                                                                                                                                                                                                                                             | Access |
|------|----------|-----------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 15:4 | Reserved | Reserved                    | -                                                                                                                                                                                                                                                       | -      |
| 3    | ETEN4    | Count Match Events Enable 4 | Events Condition: Triggered when the value of the CMPAR4 register matches the counter value (CNTENR), or when an input capture occurs.<br><ul style="list-style-type: none"> <li>0b0: Disable event output</li> <li>0b1: Enable event output</li> </ul> | RW     |
| 2    | ETEN3    | Count Match Events Enable 3 | Events Condition: Triggered when the value of the CMPAR3 register matches the counter value (CNTENR), or when an input capture occurs.<br><ul style="list-style-type: none"> <li>0b0: Disable event output</li> <li>0b1: Enable event output</li> </ul> | RW     |
| 1    | ETEN2    | Count Match Events Enable 2 | Events Condition: Triggered when the value of the CMPAR2 register matches the counter value (CNTENR), or when an input capture occurs.<br><ul style="list-style-type: none"> <li>0b0: Disable event output</li> <li>0b1: Enable event output</li> </ul> | RW     |
| 0    | ETEN1    | Count Match Events Enable 1 | Events Condition: Triggered when the value of the CMPAR1 register matches the counter value (CNTENR), or when an input capture occurs.<br><ul style="list-style-type: none"> <li>0b0: Disable event output</li> <li>0b1: Enable event output</li> </ul> | RW     |



## 24.4.9 Filter Control Register (TMRA\_FCONR)

| Offset | Bit Position |          |    |          |    |          |          |   |          |   |          |   |   |   |          |   |          |
|--------|--------------|----------|----|----------|----|----------|----------|---|----------|---|----------|---|---|---|----------|---|----------|
| 0x0098 | 15           | 14       | 13 | 12       | 11 | 10       | 9        | 8 | 7        | 6 | 5        | 4 | 3 | 2 | 1        | 0 |          |
| Reset  | 0x0000       |          |    |          |    |          |          |   |          |   |          |   |   |   |          |   |          |
| Name   | Reserved     | NOFICKCB |    | NOFIENCB |    | Reserved | NOFICKCA |   | NOFIENCA |   | Reserved |   |   |   | NOFICKTG |   | NOFIENTG |
| Access |              | RW       |    | RW       |    |          | RW       |   | RW       |   |          |   |   |   | RW       |   | RW       |

| Bit   | Name            | Function                                     | Description                                                                                                                                                                                                     | Access |
|-------|-----------------|----------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 15    | <i>Reserved</i> | <i>Reserved</i>                              | -                                                                                                                                                                                                               | -      |
| 14:13 | NOFICKCB        | Filter Sampling Reference Clock Selection CB | <ul style="list-style-type: none"> <li>0b00: PCLK</li> <li>0b01: PCLK/4</li> <li>0b10: PCLK/16</li> <li>0b11: PCLK/64</li> </ul>                                                                                | RW     |
| 12    | NOFIENCB        | Capture Input Port Filter CB                 | <ul style="list-style-type: none"> <li>0b0: Input filtering function for the TMRA_&lt;t&gt;_CLKB port is disabled</li> <li>0b1: Input filtering function for the TMRA_&lt;t&gt;_CLKB port is enabled</li> </ul> | RW     |
| 11    | <i>Reserved</i> | <i>Reserved</i>                              | -                                                                                                                                                                                                               | -      |
| 10:9  | NOFICKCA        | Filter Sampling Reference Clock Selection CA | <ul style="list-style-type: none"> <li>0b00: PCLK</li> <li>0b01: PCLK/4</li> <li>0b10: PCLK/16</li> <li>0b11: PCLK/64</li> </ul>                                                                                | RW     |
| 8     | NOFIENCA        | Capture Input Port Filter CA                 | <ul style="list-style-type: none"> <li>0b0: Input filtering function for the TMRA_&lt;t&gt;_CLKA port is disabled</li> <li>0b1: Input filtering function for the TMRA_&lt;t&gt;_CLKA port is enabled</li> </ul> | RW     |
| 7:3   | <i>Reserved</i> | <i>Reserved</i>                              | -                                                                                                                                                                                                               | -      |
| 2:1   | NOFICKTG        | Filter Sampling Reference Clock Selection TG | <ul style="list-style-type: none"> <li>0b00: PCLK</li> <li>0b01: PCLK/4</li> <li>0b10: PCLK/16</li> <li>0b11: PCLK/64</li> </ul>                                                                                | RW     |
| 0     | NOFIENTG        | Capture Input Port Filter FG                 | <ul style="list-style-type: none"> <li>0b0: Input filtering function for the TMRA_&lt;t&gt;_TRIG port is disabled</li> <li>0b1: Input filtering function for the TMRA_&lt;t&gt;_TRIG port is enabled</li> </ul> | RW     |

**Note**

For TimerA Units 1 to 4, the PCLK source is PCLK0; for TimerA Units 5 to 12, the PCLK source is PCLK1.

## 24.4.10 Status Flag Register (TMRA\_STFLR)

| Offset | Bit Position |    |    |    |       |       |       |       |          |   |   |   |       |       |       |       |
|--------|--------------|----|----|----|-------|-------|-------|-------|----------|---|---|---|-------|-------|-------|-------|
| 0x009C | 15           | 14 | 13 | 12 | 11    | 10    | 9     | 8     | 7        | 6 | 5 | 4 | 3     | 2     | 1     | 0     |
| Reset  | 0x0000       |    |    |    |       |       |       |       |          |   |   |   |       |       |       |       |
| Name   | Reserved     |    |    |    | ICPF4 | ICPF3 | ICPF2 | ICPF1 | Reserved |   |   |   | CMPF4 | CMPF3 | CMPF2 | CMPF1 |
| Access |              |    |    |    | RW    | RW    | RW    | RW    |          |   |   |   | RW    | RW    | RW    | RW    |

| Bit   | Name            | Function              | Description                                                                                                                                                                                                                                                                                                                                                | Access |
|-------|-----------------|-----------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 15:12 | <i>Reserved</i> | <i>Reserved</i>       | -                                                                                                                                                                                                                                                                                                                                                          | -      |
| 11    | ICPF4           | Repeat Capture Flag 4 | Write 0 to clear the flag, write 1 has no effect<br>Read:<br><ul style="list-style-type: none"> <li>0b0: No capture completion action has occurred after the CMPF4 flag was set to 1.</li> <li>0b1: A capture completion action has occurred again after the CMPF4 flag was set to 1.</li> </ul>                                                           | RW     |
| 10    | ICPF3           | Repeat Capture Flag 3 | Write 0 to clear the flag, write 1 has no effect<br>Read:<br><ul style="list-style-type: none"> <li>0b0: No capture completion action has occurred after the CMPF3 flag was set to 1.</li> <li>0b1: A capture completion action has occurred again after the CMPF3 flag was set to 1.</li> </ul>                                                           | RW     |
| 9     | ICPF2           | Repeat Capture Flag 2 | Write 0 to clear the flag, write 1 has no effect<br>Read:<br><ul style="list-style-type: none"> <li>0b0: No capture completion action has occurred after the CMPF2 flag was set to 1.</li> <li>0b1: A capture completion action has occurred again after the CMPF2 flag was set to 1.</li> </ul>                                                           | RW     |
| 8     | ICPF1           | Repeat Capture Flag 1 | Write 0 to clear the flag, write 1 has no effect<br>Read:<br><ul style="list-style-type: none"> <li>0b0: No capture completion action has occurred after the CMPF1 flag was set to 1.</li> <li>0b1: A capture completion action has occurred again after the CMPF1 flag was set to 1.</li> </ul>                                                           | RW     |
| 7:4   | <i>Reserved</i> | <i>Reserved</i>       | -                                                                                                                                                                                                                                                                                                                                                          | -      |
| 3     | CMPF4           | Count Match Flag 4    | Write 0 to clear the flag, write 1 has no effect<br>Read:<br><ul style="list-style-type: none"> <li>0b0: The value in the CMPAR4 register does not match the counter value, and no capture completion event has occurred.</li> <li>0b1: The value in the CMPAR4 register matches the counter value, or a capture completion event has occurred.</li> </ul> | RW     |

| Bit | Name  | Function           | Description                                                                                                                                                                                                                                                                                                                                                | Access |
|-----|-------|--------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 2   | CMPF3 | Count Match Flag 3 | Write 0 to clear the flag, write 1 has no effect<br>Read:<br><ul style="list-style-type: none"> <li>0b0: The value in the CMPAR3 register does not match the counter value, and no capture completion event has occurred.</li> <li>0b1: The value in the CMPAR3 register matches the counter value, or a capture completion event has occurred.</li> </ul> | RW     |
| 1   | CMPF2 | Count Match Flag 2 | Write 0 to clear the flag, write 1 has no effect<br>Read:<br><ul style="list-style-type: none"> <li>0b0: The value in the CMPAR2 register does not match the counter value, and no capture completion event has occurred.</li> <li>0b1: The value in the CMPAR2 register matches the counter value, or a capture completion event has occurred.</li> </ul> | RW     |
| 0   | CMPF1 | Count Match Flag 1 | Write 0 to clear the flag, write 1 has no effect<br>Read:<br><ul style="list-style-type: none"> <li>0b0: The value in the CMPAR1 register does not match the counter value, and no capture completion event has occurred.</li> <li>0b1: The value in the CMPAR1 register matches the counter value, or a capture completion event has occurred.</li> </ul> | RW     |

## 24.4.11 Buffer Control Register (TMRA\_BCONRm) (m=1-2)

Offset: 0x00B8+0x8\*m

| Offset | Bit Position |    |    |    |    |    |   |   |   |   |   |   |      |      |      |     |
|--------|--------------|----|----|----|----|----|---|---|---|---|---|---|------|------|------|-----|
| -      | 15           | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3    | 2    | 1    | 0   |
| Reset  | 0x0000       |    |    |    |    |    |   |   |   |   |   |   |      |      |      |     |
| Name   | Reserved     |    |    |    |    |    |   |   |   |   |   |   | BSEN | BSE1 | BSE0 | BEN |
| Access |              |    |    |    |    |    |   |   |   |   |   |   | RW   | RW   | RW   | RW  |

| Bit  | Name     | Function                                      | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | Access |
|------|----------|-----------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 15:4 | Reserved | Reserved                                      | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | -      |
| 3    | BSEN     | Buffer Special Enabled                        | <ul style="list-style-type: none"> <li>0b0: No buffer value transmission occurs when reading the CNTER register of the symmetric unit.</li> <li>0b1: Buffer value transmission occurs when reading the CNTER register of the symmetric unit, i.e., CMPARm -&gt; CMPARn (where m=2, 4 and n=1, 3).</li> </ul>                                                                                                                                                                                                                                 | RW     |
| 2    | BSE1     | Triangle Wave Buffer Transmission Selection 1 | <ul style="list-style-type: none"> <li>0b0: In triangle wave counting mode, the buffer value is not transmitted when the counter reaches the valley point.</li> <li>0b1: In triangle wave counting mode, the buffer value is transmitted when the counter reaches the valley point, that is: CMMARm -&gt; CMPARn (m=2, 4, n=1, 3)</li> </ul> <div>  <b>Note</b><br/>           The configuration of this bit is enable only when BEN is 1.         </div> | RW     |
| 1    | BSE0     | Triangle Wave Buffer Transmission Selection 0 | <ul style="list-style-type: none"> <li>0b0: In triangle wave counting mode, the buffer value is not transmitted when the counter reaches the peak point.</li> <li>0b1: In triangle wave counting mode, the buffer value is transmitted when the counter reaches the peak point, that is: CMMARm -&gt; CMPARn (m=2, 4, n=1, 3)</li> </ul> <div>  <b>Note</b><br/>           The configuration of this bit is enable only when BEN is 1.         </div>     | RW     |
| 0    | BEN      | Buffer enabled                                | <ul style="list-style-type: none"> <li>0b0: The buffer function for the CMPARn reference value is disabled (n=1, 3).</li> <li>0b1: The buffer function for the CMPARn reference value is enabled (n=1, 3).</li> </ul>                                                                                                                                                                                                                                                                                                                        | RW     |

**Note**

- Please do not set both BEN and BSEN to 1 simultaneously.
- When BEN is 1 and BSE1 and BSE0 are both 0, in hardware counting mode or software counting sawtooth mode, when the count reaches the overflow point or underflow point or the hardware clear counter action occurs, the buffer value is transmitted.

### 24.4.12 Capture Control Register (TMRA\_CCONRm) (m=1-4)

Offset: 0x00FC+0x4\*m

| Offset | Bit Position |          |    |          |       |       |       |       |          |       |       |       |          |   |   |       |
|--------|--------------|----------|----|----------|-------|-------|-------|-------|----------|-------|-------|-------|----------|---|---|-------|
| -      | 15           | 14       | 13 | 12       | 11    | 10    | 9     | 8     | 7        | 6     | 5     | 4     | 3        | 2 | 1 | 0     |
| Reset  | 0x0000       |          |    |          |       |       |       |       |          |       |       |       |          |   |   |       |
| Name   | Reserved     | NOFICKCP |    | NOFIENCP | HICP6 | HICP5 | HICP4 | HICP3 | Reserved | HICP2 | HICP1 | HICP0 | Reserved |   |   | CAPMD |
| Access |              | RW       |    | RW       | RW    | RW    | RW    | RW    |          | RW    | RW    | RW    |          |   |   | RW    |

| Bit   | Name     | Function                                     | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | Access |
|-------|----------|----------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 15    | Reserved | Reserved                                     | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | -      |
| 14:13 | NOFICKCP | Filter Sampling Reference Clock Selection CP | <ul style="list-style-type: none"> <li>0b00: PCLK</li> <li>0b01: PCLK/4</li> <li>0b10: PCLK/16</li> <li>0b11: PCLK/64</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | RW     |
| 12    | NOFIENCP | Capture Input Port Filtering CP              | <ul style="list-style-type: none"> <li>0b0: TMRA_&lt;t&gt;_PWMm port input filter function is disabled</li> <li>0b1: TMRA_&lt;t&gt;_PWMm port input filter function is enabled</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                              | RW     |
| 11    | HICP6    | Capture Input Condition Enable 6             | <ul style="list-style-type: none"> <li>0b0: When the falling edge is sampled on the TMRA_&lt;t&gt;_XOR port, no capture input action occurs</li> <li>0b1: When the falling edge is sampled on the TMRA_&lt;t&gt;_XOR port, a capture input action occurs</li> </ul> <div>  <b>Note</b><br/>           This bit is only valid in the CCONR4 register. That is, after this bit is enable and the corresponding event occurs, under the condition of CCONR4.CAPMD=1, the current counter value is captured and saved to CMPAR4, and STFLR.CMPF4 is set.         </div> | RW     |
| 10    | HICP5    | Capture Input Condition Enable 5             | <ul style="list-style-type: none"> <li>0b0: When the rising edge is sampled on the TMRA_&lt;t&gt;_XOR port, no capture input action occurs</li> <li>0b1: When the rising edge is sampled on the TMRA_&lt;t&gt;_XOR port, a capture input action occurs</li> </ul> <div>  <b>Note</b><br/>           This bit is only valid in the CCONR4 register. That is, after this bit is enable and the corresponding event occurs, under the condition of CCONR4.CAPMD=1, the current counter value is captured and saved to CMPAR4, and STFLR.CMPF4 is set.         </div>   | RW     |

| Bit | Name            | Function                         | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | Access |
|-----|-----------------|----------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 9   | HICP4           | Capture Input Condition Enable 4 | <ul style="list-style-type: none"> <li>0b0: When the falling edge is sampled on the TMRA_&lt;t&gt;_TRIG port input, no capture input action occurs</li> <li>0b1: When the falling edge is sampled on the TMRA_&lt;t&gt;_TRIG port input, a capture input action occurs</li> </ul> <div>  <b>Note</b><br/>           This bit is only valid in the CCONR3 register. That is, after this bit is enable and the corresponding event occurs, under the condition of CCONR3.CAPMD=1, the current counter value is captured and saved to CMPAR3, and STFLR.CMPF3 is set.         </div> | RW     |
| 8   | HICP3           | Capture Input Condition Enable 3 | <ul style="list-style-type: none"> <li>0b0: When the rising edge is sampled on the TMRA_&lt;t&gt;_TRIG port input, no capture input action occurs</li> <li>0b1: When the rising edge is sampled on the TMRA_&lt;t&gt;_TRIG port input, a capture input action occurs</li> </ul> <div>  <b>Note</b><br/>           This bit is only valid in the CCONR3 register. That is, after this bit is enable and the corresponding event occurs, under the condition of CCONR3.CAPMD=1, the current counter value is captured and saved to CMPAR3, and STFLR.CMPF3 is set.         </div>   | RW     |
| 7   | <i>Reserved</i> | <i>Reserved</i>                  | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | -      |
| 6   | HICP2           | Capture Input Condition Enable 2 | <ul style="list-style-type: none"> <li>0b0: When the event specified in the TMRA_TRGSEL register occurs, no capture input action occurs</li> <li>0b1: When the event specified in the TMRA_TRGSEL register occurs, a capture input action occurs</li> </ul> <div>  <b>Note</b><br/>           For more information, please refer to "TimerA Hardware Trigger Event Selection Register (TMRA_TRGSELx) (x=0-3) " in "Automatic Operation System (AOS) ".         </div>                                                                                                           | RW     |
| 5   | HICP1           | Capture Input Condition Enable 1 | <ul style="list-style-type: none"> <li>0b0: When the falling edge is sampled on the TMRA_&lt;t&gt;_PWMm port input, no capture input action occurs</li> <li>0b1: When the falling edge is sampled on the TMRA_&lt;t&gt;_PWMm port input, a capture input action occurs</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                  | RW     |

| Bit | Name            | Function                         | Description                                                                                                                                                                                                                                                                     | Access |
|-----|-----------------|----------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 4   | HICP0           | Capture Input Condition Enable 0 | <ul style="list-style-type: none"> <li>0b0: When the rising edge is sampled on the TMRA_&lt;t&gt;_PWMm port input, no capture input action occurs</li> <li>0b1: When the rising edge is sampled on the TMRA_&lt;t&gt;_PWMm port input, a capture input action occurs</li> </ul> | RW     |
| 3:1 | <i>Reserved</i> | <i>Reserved</i>                  | -                                                                                                                                                                                                                                                                               | -      |
| 0   | CAPMD           | Function Mode Selection          | <ul style="list-style-type: none"> <li>0b0: Compare output function</li> <li>0b1: Capture input function</li> </ul>                                                                                                                                                             | RW     |

**Note**

- For TimerA Units 1 to 4, the PCLK source is PCLK0; for TimerA Units 5 to 12, the PCLK source is PCLK1.
- The TMRA\_<t>\_XOR signal is the signal obtained by performing an XOR operation on the quadrature inputs AIN and BIN of each unit.



### 24.4.13 Port Control Register (TMRA\_PCONRm) (m=1-4)

Offset: 0x013C+0x4\*m

| Offset | Bit Position |    |    |       |          |    |      |      |   |      |   |      |   |      |   |   |
|--------|--------------|----|----|-------|----------|----|------|------|---|------|---|------|---|------|---|---|
| -      | 15           | 14 | 13 | 12    | 11       | 10 | 9    | 8    | 7 | 6    | 5 | 4    | 3 | 2    | 1 | 0 |
| Reset  | 0x0000       |    |    |       |          |    |      |      |   |      |   |      |   |      |   |   |
| Name   | Reserved     |    |    | OUTEN | Reserved |    | FORC | PERC |   | CMPC |   | STPC |   | STAC |   |   |
| Access |              |    |    | RW    |          |    | RW   | RW   |   | RW   |   | RW   |   | RW   |   |   |

| Bit   | Name     | Function                   | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | Access |
|-------|----------|----------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 15:13 | Reserved | Reserved                   | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | -      |
| 12    | OUTEN    | Output Enable              | <ul style="list-style-type: none"> <li>0b0: Output disabled for TMRA_&lt;t&gt;_PWMm port in PWM output function mode</li> <li>0b1: Output enabled for TMRA_&lt;t&gt;_PWMm port in PWM output function mode</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | RW     |
| 11:10 | Reserved | Reserved                   | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | -      |
| 9:8   | FORC     | Forced Port Status Setting | <ul style="list-style-type: none"> <li>0b0x: Setting prohibited</li> <li>0b10: TMRA_&lt;t&gt;_PWMm port output will be set to low level at the beginning of next period</li> <li>0b11: TMRA_&lt;t&gt;_PWMm port output will be set to high level at the beginning of next period</li> </ul> <div>  <b>Note</b><br/>           The "next period" refers to the point when the counter reaches overflow or underflow point in hardware counting mode or sawtooth wave counting, or when the counter reaches trough point in triangular wave counting.<br/><br/>           This bit can be used to achieve control of PWM output duty cycle 0% or 100%.         </div> | RW     |

| Bit | Name | Function                               | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | Access |
|-----|------|----------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 7:6 | PERC | Port Status Setting at Period Match    | <ul style="list-style-type: none"> <li>● 0b00: When the counter value equals the period value, the TMRA_&lt;t&gt;_PWMM output is set to low level.</li> <li>● 0b01: When the counter value equals the period value, the TMRA_&lt;t&gt;_PWMM output is set to high level.</li> <li>● 0b10: When the counter value equals the period value, the TMRA_&lt;t&gt;_PWMM output retains the previous state.</li> <li>● 0b11: When the counter value equals the period value, the TMRA_&lt;t&gt;_PWMM output toggles.</li> </ul> <div>  <b>Note</b><br/>           In sawtooth wave counting mode with up-counting, the period point is PERAR. In sawtooth wave counting mode with down-counting and triangle wave mode, the period point is zero.         </div> | RW     |
| 5:4 | CMPC | Port Status Setting at Compare Match   | <ul style="list-style-type: none"> <li>● 0b00: When counter value equals CMPARn, the TMRA_&lt;t&gt;_PWMM output is set to low level.</li> <li>● 0b01: When counter value equals CMPARn, the TMRA_&lt;t&gt;_PWMM output is set to high level.</li> <li>● 0b10: When counter value equals CMPARn, the TMRA_&lt;t&gt;_PWMM output retains the previous state.</li> <li>● 0b11: When counter value equals CMPARn, the TMRA_&lt;t&gt;_PWMM output toggles.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                           | RW     |
| 3:2 | STPC | Port Status Setting at Counting Stops  | <ul style="list-style-type: none"> <li>● 0b00: When counting stops, the TMRA_&lt;t&gt;_PWMM output is set to low level.</li> <li>● 0b01: When counting stops, the TMRA_&lt;t&gt;_PWMM output is set to high level.</li> <li>● 0b10: When counting stops, the TMRA_&lt;t&gt;_PWMM output retains the previous state.</li> <li>● 0b11: When counting stops, the TMRA_&lt;t&gt;_PWMM output retains the previous state.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                            | RW     |
| 1:0 | STAC | Port Status Setting at Counting Starts | <ul style="list-style-type: none"> <li>● 0b00: When counting starts, the TMRA_&lt;t&gt;_PWMM output is set to low level.</li> <li>● 0b01: When counting starts, the TMRA_&lt;t&gt;_PWMM output is set to high level.</li> <li>● 0b10: When counting starts, the TMRA_&lt;t&gt;_PWMM output retains the previous state.</li> <li>● 0b11: When counting starts, the TMRA_&lt;t&gt;_PWMM output retains the previous state.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                        | RW     |

## 24.4.14 Hardware Trigger Event Selection Register (TMRA\_HCONR)

| Offset | Bit Position |       |       |       |          |       |       |       |          |       |       |       |          |       |       |       |
|--------|--------------|-------|-------|-------|----------|-------|-------|-------|----------|-------|-------|-------|----------|-------|-------|-------|
| 0x0084 | 15           | 14    | 13    | 12    | 11       | 10    | 9     | 8     | 7        | 6     | 5     | 4     | 3        | 2     | 1     | 0     |
| Reset  | 0x0000       |       |       |       |          |       |       |       |          |       |       |       |          |       |       |       |
| Name   | HCLE6        | HCLE5 | HCLE4 | HCLE3 | Reserved | HCLE2 | HCLE1 | HCLE0 | Reserved | HSTP2 | HSTP1 | HSTP0 | Reserved | HSTA2 | HSTA1 | HSTA0 |
| Access | RW           | RW    | RW    | RW    |          | RW    | RW    | RW    |          | RW    | RW    | RW    |          | RW    | RW    | RW    |

| Bit | Name     | Function                   | Description                                                                                                                                                                                                                                              | Access |
|-----|----------|----------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 15  | HCLE6    | Hardware clear condition 6 | Condition: Falling edge is sampled on TMRA_<t>_PWM3 port<br><ul style="list-style-type: none"> <li>0b0: When the condition is met, hardware clear is disabled</li> <li>0b1: When the condition is met, hardware clear is enabled</li> </ul>              | RW     |
| 14  | HCLE5    | Hardware clear condition 5 | Condition: Rising edge is sampled on TMRA_<t>_PWM3 port<br><ul style="list-style-type: none"> <li>0b0: When the condition is met, hardware clear is disabled</li> <li>0b1: When the condition is met, hardware clear is enabled</li> </ul>               | RW     |
| 13  | HCLE4    | Hardware clear condition 4 | Condition: Falling edge is sampled on TRIG port of the symmetric unit<br><ul style="list-style-type: none"> <li>0b0: When the condition is met, hardware clear is disabled</li> <li>0b1: When the condition is met, hardware clear is enabled</li> </ul> | RW     |
| 12  | HCLE3    | Hardware clear condition 3 | Condition: Rising edge is sampled on TRIG port of the symmetric unit<br><ul style="list-style-type: none"> <li>0b0: When the condition is met, hardware clear is disabled</li> <li>0b1: When the condition is met, hardware clear is enabled</li> </ul>  | RW     |
| 11  | Reserved | Reserved                   | -                                                                                                                                                                                                                                                        | -      |

| Bit | Name            | Function                   | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | Access |
|-----|-----------------|----------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 10  | HCLE2           | Hardware clear condition 2 | <p>Condition: The event specified in the TMRA_TRGSEL register occurs</p> <ul style="list-style-type: none"> <li>● 0b0: When the condition is met, hardware clear is disabled</li> <li>● 0b1: When the condition is met, hardware clear is enabled</li> </ul> <div>  <b>Note</b><br/>           For more information, please refer to "Time-rA Hardware Trigger Event Selection Register (TMRA_TRGSELx) (x=0-3) " in "Automatic Operation System (AOS) ".         </div> | RW     |
| 9   | HCLE1           | Hardware clear condition 1 | <p>Condition: Falling edge is sampled on TMRA_&lt;t&gt;_TRIG port</p> <ul style="list-style-type: none"> <li>● 0b0: When the condition is met, hardware clear is disabled</li> <li>● 0b1: When the condition is met, hardware clear is enabled</li> </ul>                                                                                                                                                                                                                                                                                                | RW     |
| 8   | HCLE0           | Hardware clear condition 0 | <p>Condition: Rising edge is sampled on TMRA_&lt;t&gt;_TRIG port</p> <ul style="list-style-type: none"> <li>● 0b0: When the condition is met, hardware clear is disabled</li> <li>● 0b1: When the condition is met, hardware clear is enabled</li> </ul>                                                                                                                                                                                                                                                                                                 | RW     |
| 7   | <i>Reserved</i> | <i>Reserved</i>            | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | -      |
| 6   | HSTP2           | Hardware stop condition 2  | <p>Condition: The event specified in the TMRA_TRGSEL register occurs</p> <ul style="list-style-type: none"> <li>● 0b0: When the condition is met, hardware stop is disabled</li> <li>● 0b1: When the condition is met, hardware stop is enabled</li> </ul> <div>  <b>Note</b><br/>           For more information, please refer to "Time-rA Hardware Trigger Event Selection Register (TMRA_TRGSELx) (x=0-3) " in "Automatic Operation System (AOS) ".         </div> | RW     |
| 5   | HSTP1           | Hardware stop condition 1  | <p>Condition: Falling edge is sampled on TMRA_&lt;t&gt;_TRIG port</p> <ul style="list-style-type: none"> <li>● 0b0: When the condition is met, hardware stop is disabled</li> <li>● 0b1: When the condition is met, hardware stop is enabled</li> </ul>                                                                                                                                                                                                                                                                                                  | RW     |

| Bit | Name            | Function                   | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | Access |
|-----|-----------------|----------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 4   | HSTP0           | Hardware stop condition 0  | Condition: Rising edge is sampled on TMRA_<t>_TRIG port <ul style="list-style-type: none"> <li>● 0b0: When the condition is met, hardware stop is disabled</li> <li>● 0b1: When the condition is met, hardware stop is enabled</li> </ul>                                                                                                                                                                                                                                                                                                            | RW     |
| 3   | <i>Reserved</i> | <i>Reserved</i>            | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | -      |
| 2   | HSTA2           | Hardware start condition 2 | Condition: The event specified in the TMRA_TRGSEL register occurs <ul style="list-style-type: none"> <li>● 0b0: When the condition is met, hardware start is disabled</li> <li>● 0b1: When the condition is met, hardware start is enabled</li> </ul> <div>  <b>Note</b><br/> For more information, please refer to "Time-rA Hardware Trigger Event Selection Register (TMRA_TRGSELx) (x=0-3) " in "Automatic Operation System (AOS) ". </div>                      | RW     |
| 1   | HSTA1           | Hardware start condition 1 | Conditions: <ol style="list-style-type: none"> <li>1. Falling edge is sampled on TMRA_&lt;t&gt;_TRIG port when the synchronization start function is disabled</li> <li>2. Falling edge is sampled on TMRA_&lt;t&gt;_TRIG port of the symmetric unit when the synchronization start function is enabled (only even unit in the symmetric unit support)</li> </ol> <ul style="list-style-type: none"> <li>● 0b0: When the condition is met, hardware start is disabled</li> <li>● 0b1: When the condition is met, hardware start is enabled</li> </ul> | RW     |
| 0   | HSTA0           | Hardware start condition 0 | Conditions: <ol style="list-style-type: none"> <li>1. Rising edge is sampled on TMRA_&lt;t&gt;_TRIG port when the synchronization start function is disabled</li> <li>2. Rising edge is sampled on TMRA_&lt;t&gt;_TRIG port of the symmetric unit when the synchronization start function is enabled (only even unit in the symmetric unit support)</li> </ol> <ul style="list-style-type: none"> <li>● 0b0: When the condition is met, hardware start is disabled</li> <li>● 0b1: When the condition is met, hardware start is enabled</li> </ul>   | RW     |

## 24.4.15 Hardware Count Up Event Selection Register (TMRA\_HCUPR)

| Offset | Bit Position |    |    |        |        |        |       |       |       |       |       |       |       |       |       |       |
|--------|--------------|----|----|--------|--------|--------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| 0x0088 | 15           | 14 | 13 | 12     | 11     | 10     | 9     | 8     | 7     | 6     | 5     | 4     | 3     | 2     | 1     | 0     |
| Reset  | 0x0000       |    |    |        |        |        |       |       |       |       |       |       |       |       |       |       |
| Name   | Reserved     |    |    | HCUP12 | HCUP11 | HCUP10 | HCUP9 | HCUP8 | HCUP7 | HCUP6 | HCUP5 | HCUP4 | HCUP3 | HCUP2 | HCUP1 | HCUP0 |
| Access |              |    |    | RW     | RW     | RW     | RW    | RW    | RW    | RW    | RW    | RW    | RW    | RW    | RW    | RW    |

| Bit   | Name     | Function                       | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | Access |
|-------|----------|--------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 15:13 | Reserved | Reserved                       | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | -      |
| 12    | HCUP12   | Hardware Count Up Condition 12 | Condition: Count underflow occurs in the symmetric unit<br><ul style="list-style-type: none"> <li>0b0: When this condition is met, hardware count up is disabled.</li> <li>0b1: When this condition is met, hardware count up is enabled.</li> </ul>                                                                                                                                                                                                                                                                                                        | RW     |
| 11    | HCUP11   | Hardware Count Up Condition 11 | Condition: Count overflow occurs in the symmetric unit<br><ul style="list-style-type: none"> <li>0b0: When this condition is met, hardware count up is disabled.</li> <li>0b1: When this condition is met, hardware count up is enabled.</li> </ul>                                                                                                                                                                                                                                                                                                         | RW     |
| 10    | HCUP10   | Hardware Count Up Condition 10 | Condition: The event specified in the TMRA_TRGSEL register occurs<br><ul style="list-style-type: none"> <li>0b0: When this condition is met, hardware count up is disabled.</li> <li>0b1: When this condition is met, hardware count up is enabled.</li> </ul> <div>  <b>Note</b><br/>           For more information, please refer to "TimerA Hardware Trigger Event Selection Register (TMRA_TRGSELx) (x=0-3) " in "Automatic Operation System (AOS) ".         </div> | RW     |
| 9     | HCUP9    | Hardware Count Up Condition 9  | Condition: Falling edge is sampled on the TMRA_<t>_TRIG port<br><ul style="list-style-type: none"> <li>0b0: When this condition is met, hardware count up is disabled.</li> <li>0b1: When this condition is met, hardware count up is enabled.</li> </ul>                                                                                                                                                                                                                                                                                                   | RW     |

| Bit | Name  | Function                      | Description                                                                                                                                                                                                                                                                                           | Access |
|-----|-------|-------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 8   | HCUP8 | Hardware Count Up Condition 8 | Condition: Rising edge is sampled on the TMRA_<t>_TRIG port <ul style="list-style-type: none"> <li>0b0: When this condition is met, hardware count up is disabled.</li> <li>0b1: When this condition is met, hardware count up is enabled.</li> </ul>                                                 | RW     |
| 7   | HCUP7 | Hardware Count Up Condition 7 | Condition: When the TMRA_<t>_CLKB port is high level, the falling edge is sampled on the TMRA_<t>_CLKA port <ul style="list-style-type: none"> <li>0b0: When this condition is met, hardware count up is disabled.</li> <li>0b1: When this condition is met, hardware count up is enabled.</li> </ul> | RW     |
| 6   | HCUP6 | Hardware Count Up Condition 6 | Condition: When the TMRA_<t>_CLKB port is high level, the rising edge is sampled on the TMRA_<t>_CLKA port <ul style="list-style-type: none"> <li>0b0: When this condition is met, hardware count up is disabled.</li> <li>0b1: When this condition is met, hardware count up is enabled.</li> </ul>  | RW     |
| 5   | HCUP5 | Hardware Count Up Condition 5 | Condition: When the TMRA_<t>_CLKB port is low level, the falling edge is sampled on the TMRA_<t>_CLKA port <ul style="list-style-type: none"> <li>0b0: When this condition is met, hardware count up is disabled.</li> <li>0b1: When this condition is met, hardware count up is enabled.</li> </ul>  | RW     |
| 4   | HCUP4 | Hardware Count Up Condition 4 | Condition: When the TMRA_<t>_CLKB port is low level, the rising edge is sampled on the TMRA_<t>_CLKA port <ul style="list-style-type: none"> <li>0b0: When this condition is met, hardware count up is disabled.</li> <li>0b1: When this condition is met, hardware count up is enabled.</li> </ul>   | RW     |
| 3   | HCUP3 | Hardware Count Up Condition 3 | Condition: When the TMRA_<t>_CLKA port is high level, the falling edge is sampled on the TMRA_<t>_CLKB port <ul style="list-style-type: none"> <li>0b0: When this condition is met, hardware count up is disabled.</li> <li>0b1: When this condition is met, hardware count up is enabled.</li> </ul> | RW     |
| 2   | HCUP2 | Hardware Count Up Condition 2 | Condition: When the TMRA_<t>_CLKA port is high level, the rising edge is sampled on the TMRA_<t>_CLKB port <ul style="list-style-type: none"> <li>0b0: When this condition is met, hardware count up is disabled.</li> <li>0b1: When this condition is met, hardware count up is enabled.</li> </ul>  | RW     |

| Bit | Name  | Function                      | Description                                                                                                                                                                                                                                                                                              | Access |
|-----|-------|-------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 1   | HCUP1 | Hardware Count Up Condition 1 | Condition: When the TMRA_<t>_CLKA port is low level, the falling edge is sampled on the TMRA_<t>_CLKB port <ul style="list-style-type: none"> <li>● 0b0: When this condition is met, hardware count up is disabled.</li> <li>● 0b1: When this condition is met, hardware count up is enabled.</li> </ul> | RW     |
| 0   | HCUP0 | Hardware Count Up Condition 0 | Condition: When the TMRA_<t>_CLKA port is low level, the rising edge is sampled on the TMRA_<t>_CLKB port <ul style="list-style-type: none"> <li>● 0b0: When this condition is met, hardware count up is disabled.</li> <li>● 0b1: When this condition is met, hardware count up is enabled.</li> </ul>  | RW     |



## 24.4.16 Hardware Count Down Event Selection Register (TMRA\_HCDOR)

| Offset | Bit Position |    |    |        |        |        |       |       |       |       |       |       |       |       |       |       |
|--------|--------------|----|----|--------|--------|--------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| 0x008C | 15           | 14 | 13 | 12     | 11     | 10     | 9     | 8     | 7     | 6     | 5     | 4     | 3     | 2     | 1     | 0     |
| Reset  | 0x0000       |    |    |        |        |        |       |       |       |       |       |       |       |       |       |       |
| Name   | Reserved     |    |    | HCD012 | HCD011 | HCD010 | HCD09 | HCD08 | HCD07 | HCD06 | HCD05 | HCD04 | HCD03 | HCD02 | HCD01 | HCD00 |
| Access |              |    |    | RW     | RW     | RW     | RW    | RW    | RW    | RW    | RW    | RW    | RW    | RW    | RW    | RW    |

| Bit   | Name     | Function                         | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | Access |
|-------|----------|----------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 15:13 | Reserved | Reserved                         | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | -      |
| 12    | HCD012   | Hardware Count Down Condition 12 | Condition: Count underflow occurs in the symmetric unit<br><ul style="list-style-type: none"> <li>0b0: Hardware count up is disabled when the condition is met.</li> <li>0b1: Hardware count up is enabled when the condition is met.</li> </ul>                                                                                                                                                                                                                                                                                                         | RW     |
| 11    | HCD011   | Hardware Count Down Condition 11 | Condition: Count overflow occurs in the symmetric unit<br><ul style="list-style-type: none"> <li>0b0: Hardware count up is disabled when the condition is met.</li> <li>0b1: Hardware count up is enabled when the condition is met.</li> </ul>                                                                                                                                                                                                                                                                                                          | RW     |
| 10    | HCD010   | Hardware Count Down Condition 10 | Condition: The event specified in the TMRA_TRGSEL register occurs<br><ul style="list-style-type: none"> <li>0b0: Hardware count up is disabled when the condition is met.</li> <li>0b1: Hardware count up is enabled when the condition is met.</li> </ul> <div>  <b>Note</b><br/>           For more information, please refer to "Time-rA Hardware Trigger Event Selection Register (TMRA_TRGSELx) (x=0-3) " in "Automatic Operation System (AOS) ".         </div> | RW     |
| 9     | HCD09    | Hardware Count Down Condition 9  | Condition: Falling edge is sampled on the TMRA_<t>_TRIG port<br><ul style="list-style-type: none"> <li>0b0: Hardware count up is disabled when the condition is met.</li> <li>0b1: Hardware count up is enabled when the condition is met.</li> </ul>                                                                                                                                                                                                                                                                                                    | RW     |

| Bit | Name  | Function                        | Description                                                                                                                                                                                                                                                                                              | Access |
|-----|-------|---------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 8   | HCDO8 | Hardware Count Down Condition 8 | Condition: Rising edge is sampled on the TMRA_<t>_TRIG port<br><ul style="list-style-type: none"> <li>● 0b0: Hardware count up is disabled when the condition is met.</li> <li>● 0b1: Hardware count up is enabled when the condition is met.</li> </ul>                                                 | RW     |
| 7   | HCDO7 | Hardware Count Down Condition 7 | Condition: When the TMRA_<t>_CLKB port is high level, the falling edge is sampled on the TMRA_<t>_CLKA port<br><ul style="list-style-type: none"> <li>● 0b0: Hardware count up is disabled when the condition is met.</li> <li>● 0b1: Hardware count up is enabled when the condition is met.</li> </ul> | RW     |
| 6   | HCDO6 | Hardware Count Down Condition 6 | Condition: When the TMRA_<t>_CLKB port is high level, the rising edge is sampled on the TMRA_<t>_CLKA port<br><ul style="list-style-type: none"> <li>● 0b0: Hardware count up is disabled when the condition is met.</li> <li>● 0b1: Hardware count up is enabled when the condition is met.</li> </ul>  | RW     |
| 5   | HCDO5 | Hardware Count Down Condition 5 | Condition: When the TMRA_<t>_CLKB port is low level, the falling edge is sampled on the TMRA_<t>_CLKA port<br><ul style="list-style-type: none"> <li>● 0b0: Hardware count up is disabled when the condition is met.</li> <li>● 0b1: Hardware count up is enabled when the condition is met.</li> </ul>  | RW     |
| 4   | HCDO4 | Hardware Count Down Condition 4 | Condition: When the TMRA_<t>_CLKB port is low level, the rising edge is sampled on the TMRA_<t>_CLKA port<br><ul style="list-style-type: none"> <li>● 0b0: Hardware count up is disabled when the condition is met.</li> <li>● 0b1: Hardware count up is enabled when the condition is met.</li> </ul>   | RW     |
| 3   | HCDO3 | Hardware Count Down Condition 3 | Condition: When the TMRA_<t>_CLKA port is high level, the falling edge is sampled on the TMRA_<t>_CLKB port<br><ul style="list-style-type: none"> <li>● 0b0: Hardware count up is disabled when the condition is met.</li> <li>● 0b1: Hardware count up is enabled when the condition is met.</li> </ul> | RW     |
| 2   | HCDO2 | Hardware Count Down Condition 2 | Condition: When the TMRA_<t>_CLKA port is high level, the rising edge is sampled on the TMRA_<t>_CLKB port<br><ul style="list-style-type: none"> <li>● 0b0: Hardware count up is disabled when the condition is met.</li> <li>● 0b1: Hardware count up is enabled when the condition is met.</li> </ul>  | RW     |

| Bit | Name  | Function                        | Description                                                                                                                                                                                                                                                                                          | Access |
|-----|-------|---------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 1   | HCDO1 | Hardware Count Down Condition 1 | Condition: When the TMRA_<t>_CLKA port is low level, the falling edge is sampled on the TMRA_<t>_CLKB port <ul style="list-style-type: none"> <li>● 0b0: Hardware count up is disabled when the condition is met.</li> <li>● 0b1: Hardware count up is enabled when the condition is met.</li> </ul> | RW     |
| 0   | HCDO0 | Hardware Count Down Condition 0 | Condition: When the TMRA_<t>_CLKA port is low level, the rising edge is sampled on the TMRA_<t>_CLKB port <ul style="list-style-type: none"> <li>● 0b0: Hardware count up is disabled when the condition is met.</li> <li>● 0b1: Hardware count up is enabled when the condition is met.</li> </ul>  | RW     |

## 25 General Timer (Timer2)

### 25.1 Introduction

General Timer 2 (Timer2) is a basic timer that can realize synchronous counting and asynchronous counting. The timer contains two channels (CH-A and CH-B). Each channel has an output port, which can realize basic square wave output. Each channel has two input ports, one is a clock input port, which can realize asynchronous port counting; the other one is a trigger input port, which can realize timer start, stop, clear, counting action and counting value capture input; support pulse width measurement and period measurement. This series of products is equipped with 4 units of Timer2.

### 25.2 Features

The basic functions and features of Timer2 are as follows:

Table 25-1 Basic Functions and Features of Timer2

|                |                                                                         |
|----------------|-------------------------------------------------------------------------|
| Basic actions  | Hardware triggered counter start, stop, clear, count and capture inputs |
|                | Synchronous counting and asynchronous counting                          |
|                | Pulse width measurement                                                 |
|                | Period measurement                                                      |
|                | Square wave output                                                      |
|                | Event signal output                                                     |
| Interrupt type | Count comparison match interrupt                                        |
|                | Count overflow interrupt                                                |

### 25.3 Functional Description

#### 25.3.1 Basic Block Diagram

In the block diagram, the symbol "<t>" denotes the unit number, which ranges from 1-4. In the remainder of this section, "<t>" will consistently refer to the unit number and will not be further explained. The ANF on the TMR2\_<t>\_CLKA/B port input in the block diagram is an analog filter unit with a filter width of 40ns (typical conditions).

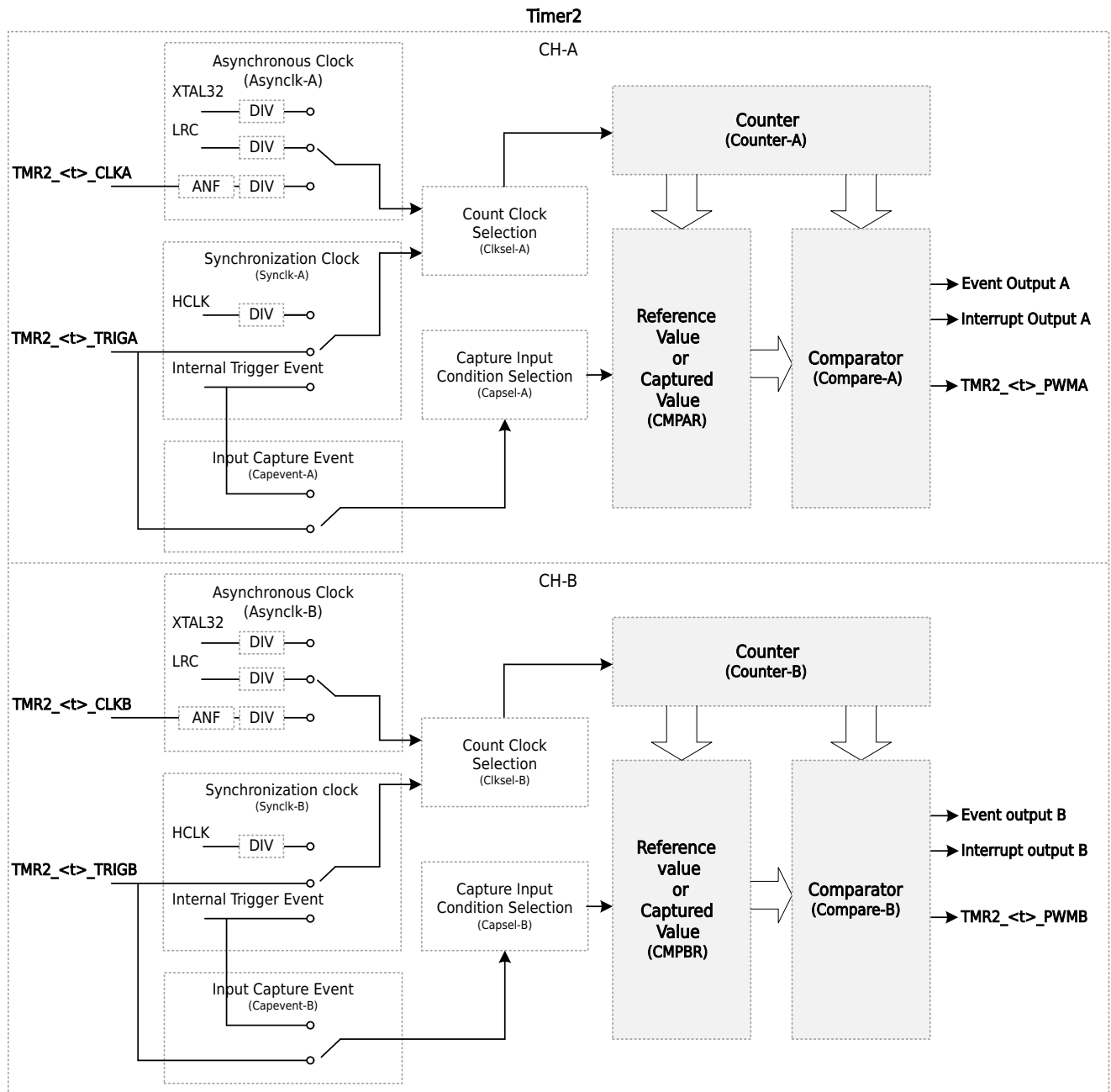


Figure 25-1 Timer2 Basic Block Diagram

The following table lists the input and output ports of Timer2.

Table 25-2 Timer2 Port List

| Port Name        | Direction | Function                                                                  |
|------------------|-----------|---------------------------------------------------------------------------|
| TMR2_<t>_CLKA/B  | Input     | Asynchronous clock input port                                             |
| TMR2_<t>_TRIGA/B | Input     | 1. Hardware start, stop, clear, count input port<br>2. Capture input port |
| TMR2_<t>_PWMA/B  | Output    | Square wave output port                                                   |

### 25.3.2 Clock Source Selection

Timer2 can be counted either synchronously or asynchronously.

The synchronous counting mode refers to the synchronous timing relationship between the timer count clock and the bus access clock (register read/write operation clock). Asynchronous counting means that the timer count clock and bus access clock (register read/write operation clock) are non-synchronous timing relationships. The status of a timer, etc., may be changing when the register is read by asynchronous counting. Therefore, in the asynchronous counting mode, the register read operation must be implemented in the count stopped state.

### Synchronous Count Timer

When operating in synchronous counting mode (BCONR.SYNSA<B>=0), the following clock sources can be selected (configured via BCONR.SYNCLKA<B>[1:0] or BCONR.SYNCLKA<B>T[1:0]) :

1. PCLK1 and its 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024 divisions are used as synchronous count clock (BCONR.SYNCLKA<B>[1:0]=0b00, determined by BCONR.CKDIVA<B>[3:0]);
2. The input enable selection edge of TMR2\_<t>\_TRIGA and TMR2\_<t>\_TRIGB ports is sampled by the internal PCLK1 as the synchronous counting clock (BCONR.SYNCLKA<B>[1:0]=0b01 or 0b10);
3. An internal hardware trigger event input can serve as the synchronous count clock (BCONR.SYNCLKA<B>[1:0]=0b11);
4. Timer6 count overflow event (BCONR.SYNCLKA<B>T[0]=1);
5. Timer6 count underflow event (BCONR.SYNCLKA<B>T[1]=1).



#### Note

When the clock source is selected as 2, 3, 4, or 5, ensure that the PCLK1 division frequency is 1.

### Asynchronous Count Timer

In asynchronous counting mode (BCONR.SYNSA<B>=1), the clock source can be selected from the following options (BCONR.ASYNCLKA<B>[1:0] setting) :

1. LRC clock source input and its 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024 divisions are used as asynchronous counting clocks (BCONR.ASYNCLKA<B>=0 & BCONR.CKDIVA<B>[3:0] settings);
2. XTAL32 clock source input and its 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024 divisions are used as asynchronous counting clocks (BCONR.ASYNCLKA<B>=1 & BCONR.CKDIVA<B>[3:0] settings);
3. TMR2\_<t>\_CLKA, TMR2\_<t>\_CLKB clock input ports and their 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024 divisions are used as asynchronous count clocks (BCONR.ASYNCLKA<B>=0b10 & BCONR.CKDIVA<B>[3:0] settings).

### 25.3.3 Compare Output

Timer2 has two output ports (TMR2\_<t>\_PWMA, TMR2\_<t>\_PWMB), which can realize basic square wave output.

The CMPAR and CMPBR registers correspond to the count compare references of TMR2\_<t>\_PWMA and TMR2\_<t>\_PWMB. When the count values CNTAR and CMPAR of the timer are equal, the TMR2\_<t>\_PWMA port outputs the specified level; when the count values CNTBR and CMPBR of the timer are equal, the TMR2\_<t>\_PWMB port outputs the specified level.

The count start level, stop level, and count comparison match level of the TMR2\_<t>\_PWMA/B port can be set by the PCONR.STACA<B>[1:0], PCONR.STPCA<B>[1:0], and PCONR.CMPCA<B>[1:0]. The figure below illustrates an example of the compare output behavior.

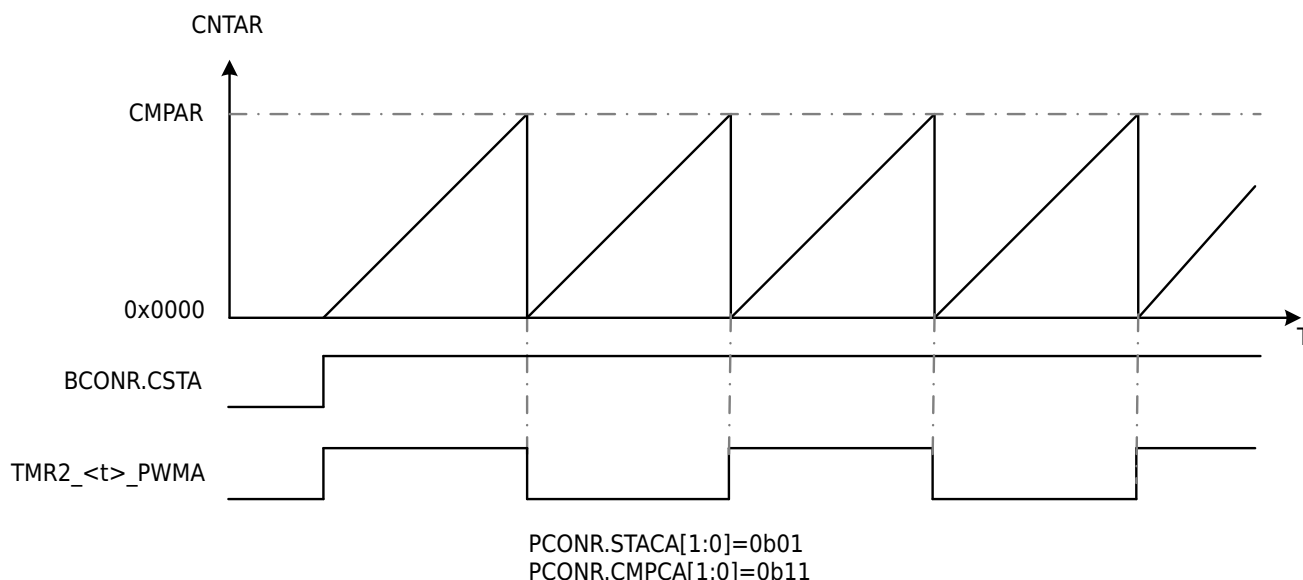


Figure 25-2 Compare Output Action

## 25.3.4 Hardware Triggering

### 25.3.4.1 Functional Overview

There are two types of hardware trigger actions for Timer2, one is the input event of TMR2\_<t>\_TRIGA/B port, and the other is the internal hardware trigger event. Both types of events can initiate actions such as start, stop, clear, and hardware capture input function. Various combinations of settings allow for the implementation of pulse width measurement, period measurement, and other functionalities.

The TMR2\_<t>\_TRIGA/B port input type selection (rising edge or falling edge) and function selection (start, stop, clear, capture input), and the function selection of the internal hardware trigger event (start, stop, clear, capture input) can be set through the corresponding bits of the hardware control register (HCONR).

Timer2 has an internal hardware trigger source. The event source can be selected by setting the corresponding number in the trigger selection register (TMR2\_TRGSEL). For specific event correspondence, please refer to the relevant description in the "Interrupt Controller (INTC)" chapter. When using the internal hardware trigger function, it is necessary to first enable the peripheral circuit trigger function at position 1 of the function clock control 0 register (PWC\_FCG0).

### 25.3.4.2 Start/Stop/Clear

The hardware start, stop, and clear conditions of each channel are determined by the relevant settings of HCONR register. The following figure is an example of the hardware start and clear action.

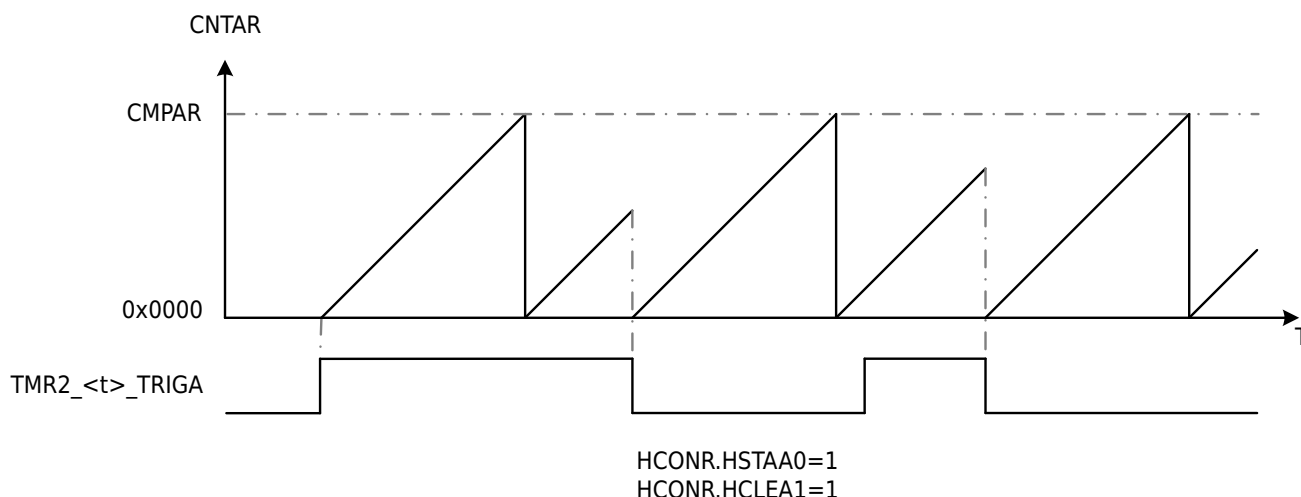


Figure 25-3 Hardware Start and Clear Action

### 25.3.4.3 Capture Input Action

When the TMR2\_<t>\_TRIGA/B port input or internal trigger Event is configured as the capture input condition, the capture input function can be implemented. When the capture input condition selected by the HCONR register is enabled, the current counter value is saved into the corresponding registers (CMPAR, CMPBR). The following figure shows an example of the capture input action.

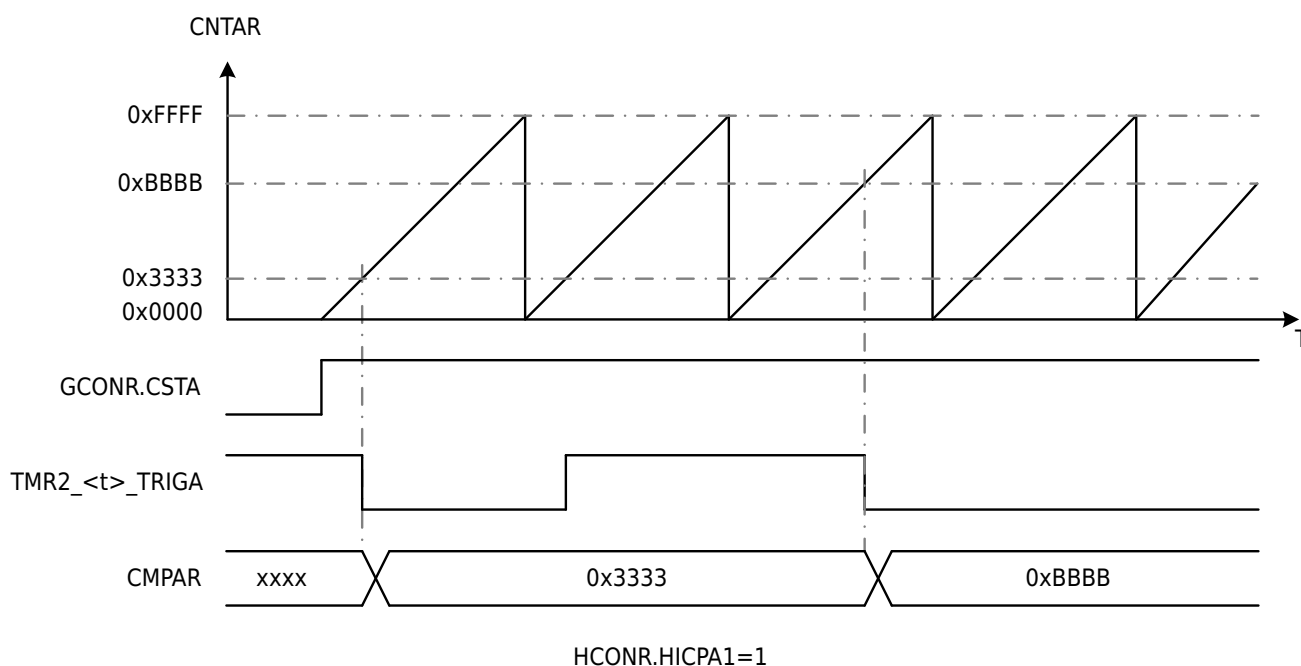


Figure 25-4 Capture Input Action

### 25.3.4.4 Pulse Width Measurement

Set the hardware start condition for channel A to the rising edge of TMR2\_<t>\_TRIGA, and set the hardware clear condition, stop condition, and capture input condition to the falling edge of TMR2\_<t>\_TRIGA to achieve continuous pulse width measurement. Channel B can achieve the same function. The corresponding operation is illustrated in the figure below.



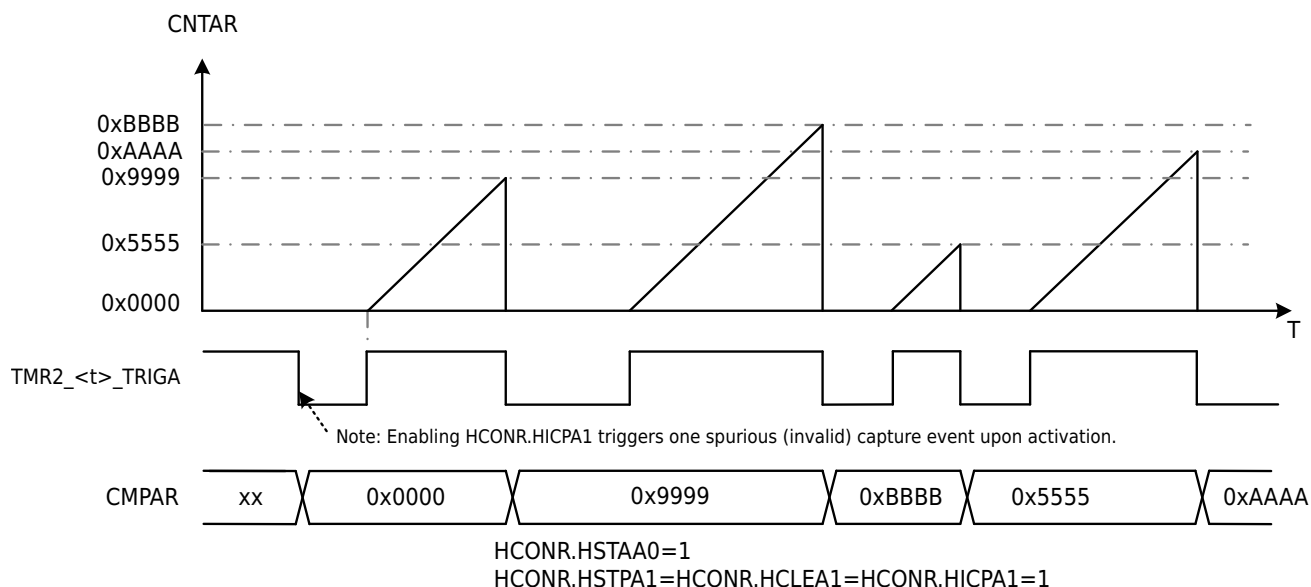


Figure 25-5 Pulse Width Measurement

### 25.3.4.5 Period Measurement

By setting the hardware start condition, hardware clear condition, and capture input condition of channel A to the same edge (rising or falling edge) of TMR2\_<t>\_TRIGA, continuous period width measurement can be achieved. Channel B can achieve the same function. The corresponding operation is illustrated in the figure below.

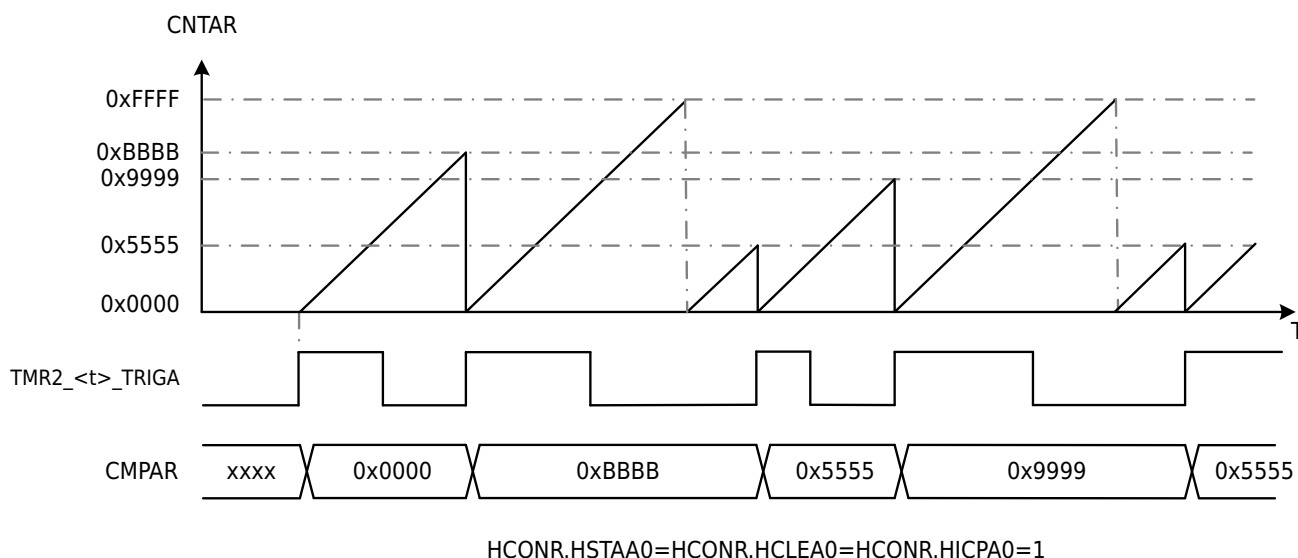


Figure 25-6 Period Measurement

### 25.3.5 Digital Filtering

The TRIG input port of Timer2 (TMR2\_<t>\_TRIGA/B) has a digital filter function. The filtering function of the corresponding port can be enabled by PCONR.NOFIENA<B> register. The reference clock for filtering is also set by the port control register (PCONR.NOFICKA<B>[1:0]).

When the port level sampled by the filter sampling reference clock remains consistent for 3 consecutive times, that level is transmitted as a active signal to the internal block. Levels with fewer than 3 consistent samples are filtered as external noise and not transmitted to the internal block. An example of this filtering behavior is shown in the figure below.

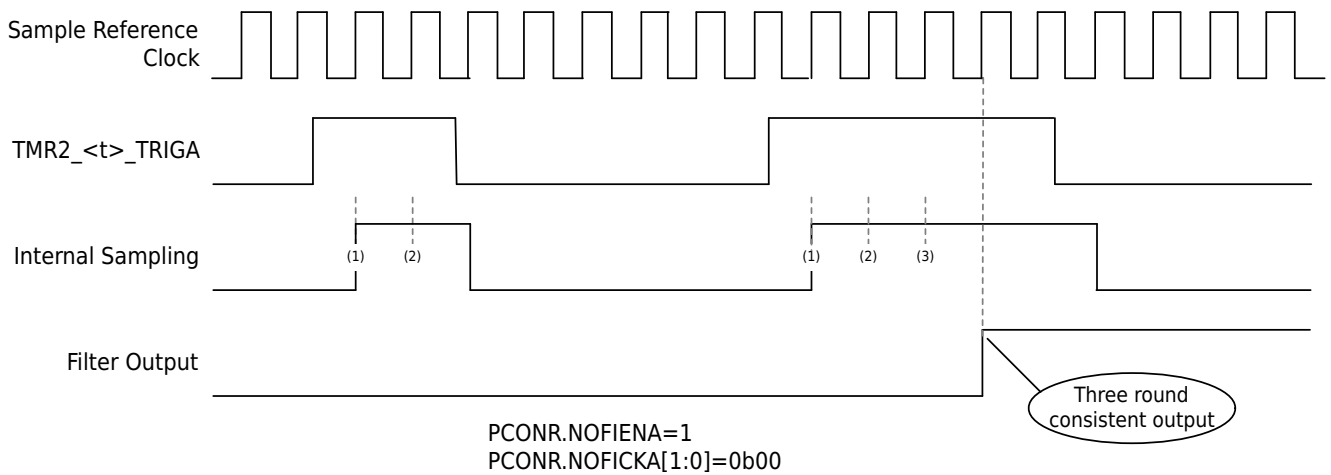


Figure 25-7 Digital Filtering of TRIG Input Port

### 25.3.6 Software Synchronization Start

The product is equipped with a count synchronization enable register (TMR\_SYNENR, TMR\_SYNENR2) to synchronously start the timer (including Timer0, Timer2, Timer4, Timer6, TimerA) carried out by the product. For usage and register description, please refer to the description in the "General Timer (Timer0)" chapter. Please do not use this function when this timer is counting asynchronously.

### 25.3.7 Interrupt and Events

#### 25.3.7.1 Interrupt Output

One Timer2 contains 4 interrupts, namely, the count compare match interrupt or capture input interrupt of channel A and channel B, and the count overflow interrupt of channel A and channel B.

There are two reference registers (CMPAR, CMPBR), which can be compared with the count value registers (CNTAR, CNTBR) to generate comparison match active signals. When the count compare matches, the STFLR.CMFA<B> will be set to 1. At this time, if the ICONR.CMENA<B> is set to enable interrupt, the corresponding interrupt request (TMR2\_<t>\_CMPm, m=A, B) will also be triggered.

When the capture input active condition selected by the hardware control register (HCONR) occurs, the corresponding capture input action can be generated. At this time, if the ICONR.CMENA<B> is set to enable interrupt, the corresponding interrupt request (TMR2\_<t>\_CMPm, m=A, B) is triggered.

When the count value register (CNTAR, CNTBR) counts to 0xFFFF, a count overflow event occurs. At this time, the STFLR.OVFA<B> will be set to 1. If the ICONR.OVENA<B> is set to enable interrupt, the corresponding interrupt request (TMR2\_<t>\_OVFm, m=A, B) will also be triggered.

When the asynchronous counting mode is selected (except BCONR.ASYNCLKA[1:0]=0b10), the compare match interrupt generated by the reference register (CMPAR) of unit 1 can be used to wake up the system in low power mode. Please refer to the relevant description in the "Interrupt Controller (INTC)" chapter.

#### 25.3.7.2 Event Output

One Timer2 contains 4 events, namely, the count compare match event or capture input interrupt of channel A and channel B, and the count overflow interrupt of channel A and channel B.

When a count compare match, capture input action or count overflow occurs during the counting process, a corresponding event request (TMR2\_<t>\_CMPm or TMR2\_<t>\_OVFm, m=A, B) output signal will be generated, which can be used to select and trigger other modules.

## 25.4 Register Description

### 25.4.1 Register List

- Timer2\_1 base address: 0x40024800
- Timer2\_2 base address: 0x40024C00
- Timer2\_3 base address: 0x40025000
- Timer2\_4 base address: 0x40025400



#### Note

Timer2 is the name of the timer, and Timer2\_<t> (<t>:1-4) is the timer unit.

Table 25-3 General Timer (Timer2) Register Offset

| Offset | Register   | Description                | Width |
|--------|------------|----------------------------|-------|
| 0x0000 | TMR2_CNTAR | Count Value Register       | 32    |
| 0x0004 | TMR2_CNTBR | Count Value Register       | 32    |
| 0x0008 | TMR2_CMPAR | Reference Value Register   | 32    |
| 0x000C | TMR2_CMPBR | Reference Value Register   | 32    |
| 0x0010 | TMR2_BCONR | Basic Control Register     | 32    |
| 0x0014 | TMR2_ICONR | Interrupt Control Register | 32    |
| 0x0018 | TMR2_PCONR | Port Control Register      | 32    |
| 0x001C | TMR2_HCONR | Hardware Control Register  | 32    |
| 0x0020 | TMR2_STFLR | Status Flag Register       | 32    |

### 25.4.2 Count Value Register m (TMR2\_CNTmR) (m=A/B)

Offset: 0x0000 (CNTAR), 0x0004 (CNTBR)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |     |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| -      | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15  | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |     |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | CNT |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RW  |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit   | Name     | Function    | Description                                                                                                                                                                                                                                                                                                                                                | Access |
|-------|----------|-------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:16 | Reserved | Reserved    | -                                                                                                                                                                                                                                                                                                                                                          | -      |
| 15:0  | CNT      | Count value | Current timer count value<br><br><div>  <b>Note</b><br/>           Writing to this register is prohibited while counting is in progress.<br/><br/>           This register must be initialized before the timer can be started again after counting stops.         </div> | RW     |

### 25.4.3 Reference Value Register m (TMR2\_CMPmR) (m=A/B)

Offset: 0x0008 (CMPAR), 0x000C (CMPBR)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |     |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| '      | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15  | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x0000FFFF   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |     |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | CMP |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RW  |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit   | Name     | Function  | Description                                                  | Access |
|-------|----------|-----------|--------------------------------------------------------------|--------|
| 31:16 | Reserved | Reserved  | -                                                            | -      |
| 15:0  | CMP      | Reference | Set the counting reference to generate a compare match event | RW     |

## 25.4.4 Basic Control Register (TMR2\_BCONR)

| Offset | Bit Position |          |    |          |    |         |    |        |    |    |    |       |    |          |        |    |      |    |          |          |    |          |   |         |   |        |   |    |   |       |   |          |        |  |      |  |    |  |
|--------|--------------|----------|----|----------|----|---------|----|--------|----|----|----|-------|----|----------|--------|----|------|----|----------|----------|----|----------|---|---------|---|--------|---|----|---|-------|---|----------|--------|--|------|--|----|--|
| 0x0010 | 31           | 30       | 29 | 28       | 27 | 26      | 25 | 24     | 23 | 22 | 21 | 20    | 19 | 18       | 17     | 16 | 15   | 14 | 13       | 12       | 11 | 10       | 9 | 8       | 7 | 6      | 5 | 4  | 3 | 2     | 1 | 0        |        |  |      |  |    |  |
| Reset  | 0x00000000   |          |    |          |    |         |    |        |    |    |    |       |    |          |        |    |      |    |          |          |    |          |   |         |   |        |   |    |   |       |   |          |        |  |      |  |    |  |
| Name   | Reserved     | SYNCLKBT |    | ASYNCLKB |    | SYNCLKB |    | CKDIVB |    |    |    | SYNSB |    | Reserved | CAPMDB |    | CSTB |    | Reserved | SYNCLKAT |    | ASYNCLKA |   | SYNCLKA |   | CKDIVA |   |    |   | SYNSA |   | Reserved | CAPMDA |  | CSTA |  |    |  |
| Access |              | RW       |    | RW       |    | RW      |    | RW     |    |    |    | RW    |    |          | RW     |    | RW   |    |          | RW       |    | RW       |   | RW      |   | RW     |   | RW |   |       |   | RW       |        |  | RW   |  | RW |  |

| Bit   | Name        | Function                                           | Description                                                                                                                                                                                                                                                                                                                                   | Access |
|-------|-------------|----------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:30 | Reserved    | Reserved                                           | -                                                                                                                                                                                                                                                                                                                                             | -      |
| 29    | SYNCLKBT[1] | Synchronous counting and clock source selection BT | Condition: Count underflow occurs in unit n of Timer6<br><ul style="list-style-type: none"> <li>0b0: When the condition matches, the synchronous counting of Timer2 unit m is disabled.</li> <li>0b1: When the condition matches, Timer2 unit m performs one synchronous count.</li> </ul> (When m=1, 2, 3, 4, n=2, 4, 6, 8)                  | RW     |
| 28    | SYNCLKBT[0] | Synchronous counting and clock source selection BT | Condition: Count overflow occurs in unit n of Timer6<br><ul style="list-style-type: none"> <li>0b0: When the condition matches, the synchronous counting of Timer2 unit m is disabled.</li> <li>0b1: When the condition matches, Timer2 unit m performs one synchronous count.</li> </ul> (When m=1, 2, 3, 4, n=2, 4, 6, 8)                   | RW     |
| 27:26 | ASYNCLKB    | Asynchronous counting and clock source selection B | <ul style="list-style-type: none"> <li>0b00: LRC</li> <li>0b01: XTAL32</li> <li>0b10: TMR2_&lt;t&gt;_CLKB clock input</li> <li>0b11: Setting prohibited</li> </ul>                                                                                                                                                                            | RW     |
| 25:24 | SYNCLKB     | Synchronous counting and clock source selection B  | <ul style="list-style-type: none"> <li>0b00: PCLK1</li> <li>0b01: TMR2_&lt;t&gt;_TRIGB rising edge (internal PCLK1 synchronization)</li> <li>0b10: TMR2_&lt;t&gt;_TRIGB falling edge (internal PCLK1 synchronization)</li> <li>0b11: Internal trigger event input</li> </ul>                                                                  | RW     |
| 23:20 | CKDIVB      | Count clock frequency division selection B         | <ul style="list-style-type: none"> <li>0b0000: Clock source</li> <li>0b0001: Clock source/2</li> <li>0b0010: Clock source/4</li> <li>0b0011: Clock source/8</li> <li>0b0100: Clock source/16</li> <li>0b0101: Clock source/32</li> <li>0b0110: Clock source/64</li> <li>0b0111: Clock source/128</li> <li>0b1000: Clock source/256</li> </ul> | RW     |

| Bit   | Name            | Function                                           | Description                                                                                                                                                                                                                                                                                                                                                                                                  | Access |
|-------|-----------------|----------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 23:20 | CKDIVB          | Count clock frequency division selection B         | <ul style="list-style-type: none"> <li>● 0b1001: Clock source/512</li> <li>● 0b1010: Clock source/1024</li> <li>● Others: Setting prohibited</li> </ul> <div>  <b>Note</b><br/>           The divided clock source can be various clock sources for asynchronous counting or PCLK1 for synchronous counting.         </div> | RW     |
| 19    | SYNSB           | Counting mode selection B                          | <ul style="list-style-type: none"> <li>● 0b0: Synchronous counting mode</li> <li>● 0b1: Asynchronous counting mode</li> </ul>                                                                                                                                                                                                                                                                                | RW     |
| 18    | <i>Reserved</i> | <i>Reserved</i>                                    | -                                                                                                                                                                                                                                                                                                                                                                                                            | -      |
| 17    | CAPMDB          | Functional mode selection B                        | <ul style="list-style-type: none"> <li>● 0b0: Compare output function</li> <li>● 0b1: Capture input function</li> </ul>                                                                                                                                                                                                                                                                                      | RW     |
| 16    | CSTB            | Timer start B                                      | <ul style="list-style-type: none"> <li>● 0b0: Channel B timer stop</li> <li>● 0b1: Channel B timer start</li> </ul> <div>  <b>Note</b><br/>           This bit will automatically be set to 0 when the hardware trigger stop condition is enable.         </div>                                                            | RW     |
| 15:14 | <i>Reserved</i> | <i>Reserved</i>                                    | -                                                                                                                                                                                                                                                                                                                                                                                                            | -      |
| 13    | SYN-CLKAT[1]    | Synchronous counting and clock source selection AT | Condition: Count underflow occurs in unit n of Timer6 <ul style="list-style-type: none"> <li>● 0b0: When the condition matches, the synchronous counting of Timer2 unit m is disabled.</li> <li>● 0b1: When the condition matches, Timer2 unit m performs one synchronous count.</li> </ul> (When m=1, 2, 3, 4, n=1, 3, 5, 7)                                                                                | RW     |
| 12    | SYN-CLKAT[0]    | Synchronous counting and clock source selection AT | Condition: Count overflow occurs in unit n of Timer6 <ul style="list-style-type: none"> <li>● 0b0: When the condition matches, the synchronous counting of Timer2 unit m is disabled.</li> <li>● 0b1: When the condition matches, Timer2 unit m performs one synchronous count.</li> </ul> (When m=1, 2, 3, 4, n=1, 3, 5, 7)                                                                                 | RW     |
| 11:10 | ASYNCLKA        | Asynchronous counting and clock source selection A | <ul style="list-style-type: none"> <li>● 0b00: LRC</li> <li>● 0b01: XTAL32</li> <li>● 0b10: TMR2_&lt;t&gt;_CLKA clock input</li> <li>● 0b11: Setting prohibited</li> </ul>                                                                                                                                                                                                                                   | RW     |
| 9:8   | SYNCLKA         | Synchronous counting and clock source selection A  | <ul style="list-style-type: none"> <li>● 0b00: PCLK1</li> <li>● 0b01: TMR2_&lt;t&gt;TRIGA rising edge (internally synchronized with PCLK1)</li> <li>● 0b10: TMR2&lt;t&gt;_TRIGA falling edge (internally synchronized with PCLK1)</li> <li>● 0b11: Internal trigger event input</li> </ul>                                                                                                                   | RW     |

| Bit | Name            | Function                                   | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | Access |
|-----|-----------------|--------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 7:4 | CKDIVA          | Count clock frequency division selection A | <ul style="list-style-type: none"> <li>● 0b0000: Clock source</li> <li>● 0b0001: Clock source/2</li> <li>● 0b0010: Clock source/4</li> <li>● 0b0011: Clock source/8</li> <li>● 0b0100: Clock source/16</li> <li>● 0b0101: Clock source/32</li> <li>● 0b0110: Clock source/64</li> <li>● 0b0111: Clock source/128</li> <li>● 0b1000: Clock source/256</li> <li>● 0b1001: Clock source/512</li> <li>● 0b1010: Clock source/1024</li> <li>● Others: Setting prohibited</li> </ul> <div>  <b>Note</b><br/>           The divided clock source can be various clock sources for asynchronous counting or PCLK1 for synchronous counting.         </div> | RW     |
| 3   | SYNSA           | Counting mode selection A                  | <ul style="list-style-type: none"> <li>● 0b0: Synchronous counting mode</li> <li>● 0b1: Asynchronous counting mode</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | RW     |
| 2   | <i>Reserved</i> | <i>Reserved</i>                            | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | -      |
| 1   | CAPMDA          | Pattern selection                          | <ul style="list-style-type: none"> <li>● 0b0: Compare output function</li> <li>● 0b1: Capture input function</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | RW     |
| 0   | CSTA            | Timer startup A                            | <ul style="list-style-type: none"> <li>● 0b0: Channel A timer stop</li> <li>● 0b1: Channel A timer start</li> </ul> <div>  <b>Note</b><br/>           This bit will automatically be set to 0 when the hardware trigger stop condition is enable.         </div>                                                                                                                                                                                                                                                                                                                                                                                 | RW     |

## 25.4.5 Interrupt Control Register (TMR2\_ICONR)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |       |       |          |    |    |    |    |    |   |   |   |   |   |   |   |   |       |       |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|-------|-------|----------|----|----|----|----|----|---|---|---|---|---|---|---|---|-------|-------|
| 0x0014 | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17    | 16    | 15       | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1     | 0     |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |       |       |          |    |    |    |    |    |   |   |   |   |   |   |   |   |       |       |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    | OVENB | CMENB | Reserved |    |    |    |    |    |   |   |   |   |   |   |   |   | OVENA | CMENA |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    | RW    | RW    |          |    |    |    |    |    |   |   |   |   |   |   |   |   | RW    | RW    |

| Bit   | Name     | Function                          | Description                                                                                                                                                                                                                                  | Access |
|-------|----------|-----------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:18 | Reserved | Reserved                          | -                                                                                                                                                                                                                                            | -      |
| 17    | OVENB    | Count overflow interrupt enable B | Interrupt generation condition: when the count value (CNTBR)=0xFFFF<br><ul style="list-style-type: none"> <li>0b0: Disable interrupt</li> <li>0b1: Enable interrupt</li> </ul>                                                               | RW     |
| 16    | CMENB    | Count match interrupt enable B    | Interrupt generation condition: When the CMPBR register is equal to the count value (CNTBR), or when a capture input event occurs<br><ul style="list-style-type: none"> <li>0b0: Disable interrupt</li> <li>0b1: Enable interrupt</li> </ul> | RW     |
| 15:2  | Reserved | Reserved                          | -                                                                                                                                                                                                                                            | -      |
| 1     | OVENA    | Count overflow interrupt enable A | Interrupt generation condition: when the count value (CNTAR)=0xFFFF<br><ul style="list-style-type: none"> <li>0b0: Disable interrupt</li> <li>0b1: Enable interrupt</li> </ul>                                                               | RW     |
| 0     | CMENA    | Count match interrupt enable A:   | Interrupt generation condition: When the CMPAR register is equal to the count value (CNTAR), or when a capture input event occurs<br><ul style="list-style-type: none"> <li>0b0: Disable interrupt</li> <li>0b1: Enable interrupt</li> </ul> | RW     |



## 25.4.6 Port Control Register (TMR2\_PCONR)

| Offset | Bit Position |         |         |          |        |          |       |       |       |          |         |         |          |        |          |       |
|--------|--------------|---------|---------|----------|--------|----------|-------|-------|-------|----------|---------|---------|----------|--------|----------|-------|
| 0x0018 | 31           | 30      | 29      | 28       | 27     | 26       | 25    | 24    | 23    | 22       | 21      | 20      | 19       | 18     | 17       | 16    |
| Reset  | 0x00000000   |         |         |          |        |          |       |       |       |          |         |         |          |        |          |       |
| Name   | Reserved     | NOFICKB | NOFIENB | Reserved | OUTENB | Reserved | CMPCB | STPCB | STACB | Reserved | NOFICKA | NOFIENA | Reserved | OUTENA | Reserved | CMPCA |
| Access |              | RW      | RW      |          | RW     |          | RW    | RW    | RW    |          | RW      | RW      |          | RW     |          | RW    |

| Bit   | Name     | Function                                           | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | Access |
|-------|----------|----------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31    | Reserved | Reserved                                           | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | -      |
| 30:29 | NOFICKB  | Filter sampling reference clock selection B        | <ul style="list-style-type: none"> <li>0b00: Clock source</li> <li>0b01: Clock source/4</li> <li>0b10: Clock source/16</li> <li>0b11: Clock source/64</li> </ul> <div>  <b>Note</b><br/>           The clock source is various clock sources for asynchronous counting and PCLK1 for synchronous counting.         </div>                                                                                                                                        | RW     |
| 28    | NOFIENB  | Trigger port filter B                              | <ul style="list-style-type: none"> <li>0b0: TMR2_&lt;t&gt;_TRIGB input port filter function is disabled</li> <li>0b1: TMR2_&lt;t&gt;_TRIGB input port filter function is enabled</li> </ul>                                                                                                                                                                                                                                                                                                                                                       | RW     |
| 27:25 | Reserved | Reserved                                           | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | -      |
| 24    | OUTENB   | Port output enable B                               | <ul style="list-style-type: none"> <li>0b0: TMR2_&lt;t&gt;_PWMB port output disabled</li> <li>0b1: TMR2_&lt;t&gt;_PWMB Port output enabled</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                             | RW     |
| 23:22 | Reserved | Reserved                                           | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | -      |
| 21:20 | CMPCB    | Port status setting B when comparison values match | <ul style="list-style-type: none"> <li>0b00: When the count value (CNTBR) is equal to CMPBR, the TMR2_&lt;t&gt;_PWMB port output is set to low level</li> <li>0b01: When the count value (CNTBR) is equal to CMPBR, the TMR2_&lt;t&gt;_PWMB port output is set to high level</li> <li>0b10: When the count value (CNTBR) is equal to CMPBR, the TMR2_&lt;t&gt;_PWMB port output retains the previous state</li> <li>0b11: When the count value (CNTBR) is equal to CMPBR, the TMR2_&lt;t&gt;_PWMB port output is set to inverted level</li> </ul> | RW     |

| Bit   | Name     | Function                                       | Description                                                                                                                                                                                                                                                                                                                                                                                                                                         | Access |
|-------|----------|------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 19:18 | STPCB    | Port status setting B when counting stops      | <ul style="list-style-type: none"> <li>0b00: When the counter stops, the TMR2_&lt;t&gt;_PWMB port output is set to low level</li> <li>0b01: When the counter stops, the TMR2_&lt;t&gt;_PWMB port output is set to high level</li> <li>0b10: When the counter stops, the TMR2_&lt;t&gt;_PWMB port output retains the previous state</li> <li>0b11: When the counter stops, the TMR2_&lt;t&gt;_PWMB port output retains the previous state</li> </ul> | RW     |
| 17:16 | STACB    | Port status setting B when counting starts     | <ul style="list-style-type: none"> <li>0b00: When counting starts, TMR2_&lt;t&gt;_PWMB port output is set to low level</li> <li>0b01: When counting starts, TMR2_&lt;t&gt;_PWMB port output is set to high level</li> <li>0b10: When counting starts, TMR2_&lt;t&gt;_PWMB port output retains the previous state</li> <li>0b11: When counting starts, TMR2_&lt;t&gt;_PWMB port output retains the previous state</li> </ul>                         | RW     |
| 15    | Reserved | Reserved                                       | -                                                                                                                                                                                                                                                                                                                                                                                                                                                   | -      |
| 14:13 | NOFICKA  | Filtering sampling reference clock selection A | <ul style="list-style-type: none"> <li>0b00: Clock source</li> <li>0b01: Clock source/4</li> <li>0b10: Clock source/16</li> <li>0b11: Clock source/64</li> </ul> <div>  <b>Note</b><br/>           The clock source is various clock sources for asynchronous counting and PCLK1 for synchronous counting.         </div>                                        | RW     |
| 12    | NOFIENA  | Trigger port filter A                          | <ul style="list-style-type: none"> <li>0b0: TMR2_&lt;t&gt;_TRIGA input port filter function disabled</li> <li>0b1: TMR2_&lt;t&gt;_TRIGA input port filter function enabled</li> </ul>                                                                                                                                                                                                                                                               | RW     |
| 11:9  | Reserved | Reserved                                       | -                                                                                                                                                                                                                                                                                                                                                                                                                                                   | -      |
| 8     | OUTENA   | Port output enable A                           | <ul style="list-style-type: none"> <li>0b0: TMR2_&lt;t&gt;_PWMA port output disabled</li> <li>0b1: TMR2_&lt;t&gt;_PWMA port output enabled</li> </ul>                                                                                                                                                                                                                                                                                               | RW     |
| 7:6   | Reserved | Reserved                                       | -                                                                                                                                                                                                                                                                                                                                                                                                                                                   | -      |

| Bit | Name  | Function                                          | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | Access |
|-----|-------|---------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 5:4 | CMPCA | Port state setting A When comparison values match | <ul style="list-style-type: none"> <li>● 0b00: When the count value (CNTAR) is equal to CMPAR, the TMR2_&lt;t&gt;_PWMA port output is set to low level</li> <li>● 0b01: When the count value (CNTAR) is equal to CMPAR, the TMR2_&lt;t&gt;_PWMA port output is set to high level</li> <li>● 0b10: When the count value (CNTAR) is equal to CMPAR, the TMR2_&lt;t&gt;_PWMA port output retains the previous state</li> <li>● 0b11: When the count value (CNTAR) is equal to CMPAR, the TMR2_&lt;t&gt;_PWMA port output is set to inverted level</li> </ul> | RW     |
| 3:2 | STPCA | Port status setting A when counting stops         | <ul style="list-style-type: none"> <li>● 0b00: When counting stops, the TMR2_&lt;t&gt;_PWMA port output is set to low level</li> <li>● 0b01: When counting stops, the TMR2_&lt;t&gt;_PWMA port output is set to high level</li> <li>● 0b10: When counting stops, the TMR2_&lt;t&gt;_PWMA port output retains the previous state</li> <li>● 0b11: When counting stops, the TMR2_&lt;t&gt;_PWMA port output retains the previous state</li> </ul>                                                                                                           | RW     |
| 1:0 | STACA | Port status setting A when counting starts        | <ul style="list-style-type: none"> <li>● 0b00: When counting starts, TMR2_&lt;t&gt;_PWMA port output is set to low level</li> <li>● 0b01: When counting starts, TMR2_&lt;t&gt;_PWMA port output is set to high level</li> <li>● 0b10: When counting starts, TMR2_&lt;t&gt;_PWMA port output retains the previous state</li> <li>● 0b11: When counting starts, TMR2_&lt;t&gt;_PWMA port output retains the previous state</li> </ul>                                                                                                                       | RW     |

## 25.4.7 Hardware Control Register (TMR2\_HCONR)

| Offset | Bit Position |        |        |        |          |        |        |        |          |        |        |        |          |        |        |        |          |        |        |        |          |        |        |        |          |        |        |        |          |        |        |        |
|--------|--------------|--------|--------|--------|----------|--------|--------|--------|----------|--------|--------|--------|----------|--------|--------|--------|----------|--------|--------|--------|----------|--------|--------|--------|----------|--------|--------|--------|----------|--------|--------|--------|
| 0x001C | 31           | 30     | 29     | 28     | 27       | 26     | 25     | 24     | 23       | 22     | 21     | 20     | 19       | 18     | 17     | 16     | 15       | 14     | 13     | 12     | 11       | 10     | 9      | 8      | 7        | 6      | 5      | 4      | 3        | 2      | 1      | 0      |
| Reset  | 0x00000000   |        |        |        |          |        |        |        |          |        |        |        |          |        |        |        |          |        |        |        |          |        |        |        |          |        |        |        |          |        |        |        |
| Name   | Reserved     | HICPB2 | HICPB1 | HICPB0 | Reserved | HCLEB2 | HCLEB1 | HCLEB0 | Reserved | HSTPB2 | HSTPB1 | HSTPB0 | Reserved | HSTAB2 | HSTAB1 | HSTAB0 | Reserved | HICPA2 | HICPA1 | HICPA0 | Reserved | HCLEA2 | HCLEA1 | HCLEA0 | Reserved | HSTPA2 | HSTPA1 | HSTPA0 | Reserved | HSTAA2 | HSTAA1 | HSTAA0 |
| Access |              | RW     | RW     | RW     |          | RW     | RW     | RW     |          | RW     | RW     | RW     |          | RW     | RW     | RW     |          | RW     | RW     | RW     |          | RW     | RW     | RW     |          | RW     | RW     | RW     |          | RW     | RW     | RW     |

| Bit | Name     | Function                            | Description                                                                                                                                                                                                                                                 | Access |
|-----|----------|-------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31  | Reserved | Reserved                            | -                                                                                                                                                                                                                                                           | -      |
| 30  | HICPB2   | Hardware capture input condition B2 | Condition: Internal Hardware Trigger Event Active <ul style="list-style-type: none"> <li>0b0: Hardware capture input is disabled when the condition is met</li> <li>0b1: Hardware capture input is enabled when the condition is met</li> </ul>             | RW     |
| 29  | HICPB1   | Hardware capture input condition B1 | Condition: TMR2_<t>_TRIGB port is sampled to the falling edge <ul style="list-style-type: none"> <li>0b0: Hardware capture input is disabled when the condition is met</li> <li>0b1: Hardware capture input is enabled when the condition is met</li> </ul> | RW     |
| 28  | HICPB0   | Hardware capture input condition B0 | Condition: TMR2_<t>_TRIGB port is sampled to the rising edge <ul style="list-style-type: none"> <li>0b0: Hardware capture input is disabled when the condition is met</li> <li>0b1: Hardware capture input is enabled when the condition is met</li> </ul>  | RW     |
| 27  | Reserved | Reserved                            | -                                                                                                                                                                                                                                                           | -      |
| 26  | HCLEB2   | Hardware clear condition B2         | Condition: Internal Hardware Trigger Event Active <ul style="list-style-type: none"> <li>0b0: Hardware clear is disabled when the condition is met</li> <li>0b1: Hardware clear is enabled when the condition is met</li> </ul>                             | RW     |
| 25  | HCLEB1   | Hardware clear condition B1         | Condition: TMR2_<t>_TRIGB port is sampled to the falling edge <ul style="list-style-type: none"> <li>0b0: Hardware clear is disabled when the condition is met</li> <li>0b1: Hardware clear is enabled when the condition is met</li> </ul>                 | RW     |

| Bit | Name            | Function                    | Description                                                                                                                                                                                                                                    | Access |
|-----|-----------------|-----------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 24  | HCLEB0          | Hardware clear condition B0 | Condition: TMR2_<t>_TRIGB port is sampled to the rising edge<br><ul style="list-style-type: none"> <li>0b0: Hardware clear is disabled when the condition is met</li> <li>0b1: Hardware clear is enabled when the condition is met</li> </ul>  | RW     |
| 23  | <i>Reserved</i> | <i>Reserved</i>             | -                                                                                                                                                                                                                                              | -      |
| 22  | HSTPB2          | Hardware stop condition B2  | Condition: Internal Hardware Trigger Event Active<br><ul style="list-style-type: none"> <li>0b0: Hardware stop is disabled when the condition is met</li> <li>0b1: Hardware stop is enabled when the condition is met</li> </ul>               | RW     |
| 21  | HSTPB1          | Hardware stop condition B1  | Condition: TMR2_<t>_TRIGB port is sampled to the falling edge<br><ul style="list-style-type: none"> <li>0b0: Hardware stop is disabled when the condition is met</li> <li>0b1: Hardware stop is enabled when the condition is met</li> </ul>   | RW     |
| 20  | HSTPB0          | Hardware stop condition B0  | Condition: TMR2_<t>_TRIGB port is sampled to the rising edge<br><ul style="list-style-type: none"> <li>0b0: Hardware stop is disabled when the condition is met</li> <li>0b1: Hardware stop is enabled when the condition is met</li> </ul>    | RW     |
| 19  | <i>Reserved</i> | <i>Reserved</i>             | -                                                                                                                                                                                                                                              | -      |
| 18  | HSTAB2          | Hardware start condition B2 | Condition: Internal Hardware Trigger Event Active<br><ul style="list-style-type: none"> <li>0b0: Hardware start is disabled when the condition is met</li> <li>0b1: Hardware start is enabled when the condition is met</li> </ul>             | RW     |
| 17  | HSTAB1          | Hardware start condition B1 | Condition: TMR2_<t>_TRIGB port is sampled to the falling edge<br><ul style="list-style-type: none"> <li>0b0: Hardware start is disabled when the condition is met</li> <li>0b1: Hardware start is enabled when the condition is met</li> </ul> | RW     |
| 16  | HSTAB0          | Hardware start condition B0 | Condition: TMR2_<t>_TRIGB port is sampled to the rising edge<br><ul style="list-style-type: none"> <li>0b0: Hardware start is disabled when the condition is met</li> <li>0b1: Hardware start is enabled when the condition is met</li> </ul>  | RW     |
| 15  | <i>Reserved</i> | <i>Reserved</i>             | -                                                                                                                                                                                                                                              | -      |

| Bit | Name            | Function                            | Description                                                                                                                                                                                                                                                 | Access |
|-----|-----------------|-------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 14  | HICPA2          | Hardware capture input condition A2 | Condition: Internal Hardware Trigger Event Active <ul style="list-style-type: none"> <li>0b0: Hardware capture input is disabled when the condition is met</li> <li>0b1: Hardware capture input is enabled when the condition is met</li> </ul>             | RW     |
| 13  | HICPA1          | Hardware capture input condition A1 | Condition: TMR2_<t>_TRIGA port is sampled to the falling edge <ul style="list-style-type: none"> <li>0b0: Hardware capture input is disabled when the condition is met</li> <li>0b1: Hardware capture input is enabled when the condition is met</li> </ul> | RW     |
| 12  | HICPA0          | Hardware capture input condition A0 | Condition: TMR2_<t>_TRIGA port is sampled to the rising edge <ul style="list-style-type: none"> <li>0b0: Hardware capture input is disabled when the condition is met</li> <li>0b1: Hardware capture input is enabled when the condition is met</li> </ul>  | RW     |
| 11  | <i>Reserved</i> | <i>Reserved</i>                     | -                                                                                                                                                                                                                                                           | -      |
| 10  | HCLEA2          | Hardware clear condition A2         | Condition: Internal Hardware Trigger Event Active <ul style="list-style-type: none"> <li>0b0: Hardware clear is disabled when the condition is met</li> <li>0b1: Hardware clear is enabled when the condition is met</li> </ul>                             | RW     |
| 9   | HCLEA1          | Hardware clear condition A1         | Condition: TMR2_<t>_TRIGA port is sampled to the falling edge <ul style="list-style-type: none"> <li>0b0: Hardware clear is disabled when the condition is met</li> <li>0b1: Hardware clear is enabled when the condition is met</li> </ul>                 | RW     |
| 8   | HCLEA0          | Hardware clear condition A0         | Condition: TMR2_<t>_TRIGA port is sampled to the rising edge <ul style="list-style-type: none"> <li>0b0: Hardware clear is disabled when the condition is met</li> <li>0b1: Hardware clear is enabled when the condition is met</li> </ul>                  | RW     |
| 7   | <i>Reserved</i> | <i>Reserved</i>                     | -                                                                                                                                                                                                                                                           | -      |
| 6   | HSTPA2          | Hardware stop condition A2          | Condition: Internal Hardware Trigger Event Active <ul style="list-style-type: none"> <li>0b0: Hardware stop is disabled when the condition is met</li> <li>0b1: Hardware stop is enabled when the condition is met</li> </ul>                               | RW     |

| Bit | Name            | Function                    | Description                                                                                                                                                                                                                                     | Access |
|-----|-----------------|-----------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 5   | HSTPA1          | Hardware stop condition A1  | Condition: TMR2_<t>_TRIGA port is sampled to the falling edge <ul style="list-style-type: none"> <li>● 0b0: Hardware stop is disabled when the condition is met</li> <li>● 0b1: Hardware stop is enabled when the condition is met</li> </ul>   | RW     |
| 4   | HSTPA0          | Hardware stop condition A0  | Condition: TMR2_<t>_TRIGA port is sampled to the rising edge <ul style="list-style-type: none"> <li>● 0b0: Hardware stop is disabled when the condition is met</li> <li>● 0b1: Hardware stop is enabled when the condition is met</li> </ul>    | RW     |
| 3   | <i>Reserved</i> | <i>Reserved</i>             | -                                                                                                                                                                                                                                               | -      |
| 2   | HSTAA2          | Hardware start condition A2 | Condition: Internal Hardware Trigger Event Active <ul style="list-style-type: none"> <li>● 0b0: Hardware start is disabled when the condition is met</li> <li>● 0b1: Hardware start is enabled when the condition is met</li> </ul>             | RW     |
| 1   | HSTAA1          | Hardware start condition A1 | Condition: TMR2_<t>_TRIGA port is sampled to the falling edge <ul style="list-style-type: none"> <li>● 0b0: Hardware start is disabled when the condition is met</li> <li>● 0b1: Hardware start is enabled when the condition is met</li> </ul> | RW     |
| 0   | HSTAA0          | Hardware start condition A0 | Condition: TMR2_<t>_TRIGA port is sampled to the rising edge <ul style="list-style-type: none"> <li>● 0b0: Hardware start is disabled when the condition is met</li> <li>● 0b1: Hardware start is enabled when the condition is met</li> </ul>  | RW     |

## 25.4.8 Status Flag Register (TMR2\_STFLR)

| Offset | Bit Position |    |    |    |    |    |    |    |          |    |    |    |    |    |    |    |    |          |    |    |    |    |   |   |   |          |   |   |   |   |   |   |  |    |    |    |  |  |  |  |  |
|--------|--------------|----|----|----|----|----|----|----|----------|----|----|----|----|----|----|----|----|----------|----|----|----|----|---|---|---|----------|---|---|---|---|---|---|--|----|----|----|--|--|--|--|--|
| 0x0020 | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23       | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14       | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6        | 5 | 4 | 3 | 2 | 1 | 0 |  |    |    |    |  |  |  |  |  |
| Reset  | 0x01000100   |    |    |    |    |    |    |    |          |    |    |    |    |    |    |    |    |          |    |    |    |    |   |   |   |          |   |   |   |   |   |   |  |    |    |    |  |  |  |  |  |
| Name   | Reserved     |    |    |    |    |    |    |    | Reserved |    |    |    |    |    |    |    |    | Reserved |    |    |    |    |   |   |   | Reserved |   |   |   |   |   |   |  |    |    |    |  |  |  |  |  |
| Access |              |    |    |    |    |    |    | RO |          |    |    |    |    |    |    | RW | RW | RW       |    |    |    |    |   |   |   | RO       |   |   |   |   |   |   |  | RW | RW | RW |  |  |  |  |  |

| Bit   | Name     | Function                          | Description                                                                                                                                                                                                                                                                          | Access |
|-------|----------|-----------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:25 | Reserved | Reserved                          | -                                                                                                                                                                                                                                                                                    | -      |
| 24    | SYDB     | Synchronization completion flag B | <ul style="list-style-type: none"> <li>0b0: B channel register is synchronizing during asynchronous counting</li> <li>0b1: B channel register synchronization is completed during asynchronous counting</li> </ul>                                                                   | RO     |
| 23:19 | Reserved | Reserved                          | -                                                                                                                                                                                                                                                                                    | -      |
| 18    | ICPB     | Repeat capture flag B             | <ul style="list-style-type: none"> <li>0b0: No capture input occurred after the count match flag bit CMFB was set to 1</li> <li>0b1: A capture input occurred again after the count match flag bit CMFB was set to 1</li> </ul>                                                      | RW     |
| 17    | OVFB     | Count overflow flag B             | <ul style="list-style-type: none"> <li>0b0: The count value (CNTBR) does not count to 0xFFFF</li> <li>0b1: The count value (CNTBR) counts to 0xFFFF</li> </ul>                                                                                                                       | RW     |
| 16    | CMFB     | Count match flag B                | <ul style="list-style-type: none"> <li>0b0: The value of the CMPBR register is not equal to the count value (CNTBR) and no capture input action occurs</li> <li>0b1: The value of the CMPBR register is equal to the count value (CNTBR) or a capture input action occurs</li> </ul> | RW     |
| 15:9  | Reserved | Reserved                          | -                                                                                                                                                                                                                                                                                    | -      |
| 8     | SYDA     | Synchronization completion flag A | <ul style="list-style-type: none"> <li>0b0: A channel register is synchronizing during asynchronous counting</li> <li>0b1: A channel register synchronization is completed during asynchronous counting</li> </ul>                                                                   | RO     |
| 7:3   | Reserved | Reserved                          | -                                                                                                                                                                                                                                                                                    | -      |
| 2     | ICPA     | Repeat capture flag A             | <ul style="list-style-type: none"> <li>0b0: No capture input occurred after the count match flag bit CMFA was set to 1</li> <li>0b1: A capture input occurred again after the count match flag bit CMFA was set to 1</li> </ul>                                                      | RW     |
| 1     | OVFA     | Count overflow flag A             | <ul style="list-style-type: none"> <li>0b0: The count value (CNTAR) does not count to 0xFFFF</li> <li>0b1: The count value (CNTAR) counts to 0xFFFF</li> </ul>                                                                                                                       | RW     |



| Bit | Name | Function           | Description                                                                                                                                                                                                                                                                           | Access |
|-----|------|--------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 0   | CMFA | Count match flag A | <ul style="list-style-type: none"><li>● 0b0: The value of the CMPAR register is not equal to the count value (CNTAR) and no capture input action occurs</li><li>● 0b1: The value of the CMPAR register is equal to the count value (CNTAR) or a capture input action occurs</li></ul> | RW     |

## 25.5 Precautions

1. During asynchronous counting operation, you must first configure the asynchronous clock source by setting the BCONR.ASYNCCLKA<B> bit, then select the asynchronous counting mode by setting the BCONR.SYNSA<B> bit, and finally start Timer2.
2. Under asynchronous counting mode, when the count value (CNTA<B>R), reference value (CMPA<B>R), start bit (BCONR.CSTA<B>), and status bit (STFLR.CMFA<B>, STFLR.OVFA<B>, STFLR.ICPA<B>) are modified, Timer2 will write the modified value into the corresponding register after 3 asynchronous counting clocks after receiving the write action.
3. Under asynchronous counting mode (BCONR.SYNSA<B>=0b1), when performing consecutive write actions to any of the following registers: CNTA<B>R, CMPA<B>R, BCONR.CSTA<B>, or the status bits (STFLR.CMFA<B>, STFLR.OVFA<B>, STFLR.ICPA<B>), you must wait until the status flag STFLR.SY-DA<B> becomes 1 before issuing the next write to any of these registers.
4. When asynchronous counting is selected, ensure that BCONR.SYNCLKA<B> is set to 0.
5. When using Timer2 events to trigger actions in other peripherals, the reference value register setting for the corresponding channel of Timer2 must be greater than 5.

## 26 General Timer (Timer0)

### 26.1 Introduction

General Timer 0 (Timer0) is a basic timer that can realize synchronous counting and asynchronous counting. The timer contains two channels (CH-A and CH-B) that can generate compare match events and count overflow events during counting. This event can trigger an interrupt and can be used to trigger other module actions. This series of products is equipped with 5 units of Timer0.

### 26.2 Functional Description

#### 26.2.1 Timer0 Block Diagram

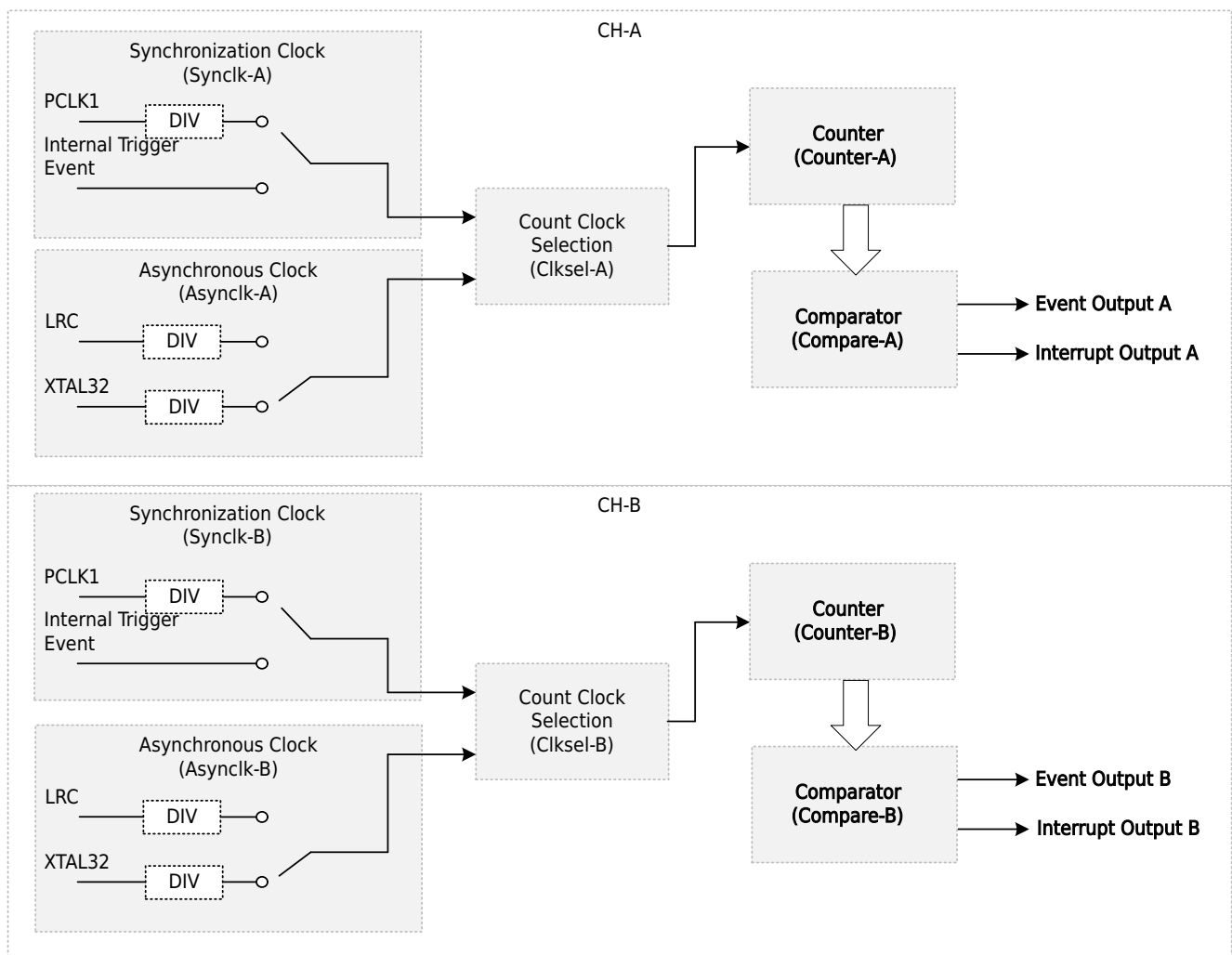


Figure 26-1 Timer0 Basic Block Diagram

#### 26.2.2 Clock Source Selection

Timer0 can be counted either synchronously or asynchronously.

The synchronous counting mode refers to the synchronous timing relationship between the timer count clock and the bus access clock (register read/write operation clock). Asynchronous counting means that the timer count clock and bus access clock (register read/write operation clock) are non-synchronous timing relationships. The status of a timer, etc., may be changing when the register is read by asyncho-

nous counting. Therefore, in the asynchronous counting mode, the register read operation must be implemented in the count stopped state.

### Synchronous Count Clock Source

In synchronous counting mode ( $\text{BCONR.SYNSA} \langle B \rangle = 0$ ), the clock source can have the following options ( $\text{BCONR.SYNCLKA} \langle B \rangle$  setting) :

1. PCLK1 and PCLK1 divided by 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024 are used as synchronous count clocks ( $\text{BCONR.SYNCLKA} \langle B \rangle = 0$  &  $\text{BCONR.CKDIVA} \langle B \rangle$  settings);
2. An internal hardware trigger event input can serve as the synchronous count clock ( $\text{BCONR.SYNCLKA} \langle B \rangle = 1$ );

### Asynchronous Count Clock Source

In asynchronous counting mode ( $\text{BCONR.SYNSA} \langle B \rangle = 1$ ), the clock source can be selected from the following options ( $\text{BCONR.ASYNCLKA} \langle B \rangle$  setting selection) :

1. LRC clock source input and its 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024 divisions are used as asynchronous counting clocks ( $\text{BCONR.ASYNCLKA} \langle B \rangle = 0$  &  $\text{BCONR.CKDIVA} \langle B \rangle$  settings);
2. XTAL32 clock source input and its 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024 divisions are used as asynchronous counting clocks ( $\text{BCONR.ASYNCLKA} \langle B \rangle = 1$  &  $\text{BCONR.CKDIVA} \langle B \rangle$  settings).

## 26.2.3 Basic Counting

Each channel of Timer0 can set the reference count value, and generate a count compare match event when the count value is equal to the reference value.

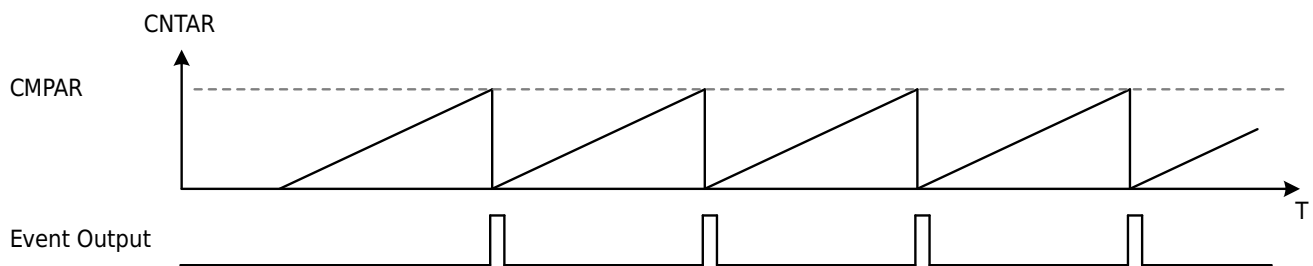


Figure 26-2 Timer0 Counting Timing Diagram

## 26.2.4 Hardware Triggering

Timer0 has two channels that share a common internal hardware trigger source. The timer's status (start, stop, reset) and capture input actions can be controlled through the relevant settings in the Basic Control Register (BCONR).

The source of this hardware trigger is selected by writing the corresponding trigger number into the Hardware Trigger Selection Register (TMR0\_TRGSEL). For details on the register contents, refer to the "Automatic Operation System (AOS) " section, and for the specific event mappings, refer to the "Interrupt Controller (INTC) " section. When using the internal hardware trigger function, the AOS function control bit in Function Clock Control 0 Register (PWC\_FCG0) must be enabled first.

## 26.2.5 Software Synchronization Start

The product is equipped with two counter synchronization enable registers (TMR\_SYNNR1 and TMR\_SYNNR2) used to synchronously start the timers integrated in the product, as shown in the table below (\* represents any TMR unit of a given type; TMR\* refers to any type of TMR). By setting multiple start bits in TMR\_SYNNR1 and TMR\_SYNNR2 to 1, multiple timers can begin counting simultaneously.

For specific bit assignments, please refer to the Counter Synchronization Enable Register (TMR\_SYNENR1 and TMR\_SYNENR2). Configure the TMR\_SYNENR1 register before configuring the TMR\_SYNENR2 register.



#### Notice

- When using the synchronous start function, ensure that all timers to be started synchronously operate with the same PCLK clock frequency and that the corresponding timer clocks (PWC\_FCG2) are enabled.
- When using the synchronous start function, please ensure that the timers that need to be started synchronously are in synchronous counting mode.
- The TMR\_SYNENR1 register must be written synchronously when the TMR\_SYNENR2 register is written, that is, setting the TMR\_SYNENR1 register alone has no effect.

**Table 26-1 Counter and PWM Synchronization Start Status**

| Synchronous start      | TMR unit             | Counter synchronization | PWM synchronization |
|------------------------|----------------------|-------------------------|---------------------|
| Same types of TMR      | TMR0_1-3             | ✓                       | ×                   |
|                        | TMR0_4-5             | ✓                       | ×                   |
|                        | TMR2_*               | ✓                       | ✓                   |
|                        | TMR4_*               | ✓                       | ✓                   |
|                        | TMR6_*               | ✓                       | ✓                   |
|                        | TMRA_*               | ✓                       | ✓                   |
| Different types of TMR | TMR0_*+TMR*          | ×                       | ×                   |
|                        | TMR4_*+TMR*          | ×                       | ×                   |
|                        | TMR2_*+TMR6_*        | ✓                       | ✓                   |
|                        | TMR2_*+TMRA_*        | ✓                       | ✓                   |
|                        | TMR6_*+TMRA_*        | ✓                       | ✓                   |
|                        | TMR2_*+TMR6_*+TMRA_* | ✓                       | ✓                   |

## 26.2.6 Interrupt and Event

### 26.2.6.1 Interrupt

One Timer0 contains 4 interrupt outputs, namely, the count compare match interrupt or capture input interrupt of channel A and channel B, and the count overflow interrupt of channel A and channel B.

There are 2 reference value registers (CMPAR, CMPBR), which can be compared with the count value registers (CNTAR, CNTBR) respectively to generate comparison matching active signals. When the count compare matches, the STFLR.CMFA<B> bit in the Status Flag Register (STFLR) will be set to 1. At this time, if the BCONR.CMENA<B> bit of the Basic Control Register (BCONR) is set to enable interrupt, the corresponding interrupt request (TMR0\_<t>\_CMPn, <t>=1-5; n=A, B) will also be triggered.

When the internal hardware trigger input is used as the capture input condition, the corresponding capture input action can be generated. At this time, if the BCONR.CMENA<B> bit of the Basic Control Register (BCONR) is set to enable interrupt, the corresponding interrupt request (TMR0\_<t>\_CMPn, <t>=1-5; n=A, B) is triggered.

When the count value register (CNTAR, CNTBR) counts to 0xFFFF, a count overflow event occurs. At this time, the STFLR.OVFA<B> bit in the Status Flag Register (STFLR) will be set to 1. If the BCONR.OVENA<B> bit of the Basic Control Register (BCONR) is set to enable interrupt, the corresponding interrupt request (TMR0\_<t>\_OVFn, <t>=1-5; n=A, B) will also be triggered.

When the asynchronous counting mode is selected, the compare match interrupt generated by the reference value register (CMPAR) of unit 1 can be used to wake up the system in Stop mode. Please refer to the relevant description in the "Interrupt Controller (INTC) " section.

#### 26.2.6.2 Event

One Timer0 has four event outputs, namely, the count compare match events or capture input events of channel A and channel B, and the count overflow events of channel A and channel B.

When a count compare match, capture input action or count overflow occurs during the counting process, a corresponding event request (TMR0\_<t>\_CMPn or TMR0\_<t>\_OVFn, <t>=1-5, n=A, B) output signal will be generated, which can be used to select and trigger other modules.

## 26.3 Register Description

### 26.3.1 Register List

- Timer0\_1 base address: 0x40024000
- Timer0\_2 base address: 0x40024400
- Timer0\_3 base address: 0x40029000
- Timer0\_4 base address: 0x40029400
- Timer0\_5 base address: 0x40029800



#### Note

Timer0 is the name of the timer, and Timer0\_<t> (<t>: 1-5) is the timer unit.

Table 26-2 General Timer (Timer0) Register Offset

| Offset | Register                   | Description                | Width |
|--------|----------------------------|----------------------------|-------|
| 0x0000 | <a href="#">TMR0_CNTAR</a> | Count Value Register A     | 32    |
| 0x0004 | <a href="#">TMR0_CNTBR</a> | Count Value Register B     | 32    |
| 0x0008 | <a href="#">TMR0_CMPAR</a> | Reference Value Register A | 32    |
| 0x000C | <a href="#">TMR0_CMPBR</a> | Reference Value Register B | 32    |
| 0x0010 | <a href="#">TMR0_BCONR</a> | Basic Control Register     | 32    |
| 0x0014 | <a href="#">TMR0_STFLR</a> | Status Flag Register       | 32    |

TMR\_SYNENR base address: 0x40055400

Table 26-3 TMR\_SYNENR Register Offset

| Offset | Register                    | Description                                | Width |
|--------|-----------------------------|--------------------------------------------|-------|
| 0x14   | <a href="#">TMR_SYNENR1</a> | Count Synchronization Enable Register<br>1 | 32    |
| 0x40   | <a href="#">TMR_SYNENR2</a> | Count Synchronization Enable Register<br>2 | 32    |

### 26.3.2 Count Value Register m (TMR0\_CNTmR) (m=A/B)

Offset: 0x0000 (CNTAR), 0x0004 (CNTBR)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |     |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| -      | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15  | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |     |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | CNT |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |     |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|        |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RW  |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit   | Name     | Function    | Description                                                                                                                                                                                                                            | Access |
|-------|----------|-------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:16 | Reserved | Reserved    | -                                                                                                                                                                                                                                      | -      |
| 15:0  | CNT      | Count value | Current timer count value<br><div>  <b>Note</b><br/>           Writing to this register is not disabled while counting is in progress.         </div> | RW     |

### 26.3.3 Reference Value Register m (TMR0\_CMPmR) (m=A/B)

Offset: 0x0008 (CMPAR), 0x000C (CMPBR)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |     |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| -      | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15  | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x0000FFFF   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |     |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | CMP |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |     |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|        |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RW  |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit   | Name     | Function        | Description                                                        | Access |
|-------|----------|-----------------|--------------------------------------------------------------------|--------|
| 31:16 | Reserved | Reserved        | -                                                                  | -      |
| 15:0  | CMP      | Reference value | Set the counting reference value to generate a compare match event | RW     |



## 26.3.4 Basic Control Register (TMR0\_BCONR)

| Offset | Bit Position |       |       |       |          |          |         |       |    |        |    |    |       |       |        |      |       |       |       |       |          |          |         |       |   |        |   |   |       |       |        |      |
|--------|--------------|-------|-------|-------|----------|----------|---------|-------|----|--------|----|----|-------|-------|--------|------|-------|-------|-------|-------|----------|----------|---------|-------|---|--------|---|---|-------|-------|--------|------|
| 0x0010 | 31           | 30    | 29    | 28    | 27       | 26       | 25      | 24    | 23 | 22     | 21 | 20 | 19    | 18    | 17     | 16   | 15    | 14    | 13    | 12    | 11       | 10       | 9       | 8     | 7 | 6      | 5 | 4 | 3     | 2     | 1      | 0    |
| Reset  | 0x00000000   |       |       |       |          |          |         |       |    |        |    |    |       |       |        |      |       |       |       |       |          |          |         |       |   |        |   |   |       |       |        |      |
| Name   | HICPB        | HCLEB | HSTPB | HSTAB | Reserved | ASYNCLKB | SYNCLKB | SYNSB |    | CKDIVB |    |    | OVENB | CMENB | CAPMDB | CSTB | HICPA | HCLEA | HSTPA | HSTAA | Reserved | ASYNCLKA | SYNCLKA | SYNSA |   | CKDIVA |   |   | OVENA | CMENA | CAPMDA | CSTA |
| Access | RW           | RW    | RW    | RW    |          | RW       | RW      | RW    |    | RW     |    |    | RW    | RW    | RW     | RW   | RW    | RW    | RW    | RW    |          | RW       | RW      | RW    |   | RW     |   |   | RW    | RW    | RW     | RW   |

| Bit | Name     | Function                                              | Description                                                                                                                                                                                                                                                                                                                                                                                                                                 | Access |
|-----|----------|-------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31  | HICPB    | Hardware trigger capture input B                      | Condition: Internal Hardware Trigger Event Active <ul style="list-style-type: none"> <li>0b0: Capture input is disabled when the condition matches</li> <li>0b1: Capture input is enabled when the condition matches</li> </ul>                                                                                                                                                                                                             | RW     |
| 30  | HCLEB    | Hardware trigger clear B                              | Condition: Internal Hardware Trigger Event Active <ul style="list-style-type: none"> <li>0b0: Timer clear operation is disabled when the condition matches</li> <li>0b1: Timer clear operation is enabled when the condition matches</li> </ul>                                                                                                                                                                                             | RW     |
| 29  | HSTPB    | Hardware trigger stop B                               | Condition: Internal Hardware Trigger Event Active <ul style="list-style-type: none"> <li>0b0: Timer stop is disabled when the condition matches</li> <li>0b1: Timer stop is enabled when the condition matches</li> </ul>                                                                                                                                                                                                                   | RW     |
| 28  | HSTAB    | Hardware trigger start B                              | Condition: Internal Hardware Trigger Event Active <ul style="list-style-type: none"> <li>0b0: Timer Start is disabled when the condition matches</li> <li>0b1: Timer Start is enabled when the condition matches</li> </ul>                                                                                                                                                                                                                 | RW     |
| 27  | Reserved | Reserved                                              | -                                                                                                                                                                                                                                                                                                                                                                                                                                           | -      |
| 26  | ASYNCLKB | Channel B asynchronous count clock source selection   | <ul style="list-style-type: none"> <li>0b0: LRC</li> <li>0b1: XTAL32</li> </ul> <div>  <b>Note</b><br/> When the receive timeout function of the corresponding UART channel is enabled, the clock source is no longer XTAL32 (XTAL32 crystal oscillator may not exist in hardware), but the clock tick generated by the UART baud rate generator. </div> | RW     |
| 25  | SYNCLKB  | Channel B synchronous counting clock source selection | <ul style="list-style-type: none"> <li>0b0: PCLK1</li> <li>0b1: Internal hardware triggering event</li> </ul>                                                                                                                                                                                                                                                                                                                               | RW     |

| Bit   | Name   | Function                                           | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | Access |
|-------|--------|----------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 24    | SYNSB  | Channel B counting mode selection                  | <ul style="list-style-type: none"> <li>0b0: Synchronous counting mode</li> <li>0b1: Asynchronous counting mode</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | RW     |
| 23:20 | CKDIVB | Channel B count clock frequency division selection | <ul style="list-style-type: none"> <li>0b0000: Clock source</li> <li>0b0001: Clock source/2</li> <li>0b0010: Clock source/4</li> <li>0b0011: Clock source/8</li> <li>0b0100: Clock source/16</li> <li>0b0101: Clock source/32</li> <li>0b0110: Clock source/64</li> <li>0b0111: Clock source/128</li> <li>0b1000: Clock source/256</li> <li>0b1001: Clock source/512</li> <li>0b1010: Clock source/1024</li> <li>Others: Setting prohibited</li> </ul> <div>  <b>Note</b><br/>           The divided clock source can be various clock sources for asynchronous counting or PCLK1 for synchronous counting.         </div> | RW     |
| 19    | OVENB  | Count overflow interrupt enable B                  | Interrupt condition: count overflow, that is, count value (CNTBR)=0xFFFF <ul style="list-style-type: none"> <li>0b0: Disable interrupt</li> <li>0b1: Enable interrupt</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | RW     |
| 18    | CMENB  | Count match interrupt enable B                     | Interrupt condition: When the CMPBR register is equal to the count value (CNTBR), or when a capture input event occurs <ul style="list-style-type: none"> <li>0b0: Disable interrupt</li> <li>0b1: Enable interrupt</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | RW     |
| 17    | CAPMDB | Function mode selection B                          | <ul style="list-style-type: none"> <li>0b0: Compare output function</li> <li>0b1: Capture input function</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | RW     |
| 16    | CSTB   | Channel B timer ON                                 | <ul style="list-style-type: none"> <li>0b0: Channel B timer off</li> <li>0b1: Channel B timer on</li> </ul> <div>  <b>Note</b><br/>           This bit will automatically be set to 0 when the hardware trigger stop condition is enable.         </div>                                                                                                                                                                                                                                                                                                                                                                 | RW     |
| 15    | HICPA  | Hardware trigger capture input A                   | Condition: Internal Hardware Trigger Event Active <ul style="list-style-type: none"> <li>0b0: Capture input is disabled when the condition matches</li> <li>0b1: Capture input is enabled when the condition matches</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | RW     |

| Bit | Name     | Function                                              | Description                                                                                                                                                                                                                                                                                                                                                                                                                                            | Access |
|-----|----------|-------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 14  | HCLEA    | Hardware trigger clear A                              | Condition: Internal Hardware Trigger Event Active <ul style="list-style-type: none"> <li>0b0: Timer clear operation is disabled when the condition matches</li> <li>0b1: Timer clear operation is enabled when the condition matches</li> </ul>                                                                                                                                                                                                        | RW     |
| 13  | HSTPA    | Hardware trigger stop A                               | Condition: Internal Hardware Trigger Event Active <ul style="list-style-type: none"> <li>0b0: Timer stop is disabled when the condition matches</li> <li>0b1: Timer stop is enabled when the condition matches</li> </ul>                                                                                                                                                                                                                              | RW     |
| 12  | HSTAA    | Hardware trigger start A                              | Condition: Internal Hardware Trigger Event Active <ul style="list-style-type: none"> <li>0b0: Timer start is disabled when the condition matches</li> <li>0b1: Timer start is enabled when the condition matches</li> </ul>                                                                                                                                                                                                                            | RW     |
| 11  | Reserved | Reserved                                              | -                                                                                                                                                                                                                                                                                                                                                                                                                                                      | -      |
| 10  | ASYNCLKA | Channel A asynchronous count clock source selection   | <ul style="list-style-type: none"> <li>0b0: LRC</li> <li>0b1: XTAL32</li> </ul> <div>  <b>Note</b><br/> When the receive timeout function of the corresponding UART channel is enabled, the clock source is no longer XTAL32 (XTAL32 crystal oscillator may not exist in hardware), but the clock tick generated by the UART baud rate generator. </div>             | RW     |
| 9   | SYNCLKA  | Channel A synchronous counting clock source selection | <ul style="list-style-type: none"> <li>0b0: PCLK1</li> <li>0b1: Internal hardware triggering event</li> </ul>                                                                                                                                                                                                                                                                                                                                          | RW     |
| 8   | SYNSA    | Channel A counting mode selection                     | <ul style="list-style-type: none"> <li>0b0: Synchronous counting mode</li> <li>0b1: Asynchronous counting mode</li> </ul>                                                                                                                                                                                                                                                                                                                              | RW     |
| 7:4 | CKDIVA   | Channel A count clock frequency division selection    | <ul style="list-style-type: none"> <li>0b0000: Clock source</li> <li>0b0001: Clock source/2</li> <li>0b0010: Clock source/4</li> <li>0b0011: Clock source/8</li> <li>0b0100: Clock source/16</li> <li>0b0101: Clock source/32</li> <li>0b0110: Clock source/64</li> <li>0b0111: Clock source/128</li> <li>0b1000: Clock source/256</li> <li>0b1001: Clock source/512</li> <li>0b1010: Clock source/1024</li> <li>Others: Setting prohibited</li> </ul> | RW     |

| Bit | Name   | Function                                           | Description                                                                                                                                                                                                                                                  | Access |
|-----|--------|----------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 7:4 | CKDIVA | Channel A count clock frequency division selection |  <b>Note</b><br>The divided clock source can be various clock sources for asynchronous counting or PCLK1 for synchronous counting.                                          | RW     |
| 3   | OVENA  | Count overflow interrupt enable A                  | Interrupt condition: count overflow, that is, count value (CNTAR)=0xFFFF<br>● 0b0: Disable interrupt<br>● 0b1: Enable interrupt                                                                                                                              | RW     |
| 2   | CMENA  | Count match interrupt enable A                     | Interrupt condition: When the CMPAR register is equal to the count value (CNTAR), or when a capture input event occurs<br>● 0b0: Disable interrupt<br>● 0b1: Enable interrupt                                                                                | RW     |
| 1   | CAPMDA | Pattern selection A                                | ● 0b0: Compare output function<br>● 0b1: Capture input function                                                                                                                                                                                              | RW     |
| 0   | CSTA   | Channel A timer ON                                 | ● 0b0: Channel A timer OFF<br>● 0b1: Channel A timer ON<br><br> <b>Note</b><br>This bit will automatically be set to 0 when the hardware trigger stop condition is enable. | RW     |

**Note**

The internal hardware trigger events (Bit31-Bit28 and Bit15-Bit12) and the XTAL32 clock source (Bit26 and Bit10) during asynchronous counting mentioned in this register are all provided by the USART module when the TIMEOUT function of the USART module is enabled. For details, please refer to the relevant description in the "Universal Synchronous Asynchronous Receiver/Transmitter (USART) " chapter.

## 26.3.5 Status Flag Register (TMR0\_STFLR)

| Offset | Bit Position |    |    |    |    |    |    |      |          |    |    |    |    |    |    |      |      |      |          |    |    |    |   |   |   |      |          |   |   |   |   |   |  |      |      |      |
|--------|--------------|----|----|----|----|----|----|------|----------|----|----|----|----|----|----|------|------|------|----------|----|----|----|---|---|---|------|----------|---|---|---|---|---|--|------|------|------|
| 0x0014 | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24   | 23       | 22 | 21 | 20 | 19 | 18 | 17 | 16   | 15   | 14   | 13       | 12 | 11 | 10 | 9 | 8 | 7 | 6    | 5        | 4 | 3 | 2 | 1 | 0 |  |      |      |      |
| Reset  | 0x01000100   |    |    |    |    |    |    |      |          |    |    |    |    |    |    |      |      |      |          |    |    |    |   |   |   |      |          |   |   |   |   |   |  |      |      |      |
| Name   | Reserved     |    |    |    |    |    |    | SYDB | Reserved |    |    |    |    |    |    | ICPB | OVFB | CMFB | Reserved |    |    |    |   |   |   | SYDA | Reserved |   |   |   |   |   |  | ICPA | OVFA | CMFA |
| Access |              |    |    |    |    |    |    | RO   |          |    |    |    |    |    |    | RW   | RW   | RW   |          |    |    |    |   |   |   | RO   |          |   |   |   |   |   |  | RW   | RW   | RW   |

| Bit   | Name     | Function                          | Description                                                                                                                                                                                                                                                                          | Access |
|-------|----------|-----------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:25 | Reserved | Reserved                          | -                                                                                                                                                                                                                                                                                    | -      |
| 24    | SYDB     | Synchronization Completion Flag B | <ul style="list-style-type: none"> <li>0b0: B channel register is synchronizing during asynchronous counting</li> <li>0b1: B channel register synchronization is completed during asynchronous counting</li> </ul>                                                                   | RO     |
| 23:19 | Reserved | Reserved                          | -                                                                                                                                                                                                                                                                                    | -      |
| 18    | ICPB     | Repeat Capture Flag B             | <ul style="list-style-type: none"> <li>0b0: No capture input occurred after the count match flag bit CMFB was set to 1</li> <li>0b1: A capture input occurred again after the count match flag bit CMFB was set to 1</li> </ul>                                                      | RW     |
| 17    | OVFB     | Count Overflow Flag B             | <ul style="list-style-type: none"> <li>0b0: The count value (CNTBR) does not count to 0xFFFF</li> <li>0b1: The count value (CNTBR) counts to 0xFFFF</li> </ul>                                                                                                                       | RW     |
| 16    | CMFB     | Count Match Flag B                | <ul style="list-style-type: none"> <li>0b0: The value of the CMPBR register is not equal to the count value (CNTBR) and no capture input action occurs</li> <li>0b1: The value of the CMPBR register is equal to the count value (CNTBR) or a capture input action occurs</li> </ul> | RW     |
| 15:9  | Reserved | Reserved                          | -                                                                                                                                                                                                                                                                                    | -      |
| 8     | SYDA     | Synchronization Completion Flag A | <ul style="list-style-type: none"> <li>0b0: A channel register is synchronizing during asynchronous counting</li> <li>0b1: A channel register synchronization is completed during asynchronous counting</li> </ul>                                                                   | RO     |
| 7:3   | Reserved | Reserved                          | -                                                                                                                                                                                                                                                                                    | -      |
| 2     | ICPA     | Repeat Capture Flag A             | <ul style="list-style-type: none"> <li>0b0: No capture input occurred after the count match flag bit CMFA was set to 1</li> <li>0b1: A capture input occurred again after the count match flag bit CMFA was set to 1</li> </ul>                                                      | RW     |
| 1     | OVFA     | Count Overflow Flag A             | <ul style="list-style-type: none"> <li>0b0: The count value (CNTAR) does not count to 0xFFFF</li> <li>0b1: The count value (CNTAR) counts to 0xFFFF</li> </ul>                                                                                                                       | RW     |

| Bit | Name | Function           | Description                                                                                                                                                                                                                                                                           | Access |
|-----|------|--------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 0   | CMFA | Count Match Flag A | <ul style="list-style-type: none"><li>● 0b0: The value of the CMPAR register is not equal to the count value (CNTAR) and no capture input action occurs</li><li>● 0b1: The value of the CMPAR register is equal to the count value (CNTAR) or a capture input action occurs</li></ul> | RW     |

## 26.3.6 Count Synchronization Enable Register 1 (TMR\_SYNENR1)

| Offset | Bit Position |        |        |        |        |        |        |        |          |    |    |    |    |    |        |        |        |          |    |    |    |    |   |          |          |          |          |          |          |          |          |          |          |    |    |
|--------|--------------|--------|--------|--------|--------|--------|--------|--------|----------|----|----|----|----|----|--------|--------|--------|----------|----|----|----|----|---|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----|----|
| 0x14   | 31           | 30     | 29     | 28     | 27     | 26     | 25     | 24     | 23       | 22 | 21 | 20 | 19 | 18 | 17     | 16     | 15     | 14       | 13 | 12 | 11 | 10 | 9 | 8        | 7        | 6        | 5        | 4        | 3        | 2        | 1        | 0        |          |    |    |
| Reset  | 0x00000000   |        |        |        |        |        |        |        |          |    |    |    |    |    |        |        |        |          |    |    |    |    |   |          |          |          |          |          |          |          |          |          |          |    |    |
| Name   | TMR6_8       | TMR6_7 | TMR6_6 | TMR6_5 | TMR6_4 | TMR6_3 | TMR6_2 | TMR6_1 | Reserved |    |    |    |    |    | TMR4_3 | TMR4_2 | TMR4_1 | Reserved |    |    |    |    |   | TMR0_5_B | TMR0_5_A | TMR0_4_B | TMR0_4_A | TMR0_3_B | TMR0_3_A | TMR0_2_B | TMR0_2_A | TMR0_1_B | TMR0_1_A |    |    |
| Access | RW           | RW     | RW     | RW     | RW     | RW     | RW     | RW     |          |    |    |    |    |    | RW     | RW     | RW     |          |    |    |    |    |   | RW       | RW       | RW       | RW       | RW       | RW       | RW       | RW       | RW       | RW       | RW | RW |

| Bit   | Name     | Function                           | Description                                                                                                                                                                          | Access |
|-------|----------|------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31    | TMR6_8   | Timer6_8 software start enable bit | <ul style="list-style-type: none"> <li>0b0: Timer6 unit 8 software start is disabled</li> <li>0b1: Timer6 unit 8 software start is enabled, and the timer starts counting</li> </ul> | RW     |
| 30    | TMR6_7   | Timer6_7 software start enable bit | <ul style="list-style-type: none"> <li>0b0: Timer6 unit 7 software start is disabled</li> <li>0b1: Timer6 unit 7 software start is enabled, and the timer starts counting</li> </ul> | RW     |
| 29    | TMR6_6   | Timer6_6 software start enable bit | <ul style="list-style-type: none"> <li>0b0: Timer6 unit 6 software start is disabled</li> <li>0b1: Timer6 unit 6 software start is enabled, and the timer starts counting</li> </ul> | RW     |
| 28    | TMR6_5   | Timer6_5 software start enable bit | <ul style="list-style-type: none"> <li>0b0: Timer6 unit 5 software start is disabled</li> <li>0b1: Timer6 unit 5 software start is enabled, and the timer starts counting</li> </ul> | RW     |
| 27    | TMR6_4   | Timer6_4 software start enable bit | <ul style="list-style-type: none"> <li>0b0: Timer6 unit 4 software start is disabled</li> <li>0b1: Timer6 unit 4 software start is enabled, and the timer starts counting</li> </ul> | RW     |
| 26    | TMR6_3   | Timer6_3 software start enable bit | <ul style="list-style-type: none"> <li>0b0: Timer6 unit 3 software start is disabled</li> <li>0b1: Timer6 unit 3 software start is enabled, and the timer starts counting</li> </ul> | RW     |
| 25    | TMR6_2   | Timer6_2 software start enable bit | <ul style="list-style-type: none"> <li>0b0: Timer6 unit 2 software start is disabled</li> <li>0b1: Timer6 unit 2 software start is enabled, and the timer starts counting</li> </ul> | RW     |
| 24    | TMR6_1   | Timer6_1 software start enable bit | <ul style="list-style-type: none"> <li>0b0: Timer6 unit 1 software start is disabled</li> <li>0b1: Timer6 unit 1 software start is enabled, and the timer starts counting</li> </ul> | RW     |
| 23:19 | Reserved | Reserved                           | -                                                                                                                                                                                    | -      |
| 18    | TMR4_3   | Timer4_3 software start enable bit | <ul style="list-style-type: none"> <li>0b0: Timer4 unit 3 software start is disabled</li> <li>0b1: Timer4 unit 3 software start is enabled, and the timer starts counting</li> </ul> | RW     |
| 17    | TMR4_2   | Timer4_2 software start enable bit | <ul style="list-style-type: none"> <li>0b0: Timer4 unit 2 software start is disabled</li> <li>0b1: Timer4 unit 2 software start is enabled, and the timer starts counting</li> </ul> | RW     |

| Bit   | Name     | Function                                     | Description                                                                                                                                                                                                                                                                                                                                                                              | Access |
|-------|----------|----------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 16    | TMR4_1   | Timer4_1 software start enable bit           | <ul style="list-style-type: none"> <li>0b0: Timer4 unit 1 software start is disabled</li> <li>0b1: Timer4 unit 1 software start is enabled, and the timer starts counting</li> </ul>                                                                                                                                                                                                     | RW     |
| 15:10 | Reserved | Reserved                                     | -                                                                                                                                                                                                                                                                                                                                                                                        | -      |
| 9     | TMR0_5_B | Timer0_5 channel B software start enable bit | <ul style="list-style-type: none"> <li>0b0: Software start for Timer0 unit 5 Channel B is disabled</li> <li>0b1: Software start for Timer0 unit 5 Channel B is enabled, and the timer starts counting</li> </ul> <div>  <b>Note</b><br/>Please use this function in synchronous counting mode. </div>   | RW     |
| 8     | TMR0_5_A | Timer0_5 channel A software start enable bit | <ul style="list-style-type: none"> <li>0b0: Software start for Timer0 unit 5 Channel A is disabled</li> <li>0b1: Software start for Timer0 unit 5 Channel A is enabled, and the timer starts counting</li> </ul> <div>  <b>Note</b><br/>Please use this function in synchronous counting mode. </div>   | RW     |
| 7     | TMR0_4_B | Timer0_4 channel B software start enable bit | <ul style="list-style-type: none"> <li>0b0: Software start for Timer0 unit 4 Channel B is disabled</li> <li>0b1: Software start for Timer0 unit 4 Channel B is enabled, and the timer starts counting</li> </ul> <div>  <b>Note</b><br/>Please use this function in synchronous counting mode. </div> | RW     |
| 6     | TMR0_4_A | Timer0_4 channel A software start enable bit | <ul style="list-style-type: none"> <li>0b0: Software start for Timer0 unit 4 Channel A is disabled</li> <li>0b1: Software start for Timer0 unit 4 Channel A is enabled, and the timer starts counting</li> </ul> <div>  <b>Note</b><br/>Please use this function in synchronous counting mode. </div> | RW     |
| 5     | TMR0_3_B | Timer0_3 channel B software start enable bit | <ul style="list-style-type: none"> <li>0b0: Software start for Timer0 unit 3 Channel B is disabled</li> <li>0b1: Software start for Timer0 unit 3 Channel B is enabled, and the timer starts counting</li> </ul> <div>  <b>Note</b><br/>Please use this function in synchronous counting mode. </div> | RW     |



| Bit | Name     | Function                                     | Description                                                                                                                                                                                                                                                                                                                                                                               | Access |
|-----|----------|----------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 4   | TMR0_3_A | Timer0_3 channel A software start enable bit | <ul style="list-style-type: none"> <li>0b0: Software start for Timer0 unit 3 Channel A is disabled</li> <li>0b1: Software start for Timer0 unit 3 Channel A is enabled, and the timer starts counting</li> </ul> <div>  <b>Note</b><br/> Please use this function in synchronous counting mode. </div>   | RW     |
| 3   | TMR0_2_B | Timer0_2 channel B software start enable bit | <ul style="list-style-type: none"> <li>0b0: Software start for Timer0 unit 2 Channel B is disabled</li> <li>0b1: Software start for Timer0 unit 2 Channel B is enabled, and the timer starts counting</li> </ul> <div>  <b>Note</b><br/> Please use this function in synchronous counting mode. </div>   | RW     |
| 2   | TMR0_2_A | Timer0_2 channel A software start enable bit | <ul style="list-style-type: none"> <li>0b0: Software start for Timer0 unit 2 Channel A is disabled</li> <li>0b1: Software start for Timer0 unit 2 Channel A is enabled, and the timer starts counting</li> </ul> <div>  <b>Note</b><br/> Please use this function in synchronous counting mode. </div> | RW     |
| 1   | TMR0_1_B | Timer0_1 channel B software start enable bit | <ul style="list-style-type: none"> <li>0b0: Software start for Timer0 unit 1 Channel B is disabled</li> <li>0b1: Software start for Timer0 unit 1 Channel B is enabled, and the timer starts counting</li> </ul> <div>  <b>Note</b><br/> Please use this function in synchronous counting mode. </div> | RW     |
| 0   | TMR0_1_A | Timer0_1 channel A software start enable bit | <ul style="list-style-type: none"> <li>0b0: Software start for Timer0 unit 1 Channel A is disabled</li> <li>0b1: Software start for Timer0 unit 1 Channel A is enabled, and the timer starts counting</li> </ul> <div>  <b>Note</b><br/> Please use this function in synchronous counting mode. </div> | RW     |


**Note**

- Please do not use the count synchronization function in asynchronous counting mode for this timer.
- TMR\_SYNNR1 is not controlled by the RMU\_FRST register.

## 26.3.7 Count Synchronization Enable Register 2 (TMR\_SYNENR2)

| Offset | Bit Position |    |    |    |    |    |    |    |          |          |          |          |          |          |          |          |          |    |    |    |    |    |   |   |         |         |         |        |        |        |        |        |        |        |        |        |    |
|--------|--------------|----|----|----|----|----|----|----|----------|----------|----------|----------|----------|----------|----------|----------|----------|----|----|----|----|----|---|---|---------|---------|---------|--------|--------|--------|--------|--------|--------|--------|--------|--------|----|
| 0x40   | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23       | 22       | 21       | 20       | 19       | 18       | 17       | 16       | 15       | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7       | 6       | 5       | 4      | 3      | 2      | 1      | 0      |        |        |        |        |    |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |          |          |          |          |          |          |          |          |          |    |    |    |    |    |   |   |         |         |         |        |        |        |        |        |        |        |        |        |    |
| Name   | Reserved     |    |    |    |    |    |    |    | TMR2_4_B | TMR2_4_A | TMR2_3_B | TMR2_3_A | TMR2_2_B | TMR2_2_A | TMR2_1_B | TMR2_1_A | Reserved |    |    |    |    |    |   |   | TMRA_12 | TMRA_11 | TMRA_10 | TMRA_9 | TMRA_8 | TMRA_7 | TMRA_6 | TMRA_5 | TMRA_4 | TMRA_3 | TMRA_2 | TMRA_1 |    |
| Access |              |    |    |    |    |    |    |    | RW       | RW       | RW       | RW       | RW       | RW       | RW       | RW       |          |    |    |    |    |    |   |   |         | RW      | RW      | RW     | RW     | RW     | RW     | RW     | RW     | RW     | RW     | RW     | RW |

| Bit   | Name     | Function                            | Description                                                                                                                                                                              | Access |
|-------|----------|-------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:24 | Reserved | Reserved                            | -                                                                                                                                                                                        | -      |
| 23    | TMR2_4_B | Timer2_4 Channel B Start Enable Bit | <ul style="list-style-type: none"> <li>0b0: Timer2 Unit 4 Channel B start is disabled.</li> <li>0b1: Timer2 Unit 4 Channel B start is enabled, and the timer begins counting.</li> </ul> | RW     |
| 22    | TMR2_4_A | Timer2_4 Channel A Start Enable Bit | <ul style="list-style-type: none"> <li>0b0: Timer2 Unit 4 Channel A start is disabled.</li> <li>0b1: Timer2 Unit 4 Channel A start is enabled, and the timer begins counting.</li> </ul> | RW     |
| 21    | TMR2_3_B | Timer2_3 Channel B Start Enable Bit | <ul style="list-style-type: none"> <li>0b0: Timer2 Unit 3 Channel B start is disabled.</li> <li>0b1: Timer2 Unit 3 Channel B start is enabled, and the timer begins counting.</li> </ul> | RW     |
| 20    | TMR2_3_A | Timer2_3 Channel A Start Enable Bit | <ul style="list-style-type: none"> <li>0b0: Timer2 Unit 3 Channel A start is disabled.</li> <li>0b1: Timer2 Unit 3 Channel A start is enabled, and the timer begins counting.</li> </ul> | RW     |
| 19    | TMR2_2_B | Timer2_2 Channel B Start Enable Bit | <ul style="list-style-type: none"> <li>0b0: Timer2 Unit 2 Channel B start is disabled.</li> <li>0b1: Timer2 Unit 2 Channel B start is enabled, and the timer begins counting.</li> </ul> | RW     |
| 18    | TMR2_2_A | Timer2_2 Channel A Start Enable Bit | <ul style="list-style-type: none"> <li>0b0: Timer2 Unit 2 Channel A start is disabled.</li> <li>0b1: Timer2 Unit 2 Channel A start is enabled, and the timer begins counting.</li> </ul> | RW     |
| 17    | TMR2_1_B | Timer2_1 Channel B Start Enable Bit | <ul style="list-style-type: none"> <li>0b0: Timer2 Unit 1 Channel B start is disabled.</li> <li>0b1: Timer2 Unit 1 Channel B start is enabled, and the timer begins counting.</li> </ul> | RW     |
| 16    | TMR2_1_A | Timer2_1 Channel A Start Enable Bit | <ul style="list-style-type: none"> <li>0b0: Timer2 Unit 1 Channel A start is disabled.</li> <li>0b1: Timer2 Unit 1 Channel A start is enabled, and the timer begins counting.</li> </ul> | RW     |
| 15:12 | Reserved | Reserved                            | -                                                                                                                                                                                        | -      |

| Bit | Name    | Function                            | Description                                                                                                                                                                                                                                                                                                                                                   | Access |
|-----|---------|-------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 11  | TMRA_12 | TimerA_12 Software Start Enable Bit | <ul style="list-style-type: none"> <li>0b0: TimerA unit 12 software start is disabled</li> <li>0b1: TimerA unit 12 software start is enabled, and the timer starts counting.</li> </ul> <div>  <b>Note</b><br/>Please use this function in synchronous counting mode. </div> | RW     |
| 10  | TMRA_11 | TimerA_11 Software Start Enable Bit | <ul style="list-style-type: none"> <li>0b0: TimerA unit 11 software start is disabled</li> <li>0b1: TimerA unit 11 software start is enabled, and the timer starts counting.</li> </ul> <div>  <b>Note</b><br/>Please use this function in synchronous counting mode. </div> | RW     |
| 9   | TMRA_10 | TimerA_10 Software Start Enable Bit | <ul style="list-style-type: none"> <li>0b0: TimerA unit 10 software start is disabled</li> <li>0b1: TimerA unit 10 software start is enabled, and the timer starts counting.</li> </ul> <div>  <b>Note</b><br/>Please use this function in synchronous counting mode. </div> | RW     |
| 8   | TMRA_9  | TimerA_9 Software Start Enable Bit  | <ul style="list-style-type: none"> <li>0b0: TimerA unit 9 software start is disabled</li> <li>0b1: TimerA unit 9 software start is enabled, and the timer starts counting.</li> </ul> <div>  <b>Note</b><br/>Please use this function in synchronous counting mode. </div> | RW     |
| 7   | TMRA_8  | TimerA_8 Software Start Enable Bit  | <ul style="list-style-type: none"> <li>0b0: TimerA unit 8 software start is disabled</li> <li>0b1: TimerA unit 8 software start is enabled, and the timer starts counting.</li> </ul> <div>  <b>Note</b><br/>Please use this function in synchronous counting mode. </div> | RW     |
| 6   | TMRA_7  | TimerA_7 Software Start Enable Bit  | <ul style="list-style-type: none"> <li>0b0: TimerA unit 7 software start is disabled</li> <li>0b1: TimerA unit 7 software start is enabled, and the timer starts counting.</li> </ul> <div>  <b>Note</b><br/>Please use this function in synchronous counting mode. </div> | RW     |

| Bit | Name   | Function                           | Description                                                                                                                                                                                                                                                                                                                                                   | Access |
|-----|--------|------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 5   | TMRA_6 | TimerA_6 Software Start Enable Bit | <ul style="list-style-type: none"> <li>0b0: TimerA unit 6 software start is disabled</li> <li>0b1: TimerA unit 6 software start is enabled, and the timer starts counting.</li> </ul> <div>  <b>Note</b><br/>Please use this function in synchronous counting mode. </div>   | RW     |
| 4   | TMRA_5 | TimerA_5 Software Start Enable Bit | <ul style="list-style-type: none"> <li>0b0: TimerA unit 5 software start is disabled</li> <li>0b1: TimerA unit 5 software start is enabled, and the timer starts counting.</li> </ul> <div>  <b>Note</b><br/>Please use this function in synchronous counting mode. </div>   | RW     |
| 3   | TMRA_4 | TimerA_4 Software Start Enable Bit | <ul style="list-style-type: none"> <li>0b0: TimerA unit 4 software start is disabled</li> <li>0b1: TimerA unit 4 software start is enabled, and the timer starts counting.</li> </ul> <div>  <b>Note</b><br/>Please use this function in synchronous counting mode. </div>   | RW     |
| 2   | TMRA_3 | TimerA_3 Software Start Enable Bit | <ul style="list-style-type: none"> <li>0b0: TimerA unit 3 software start is disabled</li> <li>0b1: TimerA unit 3 software start is enabled, and the timer starts counting.</li> </ul> <div>  <b>Note</b><br/>Please use this function in synchronous counting mode. </div> | RW     |
| 1   | TMRA_2 | TimerA_2 Software Start Enable Bit | <ul style="list-style-type: none"> <li>0b0: TimerA unit 2 software start is disabled</li> <li>0b1: TimerA unit 2 software start is enabled, and the timer starts counting.</li> </ul> <div>  <b>Note</b><br/>Please use this function in synchronous counting mode. </div> | RW     |
| 0   | TMRA_1 | TimerA_1 Software Start Enable Bit | <ul style="list-style-type: none"> <li>0b0: TimerA unit 1 software start is disabled</li> <li>0b1: TimerA unit 1 software start is enabled, and the timer starts counting.</li> </ul> <div>  <b>Note</b><br/>Please use this function in synchronous counting mode. </div> | RW     |

 **Note**

1. Please do not use the count synchronization function in asynchronous counting mode for this timer.
2. TMR\_SYNNR2 is not controlled by the RMU\_FRST register.

## 26.4 Precautions

1. During asynchronous counting operation, you must first configure the asynchronous clock source by setting the BCONR.ASYNCCLKA<B> bit, then select the asynchronous counting mode by setting the BCONR.SYNSA<B> bit, and then start Timer0.
2. When asynchronous counting is selected, when the count value (CNTA<B>R), reference value (CMPAA<B>R), start bit (BCONR.CSTA<B>), and status bit (STFLR.CMFA<B>, STFLR.OVFA<B>, STFLR.ICPA<B>) are modified, Timer0 will write the modified value into the corresponding register after three asynchronous counting clocks after receiving the write action.
3. When selecting asynchronous counting, during consecutive write operations to the count value (CNTA<B>R), reference value (CMPA<B>R), start bit (BCONR.CSTA<B>), and status bits (STFLR.CMFA<B>, STFLR.OVFA<B>, STFLR.ICPA<B>), it is necessary to wait until the status bit (STFLR.SYDA<B>) is 1 before performing the next write operation to any of the aforementioned registers.
4. When asynchronous counting is selected, ensure that BCONR.SYNCLKA<B> is set to 0.

## 27 Real Time Clock (RTC)

### 27.1 Introduction

A real time clock (RTC) is a counter that stores time information in BCD format. Record the specific calendar time from 00 to 99. Support 12/24 hour time system, automatically calculate the number of days 28, 29 (leap year), 30 and 31 according to the month and year.

### 27.2 Features

**Table 27-1 RTC Main Features**

|                    |                                                                                                                                                  |
|--------------------|--------------------------------------------------------------------------------------------------------------------------------------------------|
| Count clock source | External low-speed oscillator (32.768kHz)<br>RTC dedicated internal low-speed oscillator (32.768kHz)<br>XTAL fractional frequency division clock |
| Basic function     | BCD code indicates seconds, minutes, hours, days, weeks, months, years                                                                           |
|                    | Software start or stop                                                                                                                           |
|                    | 12/24 hour format optional, leap year automatic recognition                                                                                      |
|                    | Programmable alarm clock                                                                                                                         |
|                    | Distributed/uniform compensation 1Hz clock output                                                                                                |
|                    | Clock error compensation function                                                                                                                |
|                    | Timestamp Function                                                                                                                               |
|                    | Backup register reset function                                                                                                                   |
| Interrupt          | Cycle interrupt                                                                                                                                  |
|                    | Alarm interrupt                                                                                                                                  |
|                    | Intrusion event interrupt                                                                                                                        |

## 27.3 Functional Description

### 27.3.1 RTC Block Diagram

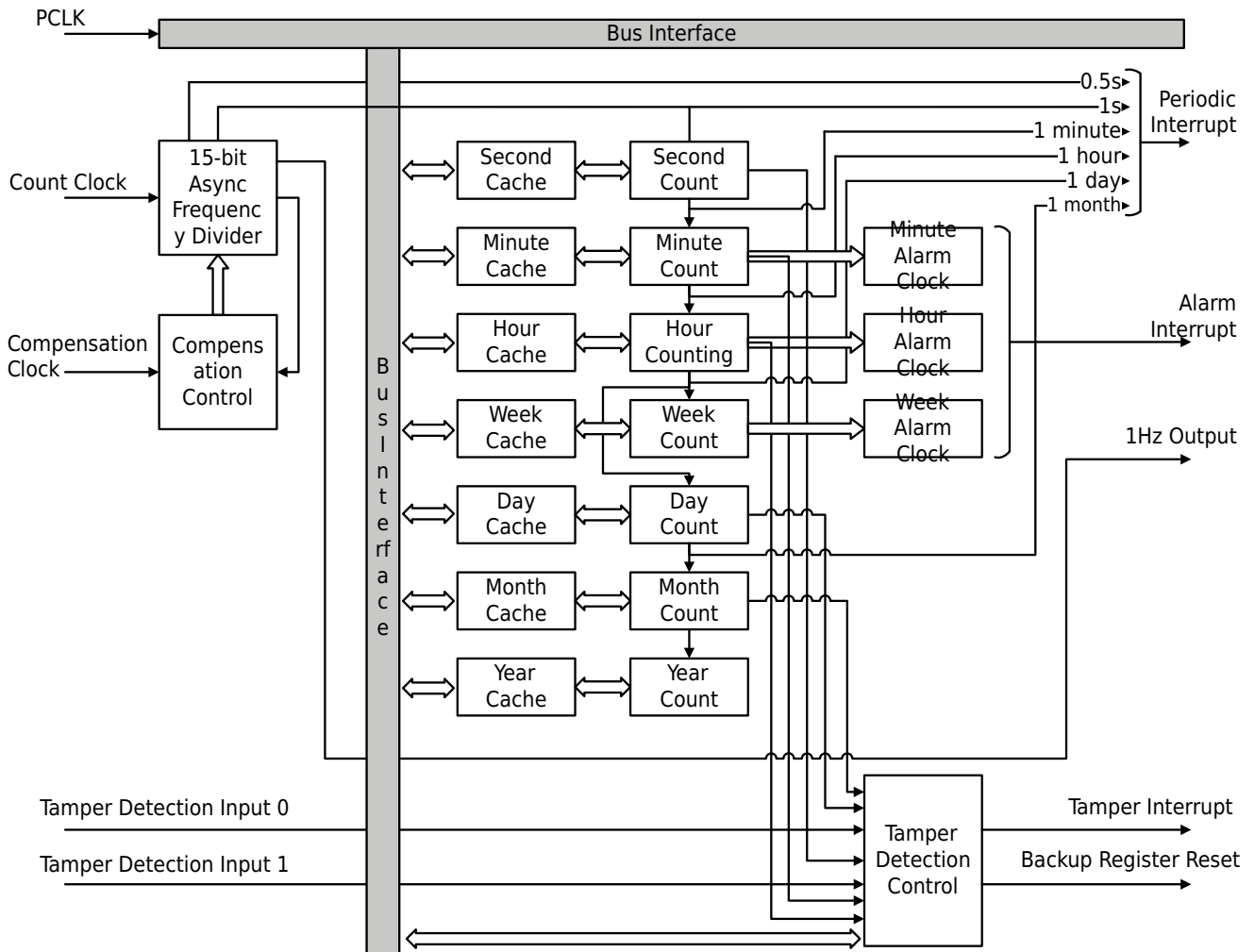


Figure 27-1 RTC Basic Block Diagram

### 27.3.2 Power-on Setting

The RTC register value is uncertain after power-on reset. After power-on reset, you must first set the backup domain reset register PWC\_VBATRSTR, and then set the control register RTC\_CR0.RESET bit to reset all registers. Then set the control register, calendar initial value, alarm setting, and start RTC. After the RTC is started, other external reset requests cannot reset the RTC, and the RTC will always be in the working state. RTC operation can be stopped by setting the RTC\_CR1.START bit in the control register to 0. When setting the RTC, the clock source must be stable.

### 27.3.3 System Low Power Mode Switching

If the system switches to low-power mode immediately after RTC counting starts, please perform any of the following confirmations before switching modes.

- After setting RTC\_CR1.START=1, switch modes after more than 2 RTC count clocks.
- After setting RTC\_CR1.START=1, set RTC\_CR2.RWREQ=1 and query RTC\_CR2.RWEN=1. Set the calendar count register, then set RTC\_CR2.RWREQ=0, query RTC\_CR2.RWEN=0, and switch modes.

### 27.3.4 Clock Error Compensation

Due to deviations of the external low-speed crystal oscillator under varying temperature conditions, error compensation is required when high-accuracy counting results are needed. For the compensation method, refer to the "[Clock Error Compensation Registers \(RTC\\_ERRCRH and RTC\\_ERRCRL\)](#)".

### 27.3.5 Intrusion Detection

When intrusion detection RTC\_TPCRn.TPEN (n=0-1) is enabled, a active intrusion event detected on the external RTC pin can trigger the intrusion event detection function, timestamp function, and backup register reset function respectively.

**Intrusion Event Detection Function:** When an intrusion event occurs, the corresponding intrusion flag of the pin will be set. If the intrusion event continues, the overflow flag will be set. When intrusion event 0 and 1 occur simultaneously, the intrusion event 0 flag is set first, and after one RTC counting clock, the intrusion event 1 flag is set, and the overflow flag is also set. When the intrusion event interrupt enable RTC\_TPCRn.TPIE (n=0-1) is set, an intrusion detection interrupt will occur.

**Timestamp Function:** When RTC\_TPCRn.TSTPE (n=0-1) timestamp function is enabled, the timestamp register will record the second, minute, hour, day, and month calendar time when an intrusion event occurs.

**Backup Register Reset:** When RTC\_TPCRn.TPRSTE (n=0-1) backup register reset is enabled, the backup register will be reset after an intrusion event occurs. The backup registers consist of 32 32-bit registers, used to store 128 bytes of user backup data.



#### Notice

When intrusion detection is enabled, regardless of whether the timestamp function, intrusion event interrupt, or backup register reset function is enabled, the intrusion event flag RTC\_TPSRn.TPFn (n=0-1) will be set after an event occurs.

### 27.3.6 Timestamp Function

When the intrusion detection RTC\_TPCRn.TPEN (n=0-1) is enabled and the timestamp recording function RTC\_TPCRn.TSTPE (n=0-1) is enabled, the timestamp function will be triggered after the external pin of the RTC detects a active intrusion event. The timestamp register value will record the calendar time in seconds, minutes, hours, days and months. A set of timestamp registers corresponds to two intrusion event input pins respectively. When an intrusion event occurs on the pin for the first time, the intrusion flag is set and the timestamp is recorded. When an intrusion event occurs again, the overflow flag is set, but the timestamp register still maintains the first recorded value. After the intrusion flag register is cleared, the timestamp will be recorded again when an intrusion detection occurs.

### 27.3.7 Interrupt

RTC supports 3 interrupt types. Timing alarm interrupt, periodic interrupt and intrusion detection interrupt.

#### Alarm Interrupt

Alarm clock interrupt RTC\_ALM, when RTC\_CR2.ALMIE=1 and RTC\_CR2.ALME=1, if the current calendar time and the minute alarm clock register (RTC\_ALMMIN), the hour alarm clock register (RTC\_ALMHOUR), and the weekly alarm clock register (RTC\_ALM WEEK) equal, trigger the alarm interrupt. The alarm clock configures an independent flag register bit RTC\_CR2.ALMF, write 0 to the RTC\_CR2.ALMF bit to clear the alarm clock flag.



## Periodic Interrupt

Periodic interrupt RTC\_PRD, when RTC\_CR2.PRDIEN=1, after the selected period occurs, a periodic wake-up interrupt is triggered. The periodic interrupt is configured with an independent flag register RTC\_CR2.PRDF, and the periodic flag can be cleared by writing 0 to the RTC\_CR2.PRDF bit.

## Intrusion Detection Interrupt

When the intrusion detection is enabled RTC\_TPCRn.TPEN (n=0-1) and the intrusion event interrupt is enabled RTC\_TPCRn.TPIE (n=0-1), an intrusion event interrupt will occur after the external pin of the RTC detects a active intrusion event. Since the two groups of intrusion detection share one interrupt, they can be distinguished by the intrusion event detection interrupt flag bit RTC\_TPSR.TPFn (n=0-1).

# 27.4 Operation Example

## 27.4.1 RTC Count Start Setting

- Step1.** After power-on, first set the backup domain reset register PWC\_VBATRSTR, then set RTC\_CR0.RESET=0. After confirming that the RESET bit is 0, set RTC\_CR0.RESET=1 to reset all registers;
- Step2.** Set RTC\_CR1.START=0, stop counting;
- Step3.** Set the system clock register, turn on the external low-speed oscillator, then set RTC\_CR3 to select the RTC count clock source;
- Step4.** Set RTC\_CR1, set time, period, 1Hz clock output;
- Step5.** Set the calendar counting registers for seconds, minutes, hours, weeks, days, months, and years;
- Step6.** When clock error compensation is required, set the counting clock error compensation registers RTC\_ERRCRL, RTC\_ERRCRH;
- Step7.** Clear the flag register bits in registers RTC\_CR2, RTC\_TPSR, and enable interrupts;
- Step8.** Set RTC\_CR1.START=1, counting starts.

## 27.4.2 Read Count Register

- Step1.** After RTC\_CR1.START=1 is set, after more than 2 RTC count clocks, set RTC\_CR2.RWREQ=1 to make a calendar register read request;
- Step2.** Query until RTC\_CR2.RWEN=1;
- Step3.** Read all or part of the seconds, minutes, hours, weeks, days, months, and years counter register values;
- Step4.** Set RTC\_CR2.RWREQ=0;
- Step5.** Query until RTC\_CR2.RWEN=0.

## 27.4.3 Write Count Register

- Step1.** After setting RTC\_CR1.START=1, after more than 2 RTC counting clocks, set RTC\_CR2.RWREQ=1 to make a calendar register write request;
- Step2.** Query until RTC\_CR2.RWEN=1;
- Step3.** Write all or part of the seconds, minutes, hours, weeks, days, months, and years counter register values;
- Step4.** Set RTC\_CR2.RWREQ=0. Note that all write operations must be completed within 1 second;
- Step5.** Query until RTC\_CR2.RWEN=0.

### 27.4.4 Alarm Clock Setting

- Step1.** Set RTC\_CR2.ALME=0, the alarm clock is disabled;
- Step2.** Set RTC\_CR2.ALMIE=1, the alarm clock interruption permission;
- Step3.** Minute alarm clock RTC\_ALMMIN, hour alarm clock RTC\_ALMHOUR, week alarm clock RTC\_ALM-WEEK setting;
- Step4.** Set RTC\_CR2.ALME=1, alarm clock permission;
- Step5.** Wait for the alarm to interrupt;
- Step6.** When an alarm occurs, RTC\_CR2.ALMF=1, and the alarm interrupt processing begins.

### 27.4.5 1Hz Output

The RTC can output a 1Hz clock and provide three precision output methods:

1. Ordinary precision 1Hz output without clock compensation;
2. Distributed compensation 1Hz output with average compensation within every 32 seconds;
3. Uniform compensation 1Hz output per second.

When the clock error compensation function is enable RTC\_ERRCRH.COMPEN=1, the distributed compensation 1Hz output and the uniform compensation 1Hz output can be selected.

#### Ordinary Precision 1Hz Output

The 1Hz output of ordinary precision is set as follows:

- Step1.** Set RTC\_CR0.RESET=0, after confirming that the RESET bit is 0, set RTC\_CR0.RESET=1, reset the calendar count register;
- Step2.** Set RTC\_CR1.START=0, counting stops;
- Step3.** 1Hz output pin setting;
- Step4.** RTC\_CR1.ONEHZOE=1, clock output permission;
- Step5.** Set RTC\_CR1.START=1, start counting;
- Step6.** Wait for more than 2 count periods;
- Step7.** 1Hz output starts.

#### Distributed Compensation 1Hz Output

The distributed compensation 1Hz output setting is as follows:

- Step1.** Set RTC\_CR0.RESET=0, after confirming that the RESET bit is 0, set RTC\_CR0.RESET=1, reset the calendar count register;
- Step2.** Set RTC\_CR1.START=0, counting stops;
- Step3.** 1Hz output pin setting;
- Step4.** RTC\_CR1.ONEHZOE=1, clock output permission;
- Step5.** RTC\_CR1.ONEHZSEL=0, select to output distributed compensation 1Hz clock;
- Step6.** Clock error compensation register RTC\_ERRCRL.COMP[7:0] and RTC\_ERRCRH.COMP[8] compensation number setting;
- Step7.** Clock error compensation register RTC\_ERRCRH.COMPEN=1, error compensation is enabled;
- Step8.** Set RTC\_CR1.START=1, start counting;
- Step9.** Wait for more than 2 count periods;
- Step10.** 1Hz output starts.

#### Uniform Compensation 1Hz Output

Uniform compensation 1Hz output setting is as follows:

- Step1.** Set RTC\_CR0.RESET=0, after confirming that the RESET bit is 0, set RTC\_CR0.RESET=1, reset the calendar count register;
- Step2.** Set RTC\_CR1.START=0, counting stops;
- Step3.** RTC output pin setting;
- Step4.** RTC\_CR1.ONEHZOE=1, clock output permission;
- Step5.** RTC\_CR1.ONEHZSEL=1, select output uniform compensation 1Hz clock;
- Step6.** Clock error compensation register RTC\_ERRCRL.COMP[7:0] and RTC\_ERRCRH.COMP[8] compensation number setting;
- Step7.** Clock error compensation register RTC\_ERRCRH.COMPEN=1, accuracy compensation is enabled;
- Step8.** Set RTC\_CR1.START=1, start counting;
- Step9.** Wait for more than 2 count periods;
- Step10.** 1Hz output starts.

## 27.5 Register Description

### 27.5.1 Register List

Base address: 0x4004C000

Table 27-2 Real Time Clock (RTC) Register Offset

| Offset | Register    | Description                         | Width |
|--------|-------------|-------------------------------------|-------|
| 0x0000 | RTC_CR0     | Control Register 0                  | 8     |
| 0x0004 | RTC_CR1     | Control Register 1                  | 8     |
| 0x0008 | RTC_CR2     | Control Register 2                  | 8     |
| 0x000C | RTC_CR3     | Control Register 3                  | 8     |
| 0x0010 | RTC_SEC     | Second Count Register               | 8     |
| 0x0014 | RTC_MIN     | Minute Count Register               | 8     |
| 0x0018 | RTC_HOUR    | Hour Count Register                 | 8     |
| 0x001C | RTC_WEEK    | Week Count Register                 | 8     |
| 0x0020 | RTC_DAY     | Day Count Register                  | 8     |
| 0x0024 | RTC_MON     | Month Count Register                | 8     |
| 0x0028 | RTC_YEAR    | Year Count Register                 | 8     |
| 0x002C | RTC_ALMMIN  | Minute Alarm register               | 8     |
| 0x0030 | RTC_ALMHOUR | Hour Alarm Register                 | 8     |
| 0x0034 | RTC_ALMWEEK | Week Alarm Register                 | 8     |
| 0x0038 | RTC_ERRCRH  | Clock Error Compensation Register H | 8     |
| 0x003C | RTC_ERRCRL  | Clock Error Compensation Register L | 8     |
| 0x0040 | RTC_TPCR0   | Intrusion Control Register 0        | 8     |
| 0x0044 | RTC_TPCR1   | Intrusion Control Register 1        | 8     |
| 0x0048 | RTC_TPSR    | Intrusion Status Register           | 8     |
| 0x004C | RTC_SECTP   | Second Timestamp Register           | 8     |
| 0x0050 | RTC_MINTP   | Minute Timestamp Register           | 8     |
| 0x0054 | RTC_HOURTP  | Hour Timestamp Register             | 8     |
| 0x0058 | RTC_DAYTP   | Day Timestamp Register              | 8     |
| 0x005C | RTC_MONTP   | Month Timestamp Register            | 8     |

## 27.5.2 Control Register 0 (RTC\_CR0)

| Offset | Bit Position |   |   |   |   |   |   |       |
|--------|--------------|---|---|---|---|---|---|-------|
| 0x0000 | 7            | 6 | 5 | 4 | 3 | 2 | 1 | 0     |
| Reset  | 0xXX         |   |   |   |   |   |   |       |
| Name   | Reserved     |   |   |   |   |   |   | RESET |
| Access |              |   |   |   |   |   |   | RW    |

| Bit | Name     | Function                   | Description                                                                                                                                                                                                                      | Access |
|-----|----------|----------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 7:1 | Reserved | Reserved                   | -                                                                                                                                                                                                                                | -      |
| 0   | RESET    | RTC Calendar Counter Reset | <p>Write 1 to initialize all RTC registers; write 0 has no effect.</p> <p>Read:</p> <ul style="list-style-type: none"> <li>0b0: Normal counting status or RTC software reset ends</li> <li>0b1: RTC is in reset state</li> </ul> | RW     |

## 27.5.3 Control Register 1 (RTC\_CR1)

| Offset | Bit Position |          |         |          |      |      |   |   |
|--------|--------------|----------|---------|----------|------|------|---|---|
| 0x0004 | 7            | 6        | 5       | 4        | 3    | 2    | 1 | 0 |
| Reset  | 0xXX         |          |         |          |      |      |   |   |
| Name   | START        | ONEHZSEL | ONEHZOE | Reserved | AMPM | PRDS |   |   |
| Access | RW           | RW       | RW      |          | RW   | RW   |   |   |

| Bit | Name            | Function                   | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | Access |
|-----|-----------------|----------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 7   | START           | RTC counting start         | <ul style="list-style-type: none"> <li>0b0: RTC counting stop</li> <li>0b1: RTC counting start</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | RW     |
| 6   | ONEHZSEL        | 1Hz output selection       | <ul style="list-style-type: none"> <li>0b0: Distributed compensation 1Hz output</li> <li>0b1: Uniform compensation 1Hz output</li> </ul> <div>  <b>Note</b><br/>           When RTC_ERRCRH.COMPEN=1, this bit setting is enable.         </div>                                                                                                                                                                                                                                                                                                                                     | RW     |
| 5   | ONEHZOE         | 1Hz output permission      | <ul style="list-style-type: none"> <li>0b0: 1Hz output disabled</li> <li>0b1: 1Hz output permission</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | RW     |
| 4   | <i>Reserved</i> | <i>Reserved</i>            | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | -      |
| 3   | AMPM            | Time format selection      | <ul style="list-style-type: none"> <li>0b0: 12-hour time format</li> <li>0b1: 24-hour time format</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | RW     |
| 2:0 | PRDS            | Period interrupt selection | Period selection settings: <ul style="list-style-type: none"> <li>0b000: No periodic interruption occurs</li> <li>0b001: 0.5 second</li> <li>0b010: 1 second</li> <li>0b011: 1 minute</li> <li>0b100: 1 hour</li> <li>0b101: 1 day (Every day at 00:00 00)</li> <li>0b110/0b111: 1 month (Every 1st of each month at 00:00 00)</li> </ul> <div>  <b>Notice</b><br/>           When writing period selection during START=1 counting, please disable the period interrupt permission to prevent malfunction. And the relevant flag should be cleared after writing.         </div> | RW     |

## 27.5.4 Control Register 2 (RTC\_CR2)

| Offset | Bit Position |       |       |          |      |      |      |       |
|--------|--------------|-------|-------|----------|------|------|------|-------|
| 0x0008 | 7            | 6     | 5     | 4        | 3    | 2    | 1    | 0     |
| Reset  | 0xXX         |       |       |          |      |      |      |       |
| Name   | ALME         | ALMIE | PRDIE | Reserved | ALMF | PRDF | RWEN | RWREQ |
| Access | RW           | RW    | RW    |          | RW   | RW   | RW   | RW    |

| Bit | Name            | Function                   | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                    | Access |
|-----|-----------------|----------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 7   | ALME            | Alarm Function Enabled     | <ul style="list-style-type: none"> <li>0b0: Alarm function disabled</li> <li>0b1: Alarm function enabled</li> </ul>                                                                                                                                                                                                                                                                                                                                            | RW     |
| 6   | ALMIE           | Alarm Interrupt Enabled    | <ul style="list-style-type: none"> <li>0b0: Alarm interrupt disabled</li> <li>0b1: Alarm interrupt enabled</li> </ul>                                                                                                                                                                                                                                                                                                                                          | RW     |
| 5   | PRDIE           | Periodic Interrupt Enabled | <ul style="list-style-type: none"> <li>0b0: Periodic interrupt disabled</li> <li>0b1: Periodic interrupt enabled</li> </ul>                                                                                                                                                                                                                                                                                                                                    | RW     |
| 4   | <i>Reserved</i> | <i>Reserved</i>            | -                                                                                                                                                                                                                                                                                                                                                                                                                                                              | -      |
| 3   | ALMF            | Alarm Clock flag           | <p>Write 0 to clear the flag, write 1 has no effect</p> <p>Read:</p> <ul style="list-style-type: none"> <li>0b0: Alarm mismatch</li> <li>0b1: Alarm match</li> </ul> <div>  <b>Note</b><br/>           It is enable when ALME=1. When the alarm matches, it is set to 1 after one count clock.         </div>                                                               | RW     |
| 2   | PRDF            | Periodic flag              | <p>Write 0 to clear the flag, write 1 has no effect</p> <p>Read:</p> <ul style="list-style-type: none"> <li>0b0: Periodic does not occur</li> <li>0b1: Periodic occur</li> </ul> <div>  <b>Note</b><br/>           This bit is set to 1 after the setting period occurs.         </div>                                                                                     | RW     |
| 1   | RWEN            | Read/Write Enabled         | <ul style="list-style-type: none"> <li>0b0: Read/write disabled</li> <li>0b1: Read/write enabled</li> </ul> <div>  <b>Note</b><br/>           Calendar register read and write enable flag. Please confirm whether this bit is 1 before reading/writing. The calendar registers include second, minute, hour, week, day, month, and year counting registers.         </div> | RW     |

| Bit | Name  | Function           | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | Access |
|-----|-------|--------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 0   | RWREQ | Read/Write Request | <ul style="list-style-type: none"> <li>● 0b0: Normal counting mode</li> <li>● 0b1: Read/write request</li> </ul> <div>  <b>Note</b><br/>           When reading/writing the calendar register, please set this bit to 1 to request reading and writing. Since the counter is counting continuously, please complete the reading/writing operation within 1 second and clear this bit to 0.<br/><br/>           If the RWREQ bit is set but not cleared within 1 second, the counter will stop counting and the real-time clock RTC will not be able to accurately time. If the RWREQ bit remains set, the real time clock RTC will stop timing.         </div> | RW     |



## 27.5.5 Control Register 3 (RTC\_CR3)

| Offset | Bit Position |   |          |       |          |   |   |   |
|--------|--------------|---|----------|-------|----------|---|---|---|
| 0x000C | 7            | 6 | 5        | 4     | 3        | 2 | 1 | 0 |
| Reset  | 0xXX         |   |          |       |          |   |   |   |
| Name   | RCKSEL       |   | Reserved | LRCEN | Reserved |   |   |   |
| Access | RW           |   |          | RW    |          |   |   |   |

| Bit | Name            | Function                              | Description                                                                                                                                                                                                                                                                                                                                                                                                                                            | Access |
|-----|-----------------|---------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 7:6 | RCKSEL          | RTC count clock selection             | <ul style="list-style-type: none"> <li>0b00: External low-speed oscillator (XTAL32)</li> <li>0b01: XTAL fractional frequency division clock</li> <li>0b10: RTC dedicated internal low-speed oscillator (RTCLRC)</li> <li>0b11: Setting prohibited</li> </ul>                                                                                                                                                                                           | RW     |
| 5   | <i>Reserved</i> | <i>Reserved</i>                       | -                                                                                                                                                                                                                                                                                                                                                                                                                                                      | -      |
| 4   | LRCEN           | Internal low speed oscillator enabled | <ul style="list-style-type: none"> <li>0b0: RTC dedicated internal low-speed oscillator RTCLRC stopped</li> <li>0b1: RTC dedicated internal low-speed oscillator RTCLRC works</li> </ul> <div>  <b>Note</b><br/>           When the RTC dedicated internal low-speed oscillator is used as the RTC clock source, please set the LRCEN bit to enable.         </div> | RW     |
| 3:0 | <i>Reserved</i> | <i>Reserved</i>                       | -                                                                                                                                                                                                                                                                                                                                                                                                                                                      | -      |

### 27.5.6 Second Count Register (RTC\_SEC)

| Offset | Bit Position |      |   |   |      |   |   |   |
|--------|--------------|------|---|---|------|---|---|---|
| 0x0010 | 7            | 6    | 5 | 4 | 3    | 2 | 1 | 0 |
| Reset  | 0xXX         |      |   |   |      |   |   |   |
| Name   | Reserved     | SECD |   |   | SECU |   |   |   |
| Access |              | RW   |   |   | RW   |   |   |   |

| Bit | Name            | Function        | Description              | Access |
|-----|-----------------|-----------------|--------------------------|--------|
| 7   | <i>Reserved</i> | <i>Reserved</i> | -                        | -      |
| 6:4 | SECD            | Second Tens     | Second tens count value  | RW     |
| 3:0 | SECU            | Second Units    | Second units count value | RW     |

Indicates 0-59 seconds, using decimal counting. Please enter a decimal BCD code of 0 to 59. Values outside the range will be ignored.

### 27.5.7 Minute Count Register (RTC\_MIN)

| Offset | Bit Position |      |   |   |      |   |   |   |
|--------|--------------|------|---|---|------|---|---|---|
| 0x0014 | 7            | 6    | 5 | 4 | 3    | 2 | 1 | 0 |
| Reset  | 0xXX         |      |   |   |      |   |   |   |
| Name   | Reserved     | MIND |   |   | MINU |   |   |   |
| Access |              | RW   |   |   | RW   |   |   |   |

| Bit | Name            | Function        | Description              | Access |
|-----|-----------------|-----------------|--------------------------|--------|
| 7   | <i>Reserved</i> | <i>Reserved</i> | -                        | -      |
| 6:4 | MIND            | Minute Tens     | Minute tens count value  | RW     |
| 3:0 | MINU            | Minute Units    | Minute units count value | RW     |

Indicates 0-59 minutes, using decimal counting. Please enter a decimal BCD code of 0 to 59. Values outside the range will be ignored.

### 27.5.8 Hour Count Register (RTC\_HOUR)

| Offset | Bit Position |   |       |   |       |   |   |   |
|--------|--------------|---|-------|---|-------|---|---|---|
| 0x0018 | 7            | 6 | 5     | 4 | 3     | 2 | 1 | 0 |
| Reset  | 0xXX         |   |       |   |       |   |   |   |
| Name   | Reserved     |   | HOURD |   | HOURU |   |   |   |
| Access |              |   | RW    |   | RW    |   |   |   |

| Bit | Name     | Function   | Description            | Access |
|-----|----------|------------|------------------------|--------|
| 7:6 | Reserved | Reserved   | -                      | -      |
| 5:4 | HOURD    | Hour Tens  | Hour tens count value  | RW     |
| 3:0 | HOURU    | Hour Units | Hour units count value | RW     |

In 24-hour format, it means 0-23 hours.

In 12-hour format, it means 0-12 hours. Bit5=0 means AM, then 01-12 means morning; Bit5=1 means PM, then 01-12 means afternoon.

Please set the correct decimal 0-23 or 01-12, 01-12 BCD code according to the value of the control bit RTC\_CR1.AMPM. The value written out of range will be ignored.

For specific time values, please refer to the following table:

| 24-hour format (AMPM=1) |                | 12-hour format (AMPM=0) |                |
|-------------------------|----------------|-------------------------|----------------|
| Time                    | Register Value | Time                    | Register Value |
| 00 o'clock              | 0x00           | AM 12 o'clock           | 0x12           |
| 01 o'clock              | 0x01           | AM 01 o'clock           | 0x01           |
| 02 o'clock              | 0x02           | AM 02 o'clock           | 0x02           |
| 03 o'clock              | 0x03           | AM 03 o'clock           | 0x03           |
| 04 o'clock              | 0x04           | AM 04 o'clock           | 0x04           |
| 05 o'clock              | 0x05           | AM 05 o'clock           | 0x05           |
| 06 o'clock              | 0x06           | AM 06 o'clock           | 0x06           |
| 07 o'clock              | 0x07           | AM 07 o'clock           | 0x07           |
| 08 o'clock              | 0x08           | AM 08 o'clock           | 0x08           |
| 09 o'clock              | 0x09           | AM 09 o'clock           | 0x09           |
| 10 o'clock              | 0x10           | AM 10 o'clock           | 0x10           |
| 11 o'clock              | 0x11           | AM 11 o'clock           | 0x11           |
| 12 o'clock              | 0x12           | PM 12 o'clock           | 0x32           |
| 13 o'clock              | 0x13           | PM 01 o'clock           | 0x21           |

| 24-hour format (AMPM=1) |                | 12-hour format (AMPM=0) |                |
|-------------------------|----------------|-------------------------|----------------|
| Time                    | Register Value | Time                    | Register Value |
| 14 o'clock              | 0x14           | PM 02 o'clock           | 0x22           |
| 15 o'clock              | 0x15           | PM 03 o'clock           | 0x23           |
| 16 o'clock              | 0x16           | PM 04 o'clock           | 0x24           |
| 17 o'clock              | 0x17           | PM 05 o'clock           | 0x25           |
| 18 o'clock              | 0x18           | PM 06 o'clock           | 0x26           |
| 19 o'clock              | 0x19           | PM 07 o'clock           | 0x27           |
| 20 o'clock              | 0x20           | PM 08 o'clock           | 0x28           |
| 21 o'clock              | 0x21           | PM 09 o'clock           | 0x29           |
| 22 o'clock              | 0x22           | PM 10 o'clock           | 0x30           |
| 23 o'clock              | 0x23           | PM 11 o'clock           | 0x31           |

### 27.5.9 Day Count Register (RTC\_DAY)

| Offset | Bit Position |   |      |   |      |   |   |   |
|--------|--------------|---|------|---|------|---|---|---|
| 0x0020 | 7            | 6 | 5    | 4 | 3    | 2 | 1 | 0 |
| Reset  | 0xXX         |   |      |   |      |   |   |   |
| Name   | Reserved     |   | DAYD |   | DAYU |   |   |   |
| Access |              |   | RW   |   | RW   |   |   |   |

| Bit | Name     | Function  | Description           | Access |
|-----|----------|-----------|-----------------------|--------|
| 7:6 | Reserved | Reserved  | -                     | -      |
| 5:4 | DAYD     | Day Tens  | Day tens count value  | RW     |
| 3:0 | DAYU     | Day Units | Day units count value | RW     |

Represents 1 to 31 days, using decimal counting, automatically calculate leap years and months. Write BCD codes corresponding to decimal values 1-28, 1-29, 1-30, or 1-31. Values outside the range will be ignored.

Refer to the table below for the specific range of day values.

| Month                                                   | Day Count Value |
|---------------------------------------------------------|-----------------|
| February (normal year)                                  | 01-28           |
| February (leap year)                                    | 01-29           |
| April, June, September, November                        | 01-30           |
| January, March, May, July, August, October and December | 01-31           |

## 27.5.10 Week Count Register (RTC\_WEEK)

| Offset | Bit Position |   |   |   |   |      |   |   |
|--------|--------------|---|---|---|---|------|---|---|
| 0x001C | 7            | 6 | 5 | 4 | 3 | 2    | 1 | 0 |
| Reset  | 0xXX         |   |   |   |   |      |   |   |
| Name   | Reserved     |   |   |   |   | WEEK |   |   |
| Access |              |   |   |   |   | RW   |   |   |

| Bit | Name     | Function | Description      | Access |
|-----|----------|----------|------------------|--------|
| 7:3 | Reserved | Reserved | -                | -      |
| 2:0 | WEEK     | Week     | Week count value | RW     |

Decimal 0-6 means Sunday-Saturday. Please enter a decimal BCD code of 0 to 6. Values outside the range will be ignored.

For specific week count values, please refer to the following table:

| Week      | Week Count |
|-----------|------------|
| Sunday    | 0x00       |
| Monday    | 0x01       |
| Tuesday   | 0x02       |
| Wednesday | 0x03       |
| Thursday  | 0x04       |
| Friday    | 0x05       |
| Saturday  | 0x06       |

27.5.11 Month Count Register (RTC\_MON)

| Offset | Bit Position |   |   |     |   |   |   |   |
|--------|--------------|---|---|-----|---|---|---|---|
| 0x0024 | 7            | 6 | 5 | 4   | 3 | 2 | 1 | 0 |
| Reset  | 0xXX         |   |   |     |   |   |   |   |
| Name   | Reserved     |   |   | MON |   |   |   |   |
| Access |              |   |   | RW  |   |   |   |   |

| Bit | Name     | Function | Description       | Access |
|-----|----------|----------|-------------------|--------|
| 7:5 | Reserved | Reserved | -                 | -      |
| 4:0 | MON      | Month    | Month count value | RW     |

Indicates 1-12 months, using decimal counting. Write BCD-coded values corresponding to decimal 1-12. Values outside the range will be ignored.

### 27.5.12 Year Count Register (RTC\_YEAR)

| Offset | Bit Position |   |   |   |       |   |   |   |
|--------|--------------|---|---|---|-------|---|---|---|
| 0x0028 | 7            | 6 | 5 | 4 | 3     | 2 | 1 | 0 |
| Reset  | 0xXX         |   |   |   |       |   |   |   |
| Name   | YEAR         |   |   |   | YEARU |   |   |   |
| Access | RW           |   |   |   | RW    |   |   |   |

| Bit | Name  | Function   | Description            | Access |
|-----|-------|------------|------------------------|--------|
| 7:4 | YEARD | Year Tens  | Year tens count value  | RW     |
| 3:0 | YEARU | Year Units | Year units count value | RW     |

Indicates a year value from 0 to 99 in decimal counting, incremented based on month overflow. Automatically calculate leap years such as: 00, 04, 08,..., 92, 96, etc. Write BCD-coded values corresponding to decimal 0-99. Values outside the range will be ignored.

### 27.5.13 Minute Alarm Register (RTC\_ALMMIN)

| Offset | Bit Position |         |   |   |   |         |   |   |
|--------|--------------|---------|---|---|---|---------|---|---|
| 0x002C | 7            | 6       | 5 | 4 | 3 | 2       | 1 | 0 |
| Reset  | 0xXX         |         |   |   |   |         |   |   |
| Name   | Reserved     | ALMMIND |   |   |   | ALMMINU |   |   |
| Access |              | RW      |   |   |   | RW      |   |   |

| Bit | Name     | Function           | Description                    | Access |
|-----|----------|--------------------|--------------------------------|--------|
| 7   | Reserved | Reserved           | -                              | -      |
| 6:4 | ALMMIND  | Minute Alarm Tens  | Minute Alarm Tens count value  | RW     |
| 3:0 | ALMMINU  | Minute Alarm Units | Minute Alarm Units count value | RW     |

Indicates 0-59 points, using decimal counting. Please write the BCD code of decimal 0-59, and the alarm clock matching will not occur if you write the value out of range.

### 27.5.14 Hour Alarm Register (RTC\_ALMHOUR)

| Offset | Bit Position |   |          |   |          |   |   |   |
|--------|--------------|---|----------|---|----------|---|---|---|
| 0x0030 | 7            | 6 | 5        | 4 | 3        | 2 | 1 | 0 |
| Reset  | 0xXX         |   |          |   |          |   |   |   |
| Name   | Reserved     |   | ALMHOURD |   | ALMHOURU |   |   |   |
| Access |              |   | RW       |   | RW       |   |   |   |

| Bit | Name     | Function         | Description                  | Access |
|-----|----------|------------------|------------------------------|--------|
| 7:6 | Reserved | Reserved         | -                            | -      |
| 5:4 | ALMHOURD | Hour Alarm Tens  | Hour Alarm Tens count value  | RW     |
| 3:0 | ALMHOURU | Hour Alarm Units | Hour Alarm Units count value | RW     |

Depending on the value of the control bit RTC\_CR1.AMPM, set the correct decimal 0~23 or 01~ 12,21 ~32 BCD code. If an out-of-range value is written, the alarm match will not occur.

### 27.5.15 Week Alarm Register (RTC\_ALMWEEK)

| Offset | Bit Position |         |   |   |   |   |   |   |
|--------|--------------|---------|---|---|---|---|---|---|
| 0x0034 | 7            | 6       | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0xXX         |         |   |   |   |   |   |   |
| Name   | Reserved     | ALMWEEK |   |   |   |   |   |   |
| Access |              | RW      |   |   |   |   |   |   |

| Bit | Name     | Function   | Description                                                                                                                                                                                                                                                                                | Access |
|-----|----------|------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 7   | Reserved | Reserved   | -                                                                                                                                                                                                                                                                                          | -      |
| 6:0 | ALMWEEK  | Week Alarm | Week Alarm Match Value<br>Bit0-Bit6 correspond to Sunday-Saturday respectively. When the corresponding position is 1, it means that the alarm is enable on that day of the week.<br>For example: Setting Bit0=1 and Bit5=1 means that the alarm settings on Sunday and Friday are enabled. | RW     |

Decimal 0-6 means Sunday-Saturday. Please enter a decimal BCD code of 0 to 6. Values outside the range will not trigger an alarm match.



## 27.5.16 Clock Error Compensation Register (RTC\_ERRCRH, RTC\_ERRCRL)

## RTC\_ERRCRH

| Offset | Bit Position |          |   |   |   |   |   |         |
|--------|--------------|----------|---|---|---|---|---|---------|
| 0x0038 | 7            | 6        | 5 | 4 | 3 | 2 | 1 | 0       |
| Reset  | 0xXX         |          |   |   |   |   |   |         |
| Name   | COMPEN       | Reserved |   |   |   |   |   | COMP[8] |
| Access | RW           |          |   |   |   |   |   | RW      |

| Bit | Name     | Function            | Description                                                                                                                             | Access |
|-----|----------|---------------------|-----------------------------------------------------------------------------------------------------------------------------------------|--------|
| 7   | COMPEN   | Compensation enable | <ul style="list-style-type: none"> <li>0b0: Clock error compensation disabled</li> <li>0b1: Clock error compensation enabled</li> </ul> | RW     |
| 6:1 | Reserved | Reserved            | -                                                                                                                                       | -      |
| 0   | COMP[8]  | Compensation value  | Set the compensation value together with COMP[7:0]                                                                                      | RW     |

## RTC\_ERRCRL

| Offset | Bit Position |   |   |   |   |   |   |   |
|--------|--------------|---|---|---|---|---|---|---|
| 0x003C | 7            | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0xXX         |   |   |   |   |   |   |   |
| Name   | COMP[7:0]    |   |   |   |   |   |   |   |
| Access | RW           |   |   |   |   |   |   |   |

| Bit | Name      | Function           | Description                                                                                                                                                                                                                                                                                                                                                   | Access |
|-----|-----------|--------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 7:0 | COMP[7:0] | Compensation value | Through the setting of compensation values, an accuracy compensation of $\pm 0.96\text{ppm}$ per second can be achieved. The compensation value is 9 bits of 2's complement with a decimal point, and the last 5 bits are the decimal part. The compensation range is $-275.5\text{ppm}$ to $+212.9\text{ppm}$ . The minimum resolution is $0.96\text{ppm}$ . | RW     |

Please refer to the following table for specific compensation accuracy:

| Compensation Value Setting |           |   |   |   |   |   |   |   |   | Compensation Value |
|----------------------------|-----------|---|---|---|---|---|---|---|---|--------------------|
| COMP-EN                    | COMP[8:0] |   |   |   |   |   |   |   |   |                    |
| 1                          | 1         | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | -275.5ppm          |

| Compensation Value Setting |           |   |   |   |   |   |   |   |   | Compensation Value |
|----------------------------|-----------|---|---|---|---|---|---|---|---|--------------------|
| COMP-EN                    | COMP[8:0] |   |   |   |   |   |   |   |   |                    |
| 1                          | 1         | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | -274.6ppm          |
|                            | ...       |   |   |   |   |   |   |   |   | ...                |
|                            | 0         | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | -30.5ppm           |
|                            | ...       |   |   |   |   |   |   |   |   | ...                |
|                            | 0         | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | -0.96ppm           |
|                            | 0         | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0ppm               |
|                            | 0         | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | +0.96ppm           |
|                            | ...       |   |   |   |   |   |   |   |   | ...                |
|                            | 0         | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | +30.5ppm           |
|                            | ...       |   |   |   |   |   |   |   |   | ...                |
|                            | 0         | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | +212.0ppm          |
|                            | 0         | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | +212.9ppm          |
| 0                          | X         | X | X | X | X | X | X | X | X | Uncompensated      |

### Compensation Calculation Instructions

When the 1Hz clock is directly output in the default state, the compensation target value is calculated by measuring the accuracy of the clock.

Assuming that the actual measured value is 0.9999888Hz, then:

Actual Vibration Frequency =  $32768 \times 0.9999888 \approx 32767.63$

Compensation Target Value =  $(\text{Actual Vibration Frequency} - \text{Target Frequency}) / \text{Target Frequency} \times 10^6$   
 $= (32767.63 - 32768) / 32768 \times 10^6$   
 $= -11.29\text{ppm}$

Based on the formula:

$$\text{COMP}[8:0] = \left( \frac{\text{Compensation Target Value [ppm]} \times 2^{15}}{10^6} \right)_{\text{Take 2's complement}} + 0b0001.00000$$

If the compensation target value is +20.3ppm, calculate the corresponding register value as follows:

$\text{COMP}[8:0] = (20.3 \times 2^{15} / 10^6)_{\text{takes 2's complement}} + 0b0001.00000$   
 $= (0.6651904)_{\text{takes 2's complement}} + 0b0001.00000$   
 $= 0b0000.10101 + 0b0001.00000$   
 $= 0b0001.10101$

If the compensation target value is -20.3ppm, calculate the corresponding register value as follows:

$\text{COMP}[8:0] = (-20.3 \times 2^{15} / 10^6)_{\text{takes 2's complement}} + 0b0001.00000$   
 $= (-0.6651904)_{\text{takes 2's complement}} + 0b0001.00000$   
 $= 0b1111.01011 + 0b0001.00000$   
 $= 0b0000.01011$

## 27.5.17 Intrusion Control Register0 (RTC\_TPCR0)

| Offset | Bit Position |        |       |         |       |   |       |   |
|--------|--------------|--------|-------|---------|-------|---|-------|---|
| 0x0040 | 7            | 6      | 5     | 4       | 3     | 2 | 1     | 0 |
| Reset  | 0xXX         |        |       |         |       |   |       |   |
| Name   | TPEN0        | TSTPE0 | TPIE0 | TPRSTE0 | TPNF0 |   | TPCT0 |   |
| Access | RW           | RW     | RW    | RW      | RW    |   | RW    |   |

| Bit | Name    | Function                          | Description                                                                                                                                                                                                                                                                                                                                                               | Access |
|-----|---------|-----------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 7   | TPEN0   | Intrusion De-<br>tection Enabled  | <ul style="list-style-type: none"> <li>0b0: The intrusion detection function is disabled on the RTC_IC0 input pin</li> <li>0b1: The intrusion detection function is enabled on the RTC_IC0 input pin</li> </ul>                                                                                                                                                           | RW     |
| 6   | TSTPE0  | Timestamp Enabled                 | <ul style="list-style-type: none"> <li>0b0: Timestamp function disabled</li> <li>0b1: Timestamp function enabled</li> </ul>                                                                                                                                                                                                                                               | RW     |
| 5   | TPIE0   | Intrusion Event Interrupt Enabled | <ul style="list-style-type: none"> <li>0b0: Intrusion event interrupt disabled</li> <li>0b1: Intrusion event interrupt enabled</li> </ul>                                                                                                                                                                                                                                 | RW     |
| 4   | TPRSTE0 | Intrusion Event Reset Enabled     | <ul style="list-style-type: none"> <li>0b0: Backup register reset disabled</li> <li>0b1: Backup register reset enabled</li> </ul>                                                                                                                                                                                                                                         | RW     |
| 3:2 | TPNF0   | Filter Function                   | <ul style="list-style-type: none"> <li>0b0X: Filter function disabled</li> <li>0b10: The RTC_IC0 input pin detection performs 3 consecutive consistency checks for filtering based on the time clock</li> <li>0b11: The RTC_IC0 input pin detection performs 3 consecutive consistency checks for filtering based on a 32-frequency division of the time clock</li> </ul> | RW     |
| 1:0 | TPCT0   | Active Edge Selection             | <ul style="list-style-type: none"> <li>0b00: RTC_IC0 input pin not detected</li> <li>0b01: RTC_IC0 input pin is enable when a rising edge is detected</li> <li>0b10: RTC_IC0 input pin is enable when a falling edge is detected</li> <li>0b11: RTC_IC0 input pin is enable when a rising or falling edge is detected</li> </ul>                                          | RW     |

## 27.5.18 Intrusion Control Register1 (RTC\_TPCR1)

| Offset | Bit Position |        |       |         |       |   |       |   |
|--------|--------------|--------|-------|---------|-------|---|-------|---|
| 0x0044 | 7            | 6      | 5     | 4       | 3     | 2 | 1     | 0 |
| Reset  | 0xXX         |        |       |         |       |   |       |   |
| Name   | TPEN1        | TSTPE1 | TPIE1 | TPRSTE1 | TPNF1 |   | TPCT1 |   |
| Access | RW           | RW     | RW    | RW      | RW    |   | RW    |   |

| Bit | Name    | Function                          | Description                                                                                                                                                                                                                                                                                                                                                               | Access |
|-----|---------|-----------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 7   | TPEN1   | Intrusion De-<br>tection Enabled  | <ul style="list-style-type: none"> <li>0b0: The intrusion detection function is disabled on the RTC_IC1 input pin</li> <li>0b1: The intrusion detection function is enabled on the RTC_IC1 input pin</li> </ul>                                                                                                                                                           | RW     |
| 6   | TSTPE1  | Timestamp enabled                 | <ul style="list-style-type: none"> <li>0b0: Timestamp function disabled</li> <li>0b1: Timestamp function enabled</li> </ul>                                                                                                                                                                                                                                               | RW     |
| 5   | TPIE1   | Intrusion Event Interrupt Enabled | <ul style="list-style-type: none"> <li>0b0: Intrusion event interrupt disabled</li> <li>0b1: Intrusion event interrupt enabled</li> </ul>                                                                                                                                                                                                                                 | RW     |
| 4   | TPRSTE1 | Intrusion Event Reset Enabled     | <ul style="list-style-type: none"> <li>0b0: Backup register reset disabled</li> <li>0b1: Backup register reset enabled</li> </ul>                                                                                                                                                                                                                                         | RW     |
| 3:2 | TPNF1   | Filter Function                   | <ul style="list-style-type: none"> <li>0b0X: Filter function disabled</li> <li>0b10: The RTC_IC1 input pin detection performs 3 consecutive consistency checks for filtering based on the time clock</li> <li>0b11: The RTC_IC1 input pin detection performs 3 consecutive consistency checks for filtering based on a 32-frequency division of the time clock</li> </ul> | RW     |
| 1:0 | TPCT1   | Active Edge Selection             | <ul style="list-style-type: none"> <li>0b00: RTC_IC1 input pin not detected</li> <li>0b01: RTC_IC1 input pin is enable when a rising edge is detected</li> <li>0b10: RTC_IC1 input pin is enable when a falling edge is detected</li> <li>0b11: RTC_IC1 input pin is enable when a rising or falling edge is detected</li> </ul>                                          | RW     |

## 27.5.19 Intrusion Status Register (RTC\_TPSR)

| Offset | Bit Position |   |   |   |   |       |      |      |
|--------|--------------|---|---|---|---|-------|------|------|
| 0x0048 | 7            | 6 | 5 | 4 | 3 | 2     | 1    | 0    |
| Reset  | 0xXX         |   |   |   |   |       |      |      |
| Name   | Reserved     |   |   |   |   | TPOVF | TPF1 | TPF0 |
| Access |              |   |   |   |   | RW    | RW   | RW   |

| Bit | Name     | Function                       | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | Access |
|-----|----------|--------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 7:3 | Reserved | Reserved                       | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | -      |
| 2   | TPOVF    | Intrusion Status Overflow Flag | Write 0 to clear the flag,write 1 has no effect<br>Read:<br><ul style="list-style-type: none"> <li>0b0: Reset state or no intrusion status overflow has occurred.</li> <li>0b1: Intrusion status overflow has occurred.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                   | RW     |
| 1   | TPF1     | Intrusion Status Flag 1        | Write 0 to clear the flag,write 1 has no effect<br>Read:<br><ul style="list-style-type: none"> <li>0b0: No intrusion event occurred on the RTC_IC1 input pin.</li> <li>0b1: A intrusion event occurred on the RTC_IC1 input pin</li> </ul> <div>  <b>Note</b><br/>           To prevent malfunction, this bit must be cleared when RTC_TPCR1.TPCT1[1:0]=0b00 and the intrusion detection edge is disabled. Please make sure that the status of this flag is 0 before setting the intrusion function to be enabled.         </div> | RW     |
| 0   | TPF0     | Intrusion Status Flag 0        | Write 0 to clear the flag,write 1 has no effect<br>Read:<br><ul style="list-style-type: none"> <li>0b0: No intrusion event occurred on the RTC_IC0 input pin.</li> <li>0b1: A intrusion event occurred on the RTC_IC0 input pin</li> </ul> <div>  <b>Note</b><br/>           To prevent malfunction, this bit must be cleared when RTC_TPCR0.TPCT0[1:0]=0b00 and the intrusion detection edge is disabled. Please make sure that the status of this flag is 0 before setting the intrusion function to be enabled.         </div> | RW     |

## 27.5.20 Second Timestamp Register (RTC\_SECTP)

| Offset | Bit Position |        |   |   |        |   |   |   |
|--------|--------------|--------|---|---|--------|---|---|---|
| 0x004C | 7            | 6      | 5 | 4 | 3      | 2 | 1 | 0 |
| Reset  | 0xXX         |        |   |   |        |   |   |   |
| Name   | Reserved     | SECTPD |   |   | SECTPU |   |   |   |
| Access |              | RO     |   |   | RO     |   |   |   |

| Bit | Name            | Function        | Description                        | Access |
|-----|-----------------|-----------------|------------------------------------|--------|
| 7   | <i>Reserved</i> | <i>Reserved</i> | -                                  | -      |
| 6:4 | SECTPD          | Second Tens     | Second timestamp tens count value  | RO     |
| 3:0 | SECTPU          | Second Units    | Second timestamp units count value | RO     |

## 27.5.21 Minute Timestamp Register (RTC\_MINTP)

| Offset | Bit Position |        |   |   |        |   |   |   |
|--------|--------------|--------|---|---|--------|---|---|---|
| 0x0050 | 7            | 6      | 5 | 4 | 3      | 2 | 1 | 0 |
| Reset  | 0xXX         |        |   |   |        |   |   |   |
| Name   | Reserved     | MINTPD |   |   | MINTPU |   |   |   |
| Access |              | RO     |   |   | RO     |   |   |   |

| Bit | Name            | Function        | Description                        | Access |
|-----|-----------------|-----------------|------------------------------------|--------|
| 7   | <i>Reserved</i> | <i>Reserved</i> | -                                  | -      |
| 6:4 | MINTPD          | Minute Tens     | Minute timestamp tens count value  | RO     |
| 3:0 | MINTPU          | Minute Units    | Minute timestamp units count value | RO     |

## 27.5.22 Hour Timestamp Register (RTC\_HOURTP)

| Offset | Bit Position |   |         |   |         |   |   |   |
|--------|--------------|---|---------|---|---------|---|---|---|
| 0x0054 | 7            | 6 | 5       | 4 | 3       | 2 | 1 | 0 |
| Reset  | 0xXX         |   |         |   |         |   |   |   |
| Name   | Reserved     |   | HOURTPD |   | HOURTPU |   |   |   |
| Access |              |   | RO      |   | RO      |   |   |   |

| Bit | Name            | Function        | Description                      | Access |
|-----|-----------------|-----------------|----------------------------------|--------|
| 7:6 | <i>Reserved</i> | <i>Reserved</i> | -                                | -      |
| 5:4 | HOURTPD         | Hour Tens       | Hour timestamp tens count value  | RO     |
| 3:0 | HOURTPU         | Hour Units      | Hour timestamp units count value | RO     |

## 27.5.23 Day Timestamp Register (RTC\_DAYTP)

| Offset | Bit Position |   |        |   |        |   |   |   |
|--------|--------------|---|--------|---|--------|---|---|---|
| 0x0058 | 7            | 6 | 5      | 4 | 3      | 2 | 1 | 0 |
| Reset  | 0xXX         |   |        |   |        |   |   |   |
| Name   | Reserved     |   | DAYTPD |   | DAYTPU |   |   |   |
| Access |              |   | RO     |   | RO     |   |   |   |

| Bit | Name            | Function        | Description                     | Access |
|-----|-----------------|-----------------|---------------------------------|--------|
| 7:6 | <i>Reserved</i> | <i>Reserved</i> | -                               | -      |
| 5:4 | DAYTPD          | Day Tens        | Day timestamp tens count value  | RO     |
| 3:0 | DAYTPU          | Day Units       | Day timestamp units count value | RO     |

27.5.24 Month Timestamp Register (RTC\_MONTP)

| Offset | Bit Position |   |   |       |   |   |   |   |
|--------|--------------|---|---|-------|---|---|---|---|
| 0x005C | 7            | 6 | 5 | 4     | 3 | 2 | 1 | 0 |
| Reset  | 0xXX         |   |   |       |   |   |   |   |
| Name   | Reserved     |   |   | MONTP |   |   |   |   |
| Access |              |   |   | RO    |   |   |   |   |

| Bit | Name     | Function | Description                 | Access |
|-----|----------|----------|-----------------------------|--------|
| 7:5 | Reserved | Reserved | -                           | -      |
| 4:0 | MONTP    | Month    | Month timestamp count value | RO     |



## 28 Watchdog Timer (WDT/SWDT)

### 28.1 Introduction

This product has two watchdog timers: one is a specialized watchdog timer (SWDT) with a dedicated internal RC (SWDTLRC: 10kHz) as the counting clock source, and the other is a universal watchdog timer (WDT) with a PCLK3 as the counting clock source. SWDT and WDT are 16-bit down-counters designed to detect software faults caused by applications deviating from normal operation due to external interference or unforeseen logical conditions.

Both watchdogs support the window function. The window interval can be preset before the count starts, and when the count value is within the window interval, the counter can be refreshed and the count restarted.

### 28.2 Features

Table 28-1 Basic Features of Watchdog Timer

|                            |                                                                                                                                                                                                                                                                                                                                                  |
|----------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Count Clock                | <ul style="list-style-type: none"> <li>● SWDT: 1/16/32/64/128/256/2048 division of SWDTLRC</li> <li>● WDT: 4/64/128/256/512/1024/2048/8192 division of PCLK3</li> </ul>                                                                                                                                                                          |
| Maximum Overflow Time      | <ul style="list-style-type: none"> <li>● SWDT: 3.72 hours (SWDTLRC=10kHz)</li> <li>● WDT: 10.7 seconds (PCLK3=50MHz)</li> </ul>                                                                                                                                                                                                                  |
| Counting Mode              | Decrement count                                                                                                                                                                                                                                                                                                                                  |
| Window Function            | Can set window interval, define the allowable interval of refresh action                                                                                                                                                                                                                                                                         |
| Startup Mode               | <ol style="list-style-type: none"> <li>1. Hardware startup</li> <li>2. Software startup</li> </ol>                                                                                                                                                                                                                                               |
| Stop Condition             | <ol style="list-style-type: none"> <li>1. Resetting</li> <li>2. Underflow, Refresh error occurs and then restart: <ul style="list-style-type: none"> <li>● In hardware startup mode, it starts automatically after a reset or interrupt request is output</li> <li>● In software startup mode, set refresh register again</li> </ul> </li> </ol> |
| Interrupt/Reset Conditions | <ol style="list-style-type: none"> <li>1. Count underflow</li> <li>2. Refresh error</li> </ol>                                                                                                                                                                                                                                                   |

### 28.3 Functional Description

#### 28.3.1 Start the Watchdog

Watchdog counter can be started in two ways: Hardware startup mode and software startup mode.

The hardware startup mode refers to the setting of the watchdog counter (ICG0 register) read from the main flash memory area during startup, and the counter automatically starts counting. The software startup mode refers to the process where, after setting the control register, the refresh register is written to complete the refresh action, and the counter starts counting.

### 28.3.2 Hardware Startup Mode

When the ICG0.WDTAUTS/SWDTAUTS is set to 0, it is the hardware startup mode. WDT/SWDT\_CR settings are ignored in hardware startup mode.

In hardware start mode, the WDT/SWDT-related settings (counting clock, window setting value, counting period, etc.) in the ICG0 register are loaded into the WDT/SWDT module during the reset period. After the reset, the counter automatically starts counting according to the settings. The figure below shows a hardware start mode example.

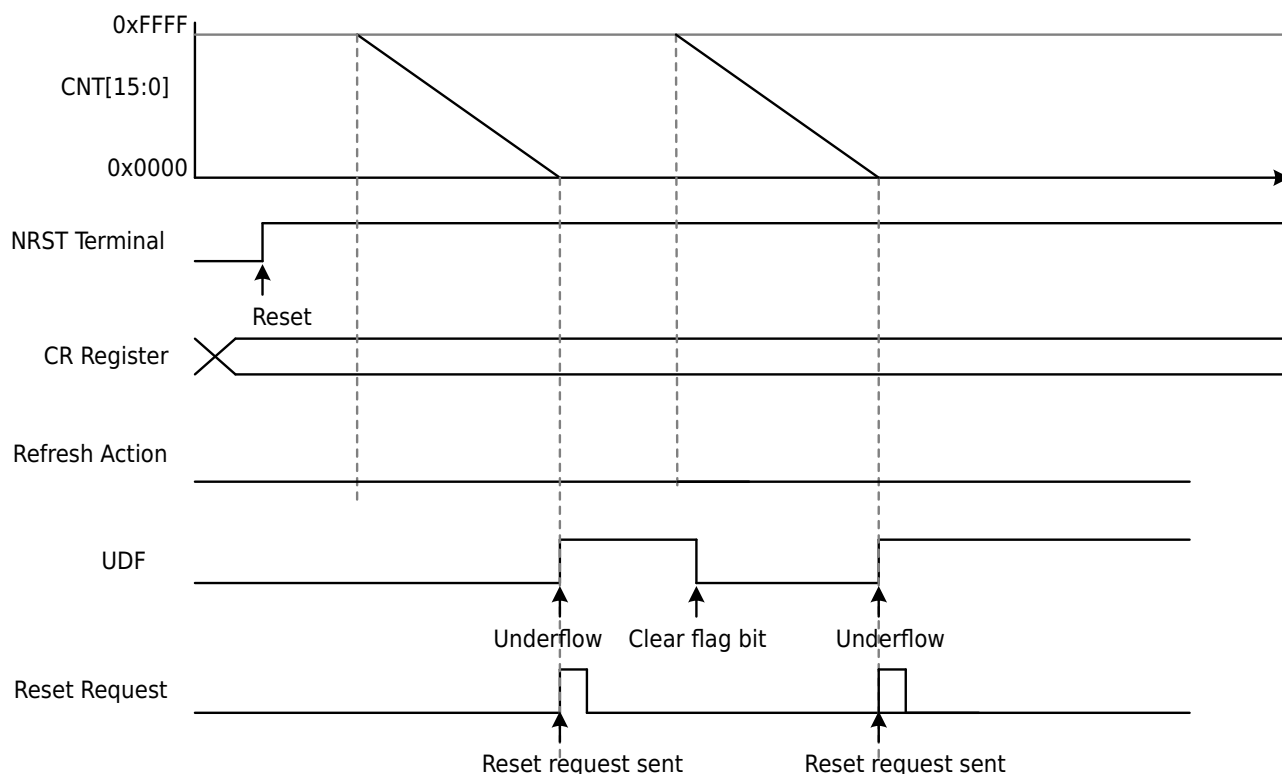


Figure 28-1 Hardware Startup Mode Example

### 28.3.3 Software Startup Mode

When the ICG0.WDTAUTS/SWDTAUTS is set to 1, WDT/SWDT is started via refresh register (software start mode). After reset, set the counting clock, window setting value, counting period, etc. in the WDT/SWDT\_CR register, and then perform the refresh action, and the counter starts to count. WDT/SWDT\_CR can be configured only once; subsequent writes are inactive. The figure below shows a software start mode example.

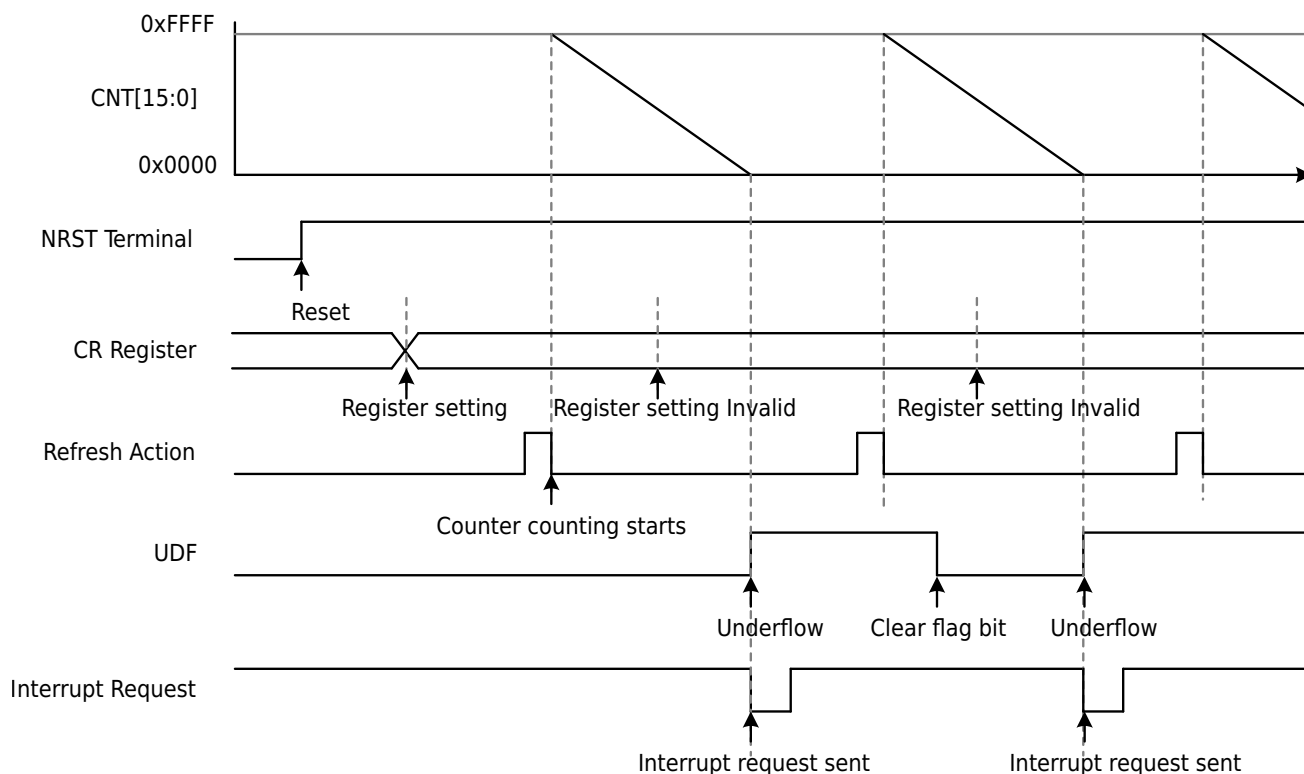


Figure 28-2 Software Startup Mode Example

### 28.3.4 Refresh Action

A refresh sequence (write 0x0123 followed by 0x3210 to WDT/SWDT\_RR) starts or restarts the counter (software start mode).

Writing 0x0123 and 0x3210 to WDT/SWDT\_RR register is unaffected by other register accesses or reads during the refresh sequence.

The figure below shows an operation example.

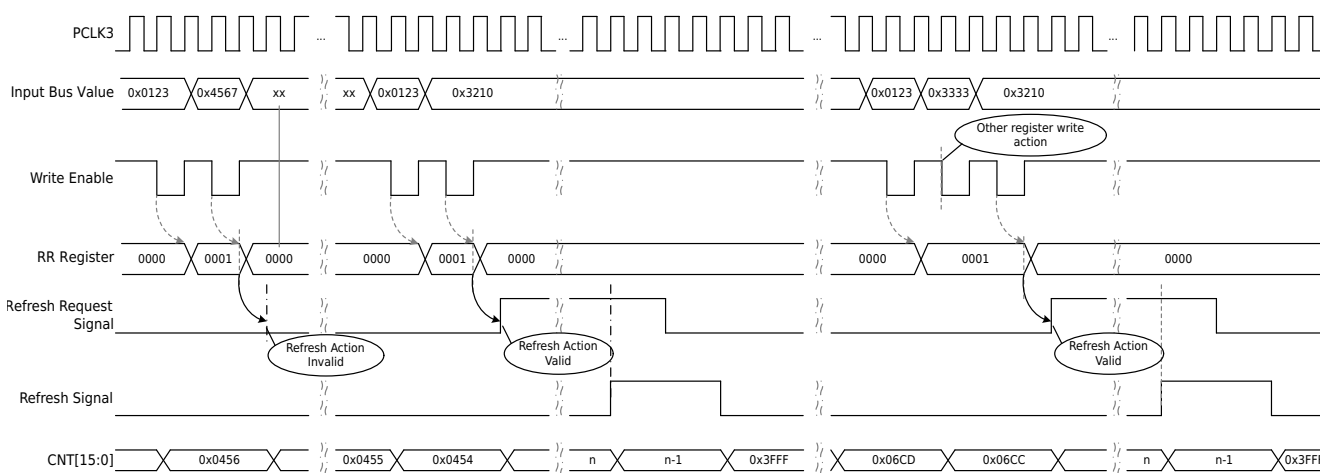


Figure 28-3 Examples of Various Refresh Action Sequences (action confirmation, refresh required signal falling edge, etc.)

Refresh operations require 4 clock periods to update the counting. Therefore, complete the writing of the refresh register 4 count values in advance of refreshing the lower window and the overflow position. Check the status register for the count value.

### 28.3.5 Flag Bit

Refresh error flag bit and underflow flag bit remain set if an interrupt is requested. Upon entering an interrupt, check the flag bit to identify the cause.

Clear the flag bit: Read "1" first, then write "0".

**Note**

Within a certain period of time when a refresh error or underflow error occurs, the clear operation of the flag bit has no effect. This period of time is: 1 counting period +2 SWDTLRC (SWDT module); 1 counting period +2 PCLK3 (WDT module).

When refreshing error flag bit and underflow flag bit, the counting of the hardware watchdog does not stop; while counting stops in software start mode. When writing "0" to the flag bit, the SWDT needs at most 3 SWDTLRC and 2 PCLK3 times before the register bit can be clear to 0. It takes at most five PCLK3 times for the register bit of WDT to be clear to 0. After clearing the flag bit, it is necessary to re-read the flag bit to confirm whether the clearing operation is successful.

Flag bit clearing procedures:

- Step1.** Read WDT/SWDT\_SR to check the flag bit.
- Step2.** Write "0" to the flag bit set to "1".
- Step3.** Read WDT/SWDT\_SR to confirm if the flag bit is cleared. Repeat Step 2 if the flag bit remain 1.

### 28.3.6 Interrupt/Reset

WDT/SWDT can trigger interrupt or reset requests upon counter underflow or refresh error. In hardware startup mode, generate an interrupt or a reset request depending on ICG0.WDTITS/SWDTITS bit. In software startup mode, set the WDT/SWDT\_CR.ITS bit to select interrupt or reset request.

WDT/SWDT generates interrupt/reset requests under 2 conditions. One is that counter counts produce underflow; One is to perform a refresh action outside the refresh allowed interval, resulting in a refresh error.

### 28.3.7 Count Underflow

As shown in the following figure, underflow occurs when the counter decrements to 0.

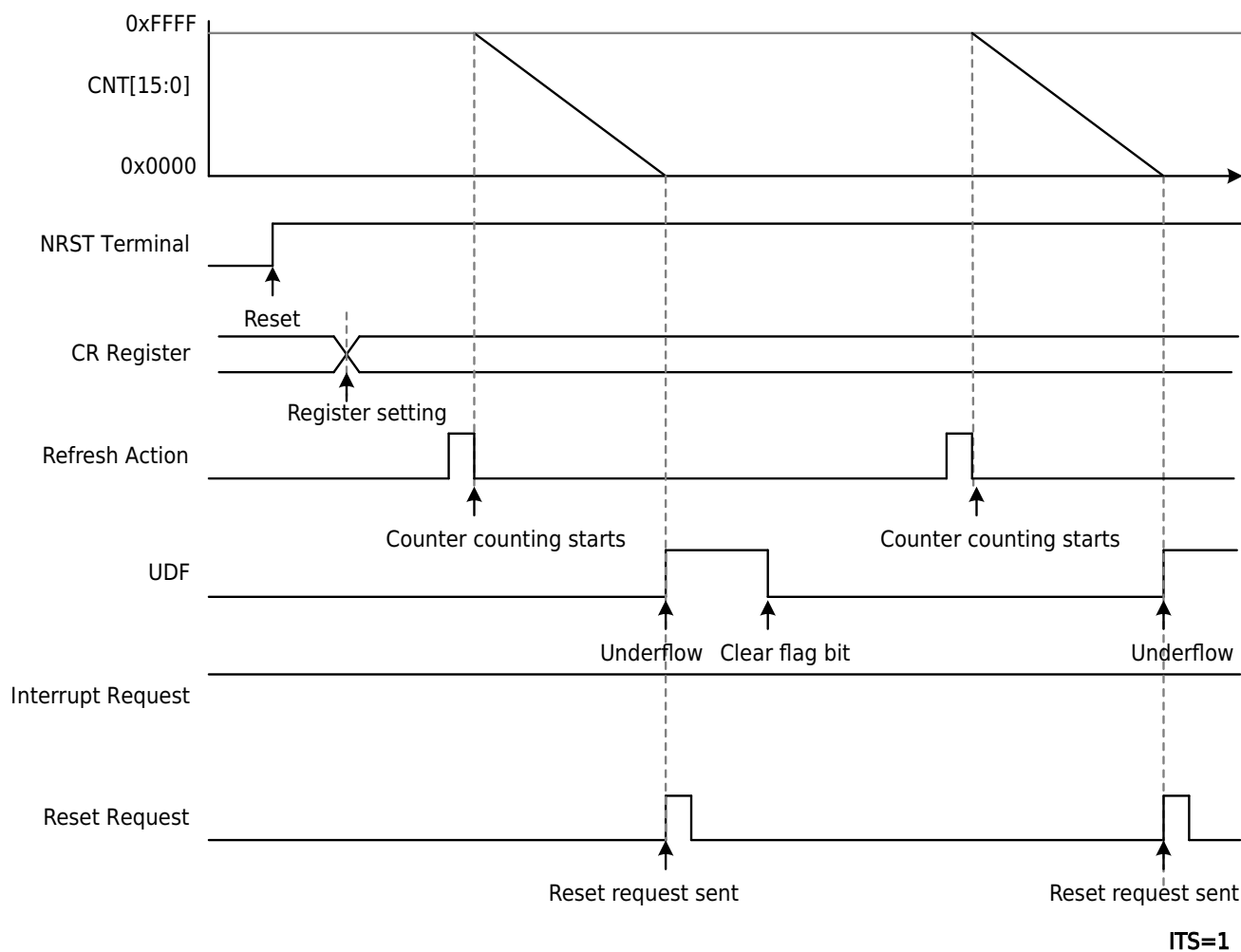


Figure 28-4 Counter Underflow Operation Example

### 28.3.8 Refresh Error

After setting the window range, the counter is refreshed and re-counting only when the refresh action is performed within the window range. A refresh error occurs when the refresh action is performed outside the window range. The figure below shows a refresh operation example.

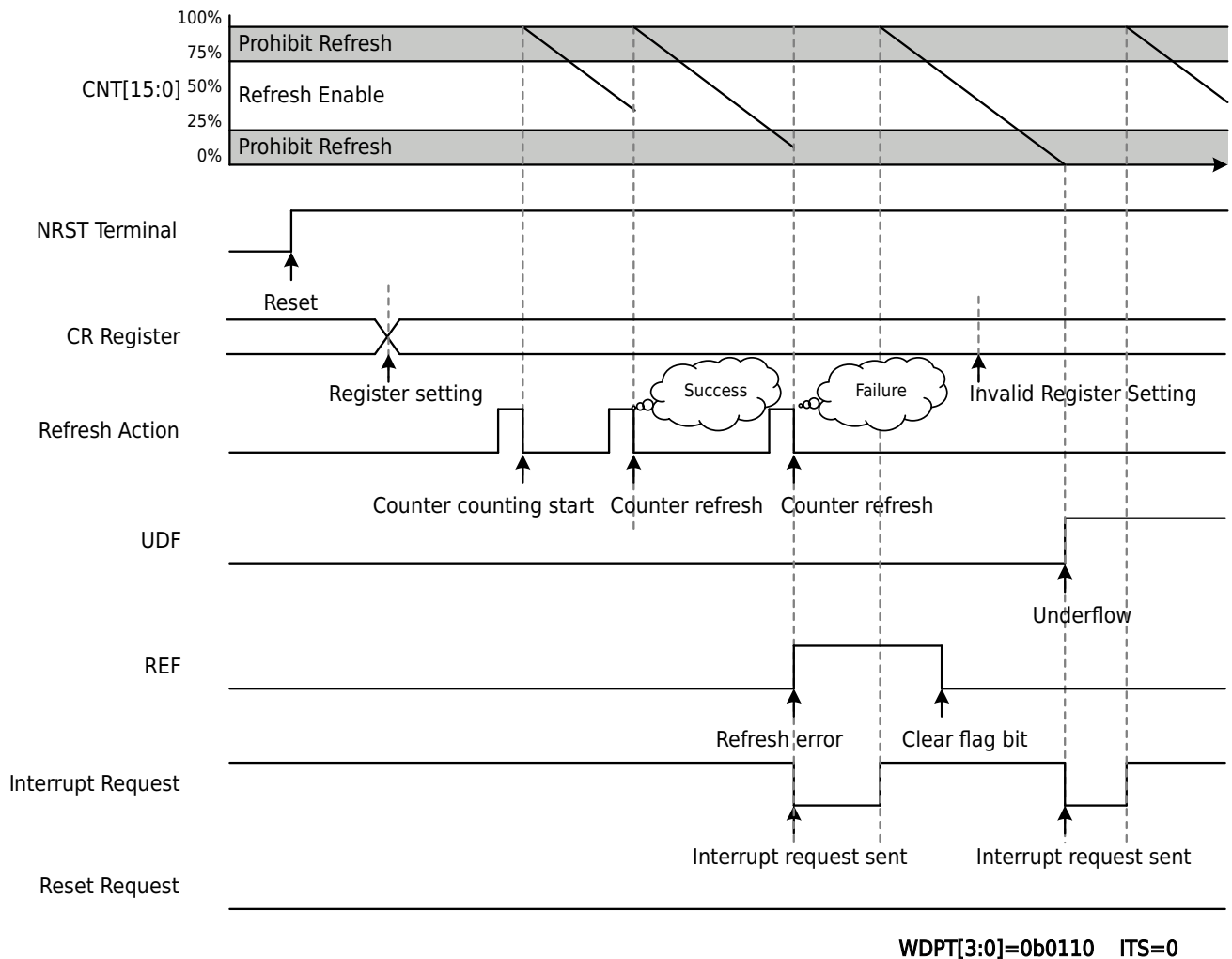


Figure 28-5 Counter Refresh Operation Example

## 28.4 Register Description

### 28.4.1 Register List

WDT base address: 0x40049000

Table 28-2 Watchdog Timer (WDT) Register Offset

| Offset | Register | Description          | Width |
|--------|----------|----------------------|-------|
| 0x00   | WDT_CR   | WDT Control Register | 32    |
| 0x04   | WDT_SR   | WDT Status Register  | 32    |
| 0x08   | WDT_RR   | WDT Refresh Register | 32    |

SWDT base address: 0x40049400

Table 28-3 Watchdog Timer (SWDT) Register Offset

| Offset | Register | Description           | Width |
|--------|----------|-----------------------|-------|
| 0x00   | SWDT_CR  | SWDT Control Register | 32    |
| 0x04   | SWDT_SR  | SWDT Status Register  | 32    |
| 0x08   | SWDT_RR  | SWDT Refresh Register | 32    |

## 28.4.2 Control Register (SWDT\_CR, WDT\_CR)

| Offset | Bit Position |          |    |    |    |    |    |    |    |    |    |    |    |    |    |        |          |    |    |    |    |      |    |   |   |     |    |   |   |          |   |      |
|--------|--------------|----------|----|----|----|----|----|----|----|----|----|----|----|----|----|--------|----------|----|----|----|----|------|----|---|---|-----|----|---|---|----------|---|------|
| 0x00   | 31           | 30       | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16     | 15       | 14 | 13 | 12 | 11 | 10   | 9  | 8 | 7 | 6   | 5  | 4 | 3 | 2        | 1 | 0    |
| Reset  | 0x80010FF3   |          |    |    |    |    |    |    |    |    |    |    |    |    |    |        |          |    |    |    |    |      |    |   |   |     |    |   |   |          |   |      |
| Name   | ITS          | Reserved |    |    |    |    |    |    |    |    |    |    |    |    |    | SLPOFF | Reserved |    |    |    |    | WDPT |    |   |   | CKS |    |   |   | Reserved |   | PERI |
| Access |              | RW       |    |    |    |    |    |    |    |    |    |    |    |    |    |        | RW       |    |    |    |    |      | RW |   |   |     | RW |   |   |          |   |      |

| Bit   | Name     | Function                                               | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | Access |
|-------|----------|--------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31    | ITS      | Refresh error/overflow interrupt/reset selection       | <ul style="list-style-type: none"> <li>0b0: Interrupt request</li> <li>0b1: Reset request</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                  | RW     |
| 30:17 | Reserved | Reserved                                               | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | -      |
| 16    | SLPOFF   | WDT/SWDT counting is disabled in low-power modes       | WDT_CR: <ul style="list-style-type: none"> <li>0b0: Enable WDT counting in Sleep mode</li> <li>0b1: Disable WDT counting in Sleep mode</li> </ul> SWDT_CR: <ul style="list-style-type: none"> <li>0b0: Enable SWDT counting in Sleep/Stop mode</li> <li>0b1: Disable SWDT counting in Sleep/Stop mode</li> </ul>                                                                                                                                                                                      | RW     |
| 15:12 | Reserved | Reserved                                               | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | -      |
| 11:8  | WDPT     | Refresh the percentage of the allowed area count value | <ul style="list-style-type: none"> <li>0b0000: 0%-100%</li> <li>0b0001: 0%-25%</li> <li>0b0010: 25%-50%</li> <li>0b0011: 0%-50%</li> <li>0b0100: 50%-75%</li> <li>0b0101: 0%-25%, 50%-75%</li> <li>0b0110: 25%-75%</li> <li>0b0111: 0%-75%</li> <li>0b1000: 75%-100%</li> <li>0b1001: 0%-25%, 75%-100%</li> <li>0b1010: 25%-50%, 75%-100%</li> <li>0b1011: 0%-50%, 75%-100%</li> <li>0b1100: 50%-100%</li> <li>0b1101: 0%-25%, 50%-100%</li> <li>0b1110: 25%-100%</li> <li>0b1111: 0%-100%</li> </ul> | RW     |
| 7:4   | CKS      | Count clock                                            | WDT_CR: <ul style="list-style-type: none"> <li>0b0010: PCLK3/4</li> <li>0b0110: PCLK3/64</li> <li>0b0111: PCLK3/128</li> </ul>                                                                                                                                                                                                                                                                                                                                                                        | RW     |



| Bit | Name     | Function        | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | Access |
|-----|----------|-----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 7:4 | CKS      | Count clock     | <ul style="list-style-type: none"> <li>● 0b1000: PCLK3/256</li> <li>● 0b1001: PCLK3/512</li> <li>● 0b1010: PCLK3/1024</li> <li>● 0b1011: PCLK3/2048</li> <li>● 0b1101: PCLK3/8192</li> <li>● Others: Setting prohibited</li> </ul> SWDT_CR: <ul style="list-style-type: none"> <li>● 0b0000: SWDTCLK</li> <li>● 0b0100: SWDTCLK/16</li> <li>● 0b0101: SWDTCLK/32</li> <li>● 0b0110: SWDTCLK/64</li> <li>● 0b0111: SWDTCLK/128</li> <li>● 0b1000: SWDTCLK/256</li> <li>● 0b1011: SWDTCLK/2048</li> <li>● Others: Setting prohibited</li> </ul> | RW     |
| 3:2 | Reserved | Reserved        | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | -      |
| 1:0 | PERI     | Counting period | <ul style="list-style-type: none"> <li>● 0b00: 256 cycle</li> <li>● 0b01: 4096 cycle</li> <li>● 0b10: 16384 cycle</li> <li>● 0b11: 65536 cycle</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                     | RW     |

### 28.4.3 Status Register (SWDT\_SR, WDT\_SR)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |     |     |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-----|-----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0x04   | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16  | 15  | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |     |     |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |     |     |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|        |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    | REF | UDF |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RW  | RW  | RO |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|        |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |     |     |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit   | Name     | Function                | Description                                                                                                                                                                                             | Access |
|-------|----------|-------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:18 | Reserved | Reserved                | -                                                                                                                                                                                                       | -      |
| 17    | REF      | Refresh error flag      | <ul style="list-style-type: none"> <li>● 0b0: No refresh error</li> <li>● 0b1: Refresh error occurred</li> </ul> After reading a value of 1 from this bit, writing 0 to it will clear this bit.         | RW     |
| 16    | UDF      | Counting underflow mark | <ul style="list-style-type: none"> <li>● 0b0: No counter underflow</li> <li>● 0b1: Counter underflow occurred</li> </ul> After reading a value of 1 from this bit, writing 0 to it will clear this bit. | RW     |
| 15:0  | CNT      | Count value             | Counter current count                                                                                                                                                                                   | RO     |

### 28.4.4 Refresh Register (SWDT\_RR, WDT\_RR)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0x08   | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RF |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit   | Name     | Function      | Description                                                                                                                                          | Access |
|-------|----------|---------------|------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:16 | Reserved | Reserved      | -                                                                                                                                                    | -      |
| 15:0  | RF       | Refresh value | Write 0x0123 and 0x3210 to complete the refresh. After writing 0x0123, the register reads as 0x00000001; in all other cases, it reads as 0x00000000. | RW     |

## 28.5 Precautions

During SWDT operation, the frequency of the peripheral clock PCLK3 must equal to or greater than 4 timers of counting clock frequency. That is, PCLK3 frequency  $\geq$  counting clock frequency  $\times$  4.

## 29 Universal Synchronous Asynchronous Receiver/Transmitter (USART)

### 29.1 Introduction

This product is equipped with 10 units of Universal Synchronous Asynchronous Receiver/Transmitter (USART) modules, which can flexibly exchange full-duplex data with external devices; the USART equipped in this product supports universal asynchronous serial communication interface (UART), clock synchronous communication interface, smart card interface (ISO/IEC7816-3) and LIN communication interface. Support modem operation (CTS/RTS operation), multiprocessor operation, DE function. UART receive timeout function is supported by cooperating with the Timer0 module. USART1 supports the STOP mode wake-up function via the RX line.

### 29.2 Features

| Function                            | Support Module |        |        |        |        |        |        |        |        |         |
|-------------------------------------|----------------|--------|--------|--------|--------|--------|--------|--------|--------|---------|
|                                     | USART1         | USART2 | USART3 | USART4 | USART5 | USART6 | USART7 | USART8 | USART9 | USART10 |
| UART                                | ✓              | ✓      | ✓      | ✓      | ✓      | ✓      | ✓      | ✓      | ✓      | ✓       |
| Multiprocessor Communication        | ✓              | ✓      | ✓      | ✓      | ✓      | ✓      | ✓      | ✓      | ✓      | ✓       |
| Clock Synchronization Communication | ✓              | ✓      | ✓      | ✓      | ✓      | ✓      | ✓      | ✓      | ✓      | ✓       |
| Wake-up from STOP Mode via RX Line  | ✓              | -      | -      | -      | -      | -      | -      | -      | -      | -       |
| Fractional Baud Rate                | ✓              | ✓      | ✓      | ✓      | ✓      | ✓      | ✓      | ✓      | ✓      | ✓       |
| LIN                                 | -              | -      | -      | -      | ✓      | -      | -      | -      | -      | ✓       |
| Smart Card                          | ✓              | ✓      | ✓      | ✓      | -      | ✓      | ✓      | ✓      | ✓      | -       |
| UART Receive Timeout Function       | ✓              | ✓      | ✓      | ✓      | ✓      | ✓      | ✓      | ✓      | ✓      | ✓       |
| RS485 Driver Enable                 | ✓              | ✓      | ✓      | ✓      | ✓      | ✓      | ✓      | ✓      | ✓      | ✓       |

#### USART Features

- Support full-duplex asynchronous communication and full-duplex clock synchronous communication
- Support LIN bus
- Support smart card interface (ISO/IEC7816-3)
- Transmitter and receiver have independent enable bits
- Built-in double buffer
- LSB/MSB optional
- Support modem operation (CTS/RTS)
- Transmission flag: transmit data register empty, transmit data complete, receive data register full, receive error flag, UART receive timeout flag, LIN wake-up signal detection flag, LIN interval segment detection flag, LIN bus error flag
- Support RS485 Driver Enable function

#### UART Features

- Data length programmable: 8 bits/9 bits
- Parity function configurable: odd parity/even parity/no parity
- Stop bit configurable: 1 bit/2 bits

- Clock source optional: internal clock source (clock generated by internal baud rate generator) /external clock source (clock input from USARTn\_CK pin)
- Receive error: parity error, frame error, overflow error
- Support for inter-processor communications
- Built-in digital filter
- Supports receiving data TIMEOUT function
- Unit 1 supports Stop mode wake-up function
- Supports full-duplex/half-duplex communication mode

#### Clock Synchronization Mode Features

- Data length: 8 bits
- Receive error: overflow error
- Clock source: internal clock source (clock generated by internal baud rate generator) /external clock source (clock input by USARTn\_CK pin)
- Supports full-duplex communication mode

#### Smart Card Interface Features

- Data length: 8 bits
- When a parity error is detected, it can automatically send out an error signal
- Support data retransmission

#### LIN Features

- Data length: 8 bits
- Supports wake-up signal detection
- Supports 10/11-bit synchronization interval segment (BF: Break Field) detection
- Supports synchronization segment measurement, register records measurement value
- Supports 10/11/13/14-bit interval segment (BF) transmission
- Supports bus conflict detection
- Supports loop mode

## 29.3 Functional Description

### 29.3.1 System Block Diagram

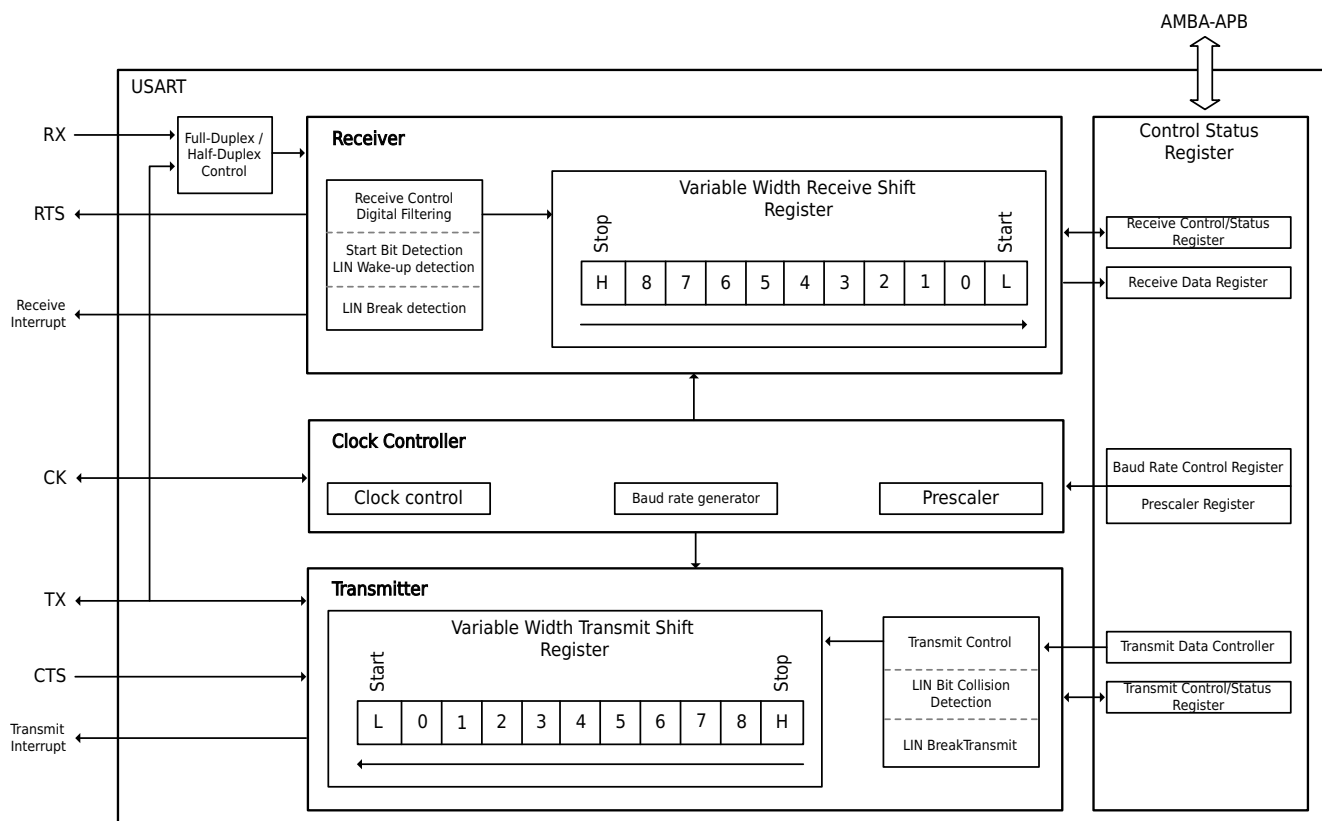


Figure 29-1 USART System Block Diagram

Table 29-2 USART pin description

| Pin Name      | Direction    | Description                                                            |
|---------------|--------------|------------------------------------------------------------------------|
| USARTn_CK     | Input/Output | Clock                                                                  |
| USARTn_TX     | Input/Output | Transmit data pin<br>Also used as receive data pin in half-duplex mode |
| USARTn_RX     | Input        | Receive data pin                                                       |
| USARTn_CTS    | Input        | Modem operation pin<br>Clear send pin                                  |
| USARTn_RTS/DE | Output       | Modem operation pin<br>Request send pin/Driver Enable pin              |



#### Note

The value of n is 1-10.

## 29.3.2 UART Function

### 29.3.2.1 Clock

The UART can select either the clock generated by the internal baud rate generator (internal clock source) or the clock input from the USARTn\_CK pin (external clock source) as the communication clock source.

#### Internal Clock Source

When USARTn\_CR2.CLKC bit is set to 0b00 or 0b01, the clock source is selected as the internal clock source, i.e., the clock generated by the internal baud rate generator.

- When USARTn\_CR2.CLKC=0b00, the USARTn\_CK pin is not used as a clock pin and can function as a general-purpose IO.
- When USARTn\_CR2.CLKC=0b01, the USARTn\_CK pin outputs a clock with the same frequency as the communication baud rate.

When USARTn\_PR.ULBREN is 0, the clock source of the internal baud rate generator is selected by the USARTn\_PR.PSC bit setting as PCLK1, PCLK1/4, PCLK1/16, PCLK1/64. When USARTn\_PR.ULBREN is 1, the ultra-low baud rate function is enabled, and the clock source of the internal baud rate generator is selected by the USARTn\_PR.PSC bit setting as PCLK1/128, PCLK1/256, PCLK1/512, PCLK1/1024.

#### External Clock Source

When USARTn\_CR2.CLKC bit is set to 0b10 or 0b11, the clock source is selected as the external clock input from the USARTn\_CK pin. The input clock frequency is 16 times the Baud rate (when USARTn\_CR1.OVER8=0) or 8 times the Baud rate (when USARTn\_CR1.OVER8=1).

#### Maximum Baud Rate

When the internal clock source is selected, the baud rate generated by the internal baud rate generator is calculated using the following formula:

$$B = \frac{C}{8 \times (2 - \text{OVER8}) \times (\text{DIV\_Integer} + 1)}$$

Where:

- B: Baud rate, unit: Mbps
- C: When USARTn\_PR.ULBREN is 0, the clock set by USARTn\_PR.PSC bit (PCLK1, PCLK1/4, PCLK1/16, PCLK1/64), unit: MHz  
When USARTn\_PR.ULBREN is 1, the clock set by USARTn\_PR.PSC bit (PCLK1/128, PCLK1/256, PCLK1/512, PCLK1/1024), unit: MHz
- OVER8: USARTn\_CR1.OVER8 setting value
- DIV\_Integer: USARTn\_BRR.DIV\_Integer setting value

Maximum Baud Rate is PCLK1/8 (Mbps).

When using an External Clock Source, the maximum frequency requirement for the external input UART Clock is PCLK1/4 (MHz). Therefore, when the Clock Source is an external input clock:

- If USARTn\_CR1.OVER8=0, the Maximum Baud Rate is PCLK1/64 (Mbps).
- If USARTn\_CR1.OVER8=1, the Maximum Baud Rate is PCLK1/32 (Mbps).

Please note that in addition to the PCLK1-based calculation method described above for the maximum UART baud rate, you must also refer to the maximum communication baud rate specified in the "Electrical Specifications" chapter in the Datasheet.

### 29.3.2.2 Data Format

A frame of data in UART mode consists of start bit, data bit, check bit, and stop bit.

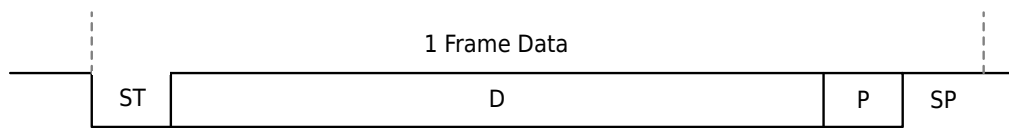


Figure 29-2 UART Data Format

- **Start Bit (ST)** : The start bit is always composed of one bit at Low level.
- **Data Bit (D)** : The data bit can be configured as 8-bit or 9-bit.
- **Check Bit (P)** : The check bit can be configured as 1-bit even parity, 1-bit odd parity, or no check bit.
- **Stop Bit (SP)** : The stop bit is fixed at high level and can be configured as 1 bit or 2 bits.

### 29.3.2.3 Modem Operation

Modem operations include the CTS function and RTS function. When USARTn\_CR3.RTSE=1, the RTS Function is Enabled. When USARTn\_CR3.CTSE=1, the CTS Function is Enabled.

#### CTS Function

The CTS function controls data transmission via the input of the USARTn\_CTS pin. Data can only be transmitted when the USARTn\_CTS pin input is at low level. If the USARTn\_CTS input transitions to high level during transmission, the ongoing data transmission is unaffected.

#### RTS Function

The RTS function outputs a low level through the USARTn\_RTS pin to request the peer to transmit data.

The USARTn\_RTS pin outputs a low level when all of the following conditions are met:

- Receive Enable (USARTn\_CR1.RE=1) and not currently receiving data.
- No unread receive data is in the USARTn\_RDR.RDR Register.
- No reception errors occurred, including frame errors, parity errors, and overflow errors.

### 29.3.2.4 Transmitter

The transmitter can send 8-bit or 9-bit data, depending on the setting of the USARTn\_CR1.M bit.

When USARTn\_CR1.TE bit is set to 1, the data is serially output on the TX pin after writing transmit data. The corresponding clock pulses can be optionally output or not output on the USARTn\_CK pin.

The sequence for transmitting data is: Start Bit -> Data Bit (MSB/LSB) -> Check Bit (if present) -> Stop Bit.

The transmit data register (TDR) and the internal transmit shift register form a double-buffered structure, enabling continuous data transmission.

When transmitting data via the transmit data register empty interrupt or DMA, to ensure transmission accuracy, only one data write can be performed per request.

#### Transmit Data Setting Procedure

- Step1.** Set the USARTn\_CR1 register to its reset value;
- Step2.** Configure the pins required for UART operation;
- Step3.** Select the clock source via the USARTn\_CR2.CLKC bit;
- Step4.** Configure the USARTn\_CR1, USARTn\_CR2, and USARTn\_CR3 Registers;

- Step5.** Set USARTn\_PR to select the prescaler value, and configure the USARTn\_BRR register for communication baud rate (no setting required when the clock source is an external clock source);
- Step6.** Enable the transmitter (USARTn\_CR1.TE=1). If the transmit data register empty interrupt is required, set USARTn\_CR1.TXEIE=1.
- Step7.** Wait for the transmit data register to be empty, write communication data to USARTn\_TDR.TDR, data is transferred to the transmit shift register, transmission starts;

**Note**

When the CTS function is enabled, data is transferred to the transmit shift register and transmission starts when the USARTn\_CTS input is at Low level.

- Step8.** To continuously transmit data, repeat7;
- Step9.** Check whether the transmission is complete by verifying the USARTn\_SR.TC bit. For continuous data transmission using transmit interrupts, the last data can be written via the TI interrupt by setting USARTn\_CR1.TXEIE to 0 and USARTn\_CR1.TCIE to 1. A transmit completion interrupt will be generated after the last frame transmission ends.

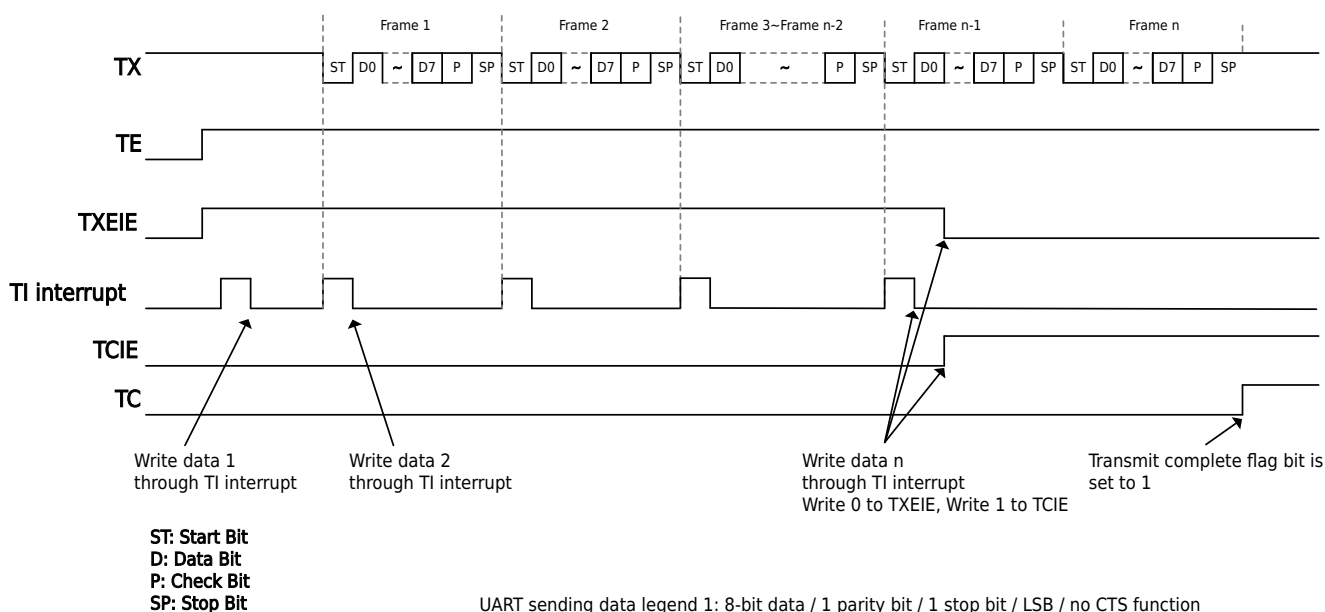


Figure 29-3 UART Transmit Data Diagram 1

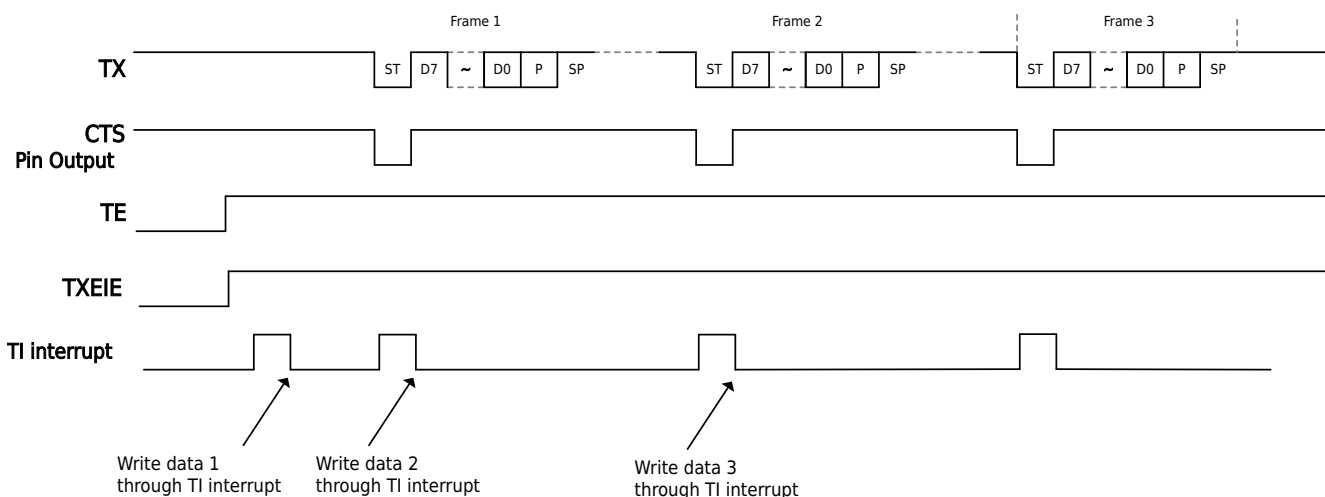


Figure 29-4 UART transmit Data Diagram 2



## Transmitter Interrupt

In UART Mode, the transmitter supports two interrupts: transmit data register empty interrupt (TI) and transmit completion interrupt (TCI).

- TXEIE=1, when the value of USARTn\_TDR.TDR register is transferred to the transmit shift register, a TI Interrupt is generated.
- TCIE=1, when the last bit of transmitted data is sent and the USARTn\_TDR.TDR register is not updated, a TCI interrupt is generated.

### 29.3.2.5 Receiver

The receiver can receive 8-bit or 9-bit data, depending on the setting of the USARTn\_CR1.M bit. After the receiver enable bit (USARTn\_CR.RE) is set to 1 and the start bit is detected, the data on the RX pin is received into the receive shift register. Once a complete frame of data is received, the data is transferred from the receive shift register to the receive data register USARTn\_RDR.RDR.

The sequence for receive data is: start bit -> data bit (MSB/LSB) -> check bit (present or absent) -> stop bit.

The receive data register USARTn\_RDR.RDR and the internal receive shift register form a double-buffered structure, enabling continuous data reception.

When receiving data through the receive data register full interrupt or DMA, only one data read operation can be performed per request.

## Start Bit Detection

Start bit detection can be configured for either low level or falling edge detection, determined by the USARTn\_CR1.SBS bit. When USARTn\_CR1.SBS=0, low level detection is selected; when USARTn\_CR1.SBS=1, falling edge detection is selected.

## Sampling and Reception Tolerance

Upon detecting a start condition (low level or falling edge), the USART synchronizes the receive data with the internal basic clock to initiate data reception.

Data sampling occurs at the center of the data. When USARTn\_CR1.OVER8=0, sampling is performed on the 8th internal basic clock; when USARTn\_MR.OVER8=1, sampling is performed on the 4th internal basic clock.

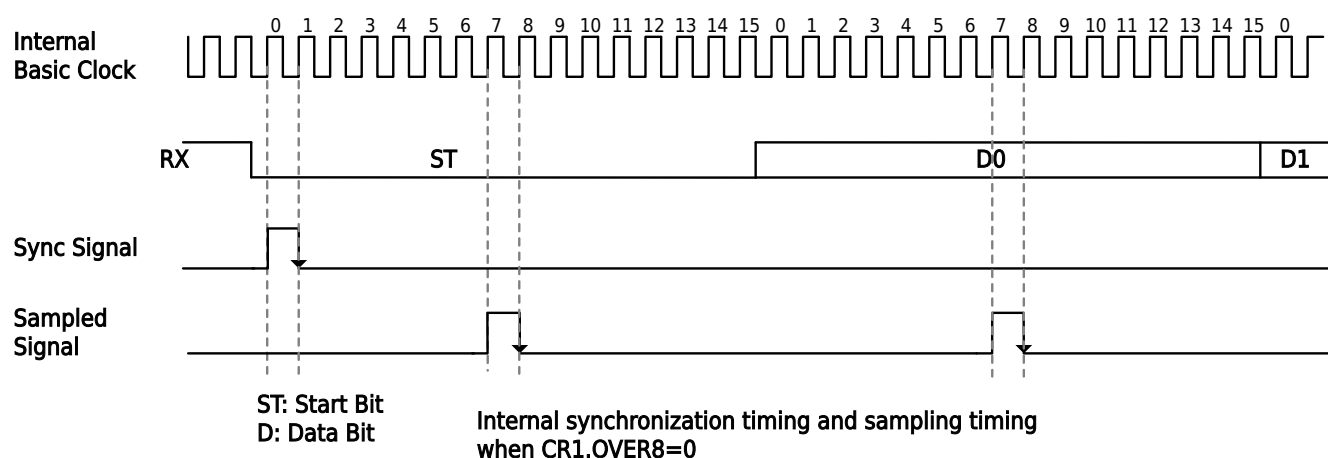


Figure 29-5 USART Internal Synchronization and Sample Timing

The UART asynchronous receiver can only operate normally when the total clock system deviation is less than the tolerance of the UART receiver. Factors affecting the total deviation include:

- Deviation caused by transmitter error (including the deviation of the transmitter local oscillator)
- Error caused by baud rate quantization of the receiver
- Deviation of the receiver local oscillator
- Deviation caused by transmission line

For correct receive data, the maximum deviation value allowed by the USART asynchronous receiver depends on the following options:

- Data length FL (Frame Length). FL is determined by the 8 or 9 data bits defined by the USART\_CR1.M bit and the parity enable bit defined by the PCE bit.
- The oversampling ratio (8x or 16x) is defined by the USART\_CR1.OVER8 bit.
- The USART\_CR1.FBME bit defines whether to use fractional baud rate.

Table 29-3 Tolerance of USART Receiver

| FL | DIV_Fraction is 0 |             | DIV_Fraction is not 0 |             |
|----|-------------------|-------------|-----------------------|-------------|
|    | OVER8 Bit=0       | OVER8 bit=1 | OVER8 bit=0           | OVER8 bit=1 |
| 10 | 4.375%            | 3.75%       | 3.88%                 | 3%          |
| 11 | 3.97%             | 3.41%       | 3.53%                 | 2.73%       |
| 12 | 3.646%            | 3.125%      | 3.23%                 | 2.5%        |



#### Note

Under special circumstances, when the RX pin receives continuous high level signals for the duration of FL, the data specified in the above table may vary slightly.

### Receive Data Setting Procedure

- Step1.** Set the USARTn\_CR1 register to its reset value;
- Step2.** Configure the pins required for UART operation;
- Step3.** Select the clock source via the USARTn\_CR2.CLKC bit;
- Step4.** Configure the USARTn\_CR1, USARTn\_CR2, and USARTn\_CR3 registers;
- Step5.** Set USARTn\_PR to select the prescaler value, and configure the USARTn\_BRR register for communication baud rate (no setting required when the clock source is an external clock source);
- Step6.** Enable the receiver (USARTn\_CR1.RE=1). If receive interrupt is required, set USARTn\_CR1.RIE=1.
- Step7.** Upon detecting the start bit, the receiver loads the data into the receive shift register and verifies the check bit and stop bit.
  - When a parity error occurs, the received data is transferred to the USARTn\_RDR.RDR register and the USARTn\_SR.PE flag is set.
  - When the stop bit is not at a high level, a frame error occurs. The received data is transferred to the USARTn\_RDR.RDR register, and the USARTn\_SR.FE flag is set.
  - When an overflow error occurs, data is lost and the USARTn\_SR.ORE flag is set.
  - When no error occurs, the received data is transferred to the USARTn\_RDR.RDR register, and the USARTn\_SR.RXNE flag is set. By repeating 7 after reading the received data, continuous data reception can be achieved.

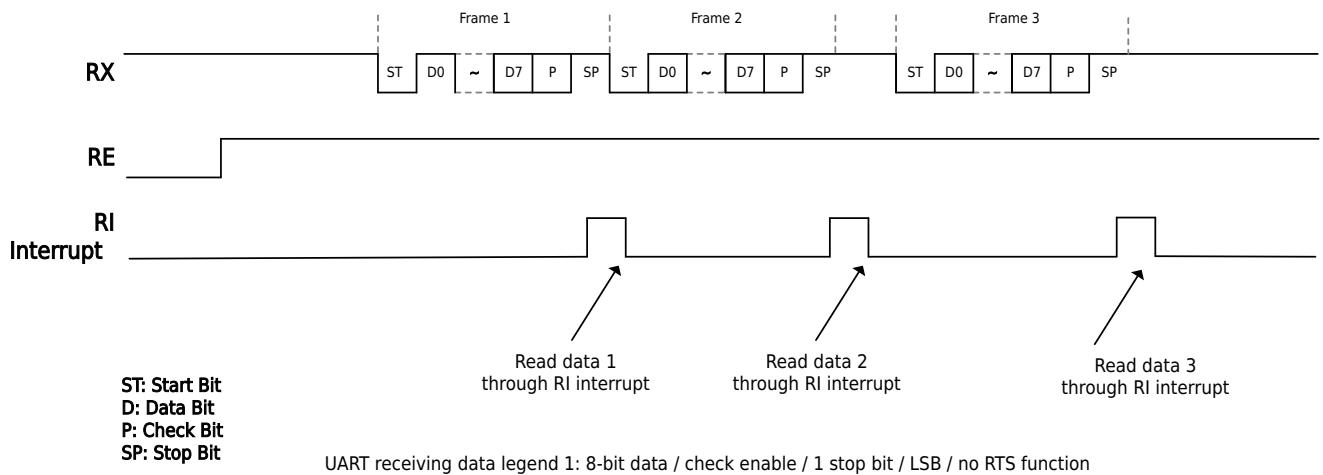


Figure 29-6 UART Receive Data Diagram 1

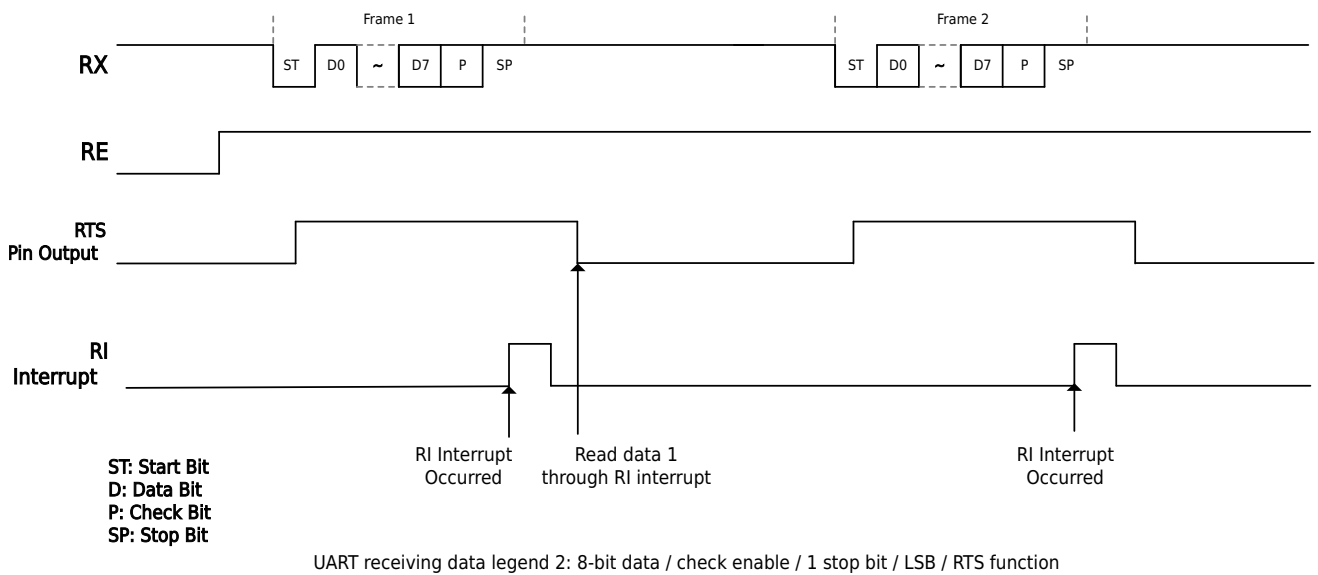


Figure 29-7 UART Receive Data Diagram 2

## Error Handling

During data reception, there are three types of receive errors: overflow error (USARTn\_SR.ORE), parity error (USARTn\_SR.PE), and frame error (USARTn\_SR.FE). If any type of reception error occurs, data reception is disabled. To restart data reception, all error flags can be cleared by writing to the corresponding clear register.

The condition for overflow error occurrence is when a new frame of data is received while the USARTn\_RDR.RDR register value remains unread. Therefore, the previously received frame data should be read before the last bit of the current frame is received.

The condition for check error occurrence is when a parity check error happens.

The condition for frame error occurrence is when the stop bit is at low level. For 2-stop-bit cases, only the first stop bit is checked.

When an overflow error occurs, the received data is lost and no RI interrupt is generated.

When a parity error occurs, the received data is transferred to USARTn\_RDR.RDR, and no RI interrupt is generated.

When a frame error occurs, the received data is transferred to USARTn\_RDR.RDR, and no RI interrupt is generated.

## Receiver Interrupt

The USART Mode receiver supports two types of interrupts: receive data register full interrupt (RI) and error interrupt (EI).

- USARTn\_CR.RIE=1, when no receive error occurs and data is transferred from the receive shift register to the receive data register, an RI interrupt is generated.
- USARTn\_CR.RIE=1, when an overflow error, parity error, or frame error occurs during reception, an EI interrupt is generated.

### 29.3.2.6 UART Receive TIMEOUT Function

The TIMEOUT Counter starts when the UART Receive Data stop bit is detected. If the next frame of Receive Data is not detected after the configured TIMEOUT period (set in units of receive bits), a TIMEOUT occurs. If CR1.RE=1 at this time, the TIMEOUT status bit USARTn\_SR.RTOF is set. If USARTn\_CR1.RE=0, the USARTn\_SR.RTOF bit will be set after USARTn\_CR1.RE becomes 1.

The TIMEOUT counter utilizes the counter of the Timer0 module, the Timer0 counting mode must be configured as asynchronous count clock mode, with the asynchronous count clock source set to XTAL32 (the Timer0 count clock is provided by the clock tick generated from the UART Baud Rate Generator, and an XTAL32 crystal may not be required). The specific correspondence is as follows:

| USART1                | USART2                | USART3                | USART4                | USART5                | USART6                | USART7                | USART8                | USART9                | USART10               |
|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|
| Timer0_1<br>Channel A | Timer0_1<br>Channel B | Timer0_3<br>Channel A | Timer0_3<br>Channel B | Timer0_4<br>Channel A | Timer0_2<br>Channel A | Timer0_2<br>Channel B | Timer0_4<br>Channel B | Timer0_5<br>Channel A | Timer0_5<br>Channel B |

### TIMEOUT Function Timer0 Compare Counter Value Setting

Timer0 is a 16-bit counter. The maximum count clock frequency division selectable is 1024. The TMR0\_CMPA<B>R value setting calculation formula is as follows:

$$\text{CMPA<B>R} = \frac{\text{RTB}}{2^{\text{CKDIVA<B>}}} - \alpha$$

Where:

- CMPA<B>R: TMR0\_CMPA<B>R register value, round up the calculation result to the nearest integer.
- $\alpha$ : Delay caused by the Timer0 asynchronous counting synchronization circuit  
 When the count clock is not divided,  $\alpha=7$ ;  
 When the count clock is divided by 2,  $\alpha=5$ ;  
 When the count clock is divided by 4, 8, or 16,  $\alpha=3$ .  
 When the count clock frequency division is 32 or above,  $\alpha=2$ .
- RTB: Receive Timeout Bits, minimum value=Receive Data Frame length +  $\alpha \times 2^{\text{CKDIVA<B>}}$   
 The actual TIMEOUT occurrence time has a certain deviation from the RTB value, with an error  $\leq 2^{\text{CKDIVA<B>}}$ .
- CKDIRA<B>: TMR0\_BCONR.CKDIVA<B> bit register value

### TIMEOUT Function Setting Procedure

- Step1.** Set the USARTn\_CR1 register to its reset value;
- Step2.** Configure the pins required for UART operation;
- Step3.** Select the clock source via the USARTn\_CR2.CLKC bits (if the internal clock source is selected, set CR2.CLKC=0b01).
- Step4.** Configure the USARTn\_CR1, USARTn\_CR2, and USARTn\_CR3 Registers;

- Step5.** Set USARTn\_PR to select the prescaler value, and configure the USARTn\_BRR register for communication baud rate (no setting required when the clock source is an external clock source);
- Step6.** USARTn\_CR1.RTOE=1. If interrupts are required, set USARTn\_CR1.RTOIE=1.
- Step7.** Set TMR0\_BCONR.CSTA<B>=0;
- Step8.** Set TMR0\_CNTR<B>R to 0, configure the TMR0\_CMPA<B>R register and TMR0\_BCONR.CKDIVA<B> register to determine the TIMEOUT period;
- Step9.** Set TMR0\_BCONR.HCLEA<B>=1, TMR0\_BCONR.HSTAA<B>=1, TMR0\_BCONR.ASYNCLKA<B>=1, TMR0\_BCONR.SYNCLKA<B>=0, TMR0\_BCONR.SYNSEA<B>=1;
- Step10.** Enable the receiver (USARTn\_CR1.RE=1). If receive interrupt is required, set USARTn\_CR1.RIE=1.
- Step11.** Upon detecting TIMEOUT, follow these steps to disable the TMR0 timer and clear the USARTn\_SR.RTOF bit.  
Set both TMR0\_BCONR.SYNCLKA<B>=1 and TMR0\_BCONR.SYNA<B>=0, then simultaneously set TMR0\_BCONR.CSTA<B>=0, TMR0\_BCONR.SYNCLKA<B>=0, and TMR0\_BCONR.SYNSEA<B>=1. Clear the USARTn\_SR.RTOF bit by writing to USARTn\_CR1.CRTOF.

### 29.3.2.7 RX Line Wake-up Stop Mode Function

When the UART communication is idle, the system can enter Stop mode to reduce current consumption. Without altering the UART PORT settings, UART Unit 1 can wake up the system from Stop mode via the RX line. The specific steps are as follows:

- Step1.** When UART communication is idle, set the USART\_1\_WUPI interrupt vector and enable the UART receive signal line wake-up Stop mode function by setting the INTC\_WKEN.RXWKEN bit.
- Step2.** The system enters Stop mode;
- Step3.** When the system detects a falling edge on the RX line, it exits Stop mode and disables this function in the USART\_1\_WUPI interrupt handler.

Please note that when the communication party needs to wake up the system, it should send a wake-up frame (recommended as 0x00). This data will not be received by UART and will not set the related flag. Additionally, the communication party must wait for the wake-up time required by the system Stop Mode before initiating UART communication data transmission.

UART RX line wake-up function can filter noise on the RX line. For details, refer to the [USART\\_NFC Register](#).

### 29.3.2.8 UART Half-duplex Communication Mode

UART mode supports single-wire half-duplex operation. In UART mode, single-wire half-duplex mode is enabled by setting USARTn\_CR3.HDSEL=1.

In single-wire half-duplex mode:

- The TX and RX lines are internally connected, and the RX pin is no longer used.
- When there is no data transmission and the TX function is disabled, the TX pin is in a released state. Therefore, when using the single-wire half-duplex mode, the TX line is pulled up to prevent floating input when no data is being transmitted.
- When there is no data transmission and the transmit function is enabled, the TX pin outputs a high level.

In addition, the half-duplex mode is similar to communication in normal UART mode. Please note that the transmission process will not be blocked by hardware. As long as data is written when

USARTn\_CR1.TE=1, the transmission will proceed. Therefore, conflicts on the line must be managed by software.

In half-duplex communication, to prevent the UART from receiving its own transmitted data, it is recommended to set USARTn\_CR1.RE=0.

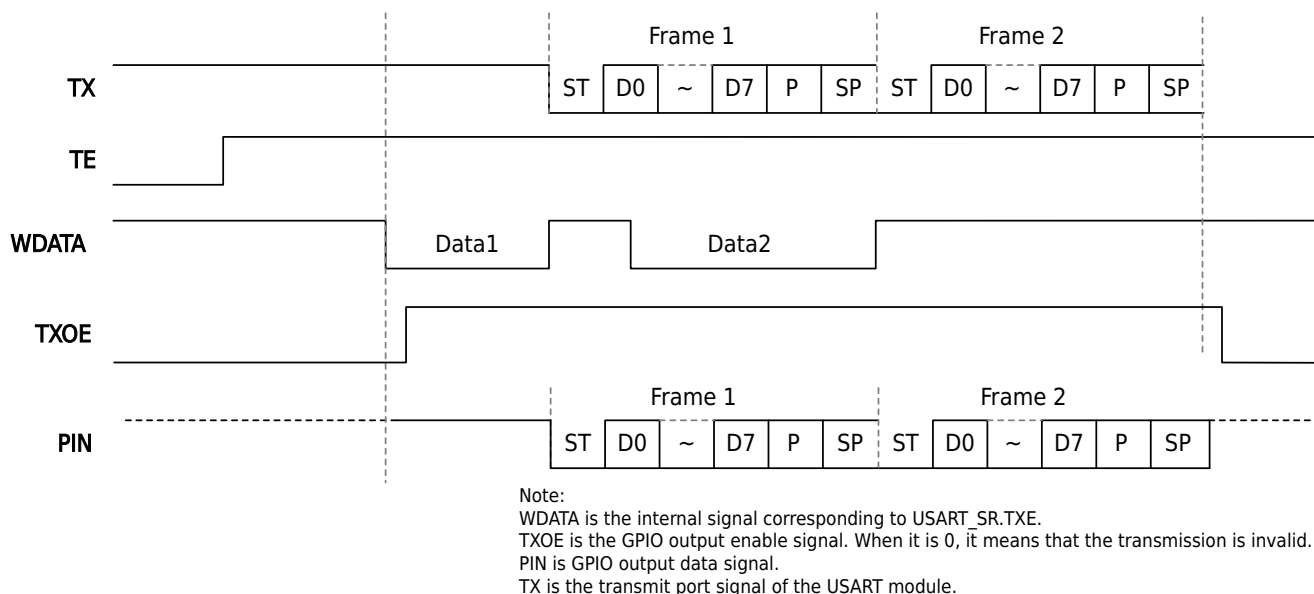


Figure 29-8 Half-Duplex Operation Timing

### 29.3.2.9 RS485 Driver Enable Function

UART Mode supports driver enable mode, sharing the output signal with RTS. First configure USART\_CR3.DEM to 1, then set USART\_CR3.RTSE to 1.

By setting the USART\_CR3.DEM to enable the driver function, users can activate external transceiver control through the DE (Driver Enable) signal. The assertion time programmed by the USART\_CR3.DEA is the time between DE signal activation and the start of the START bit.

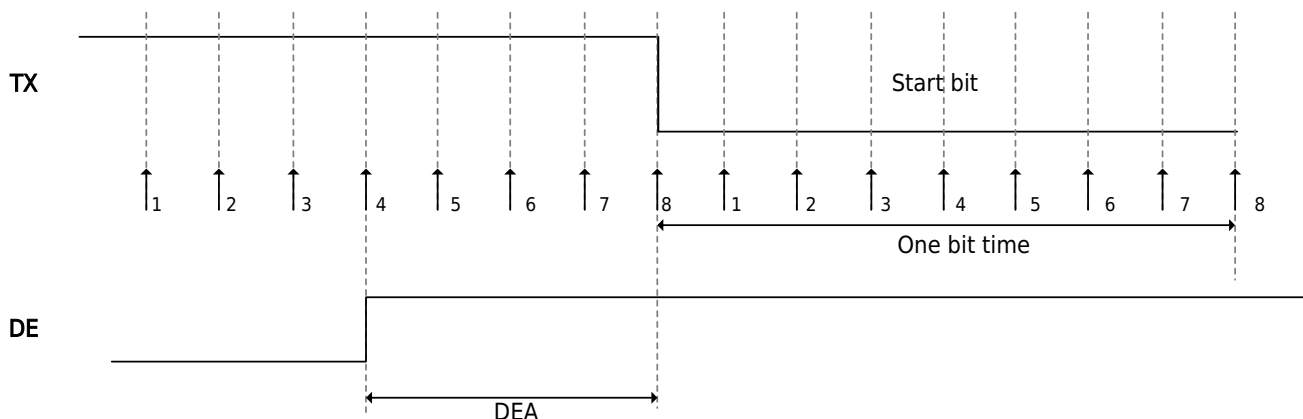


Figure 29-9 OVER8=1 DE Asserted

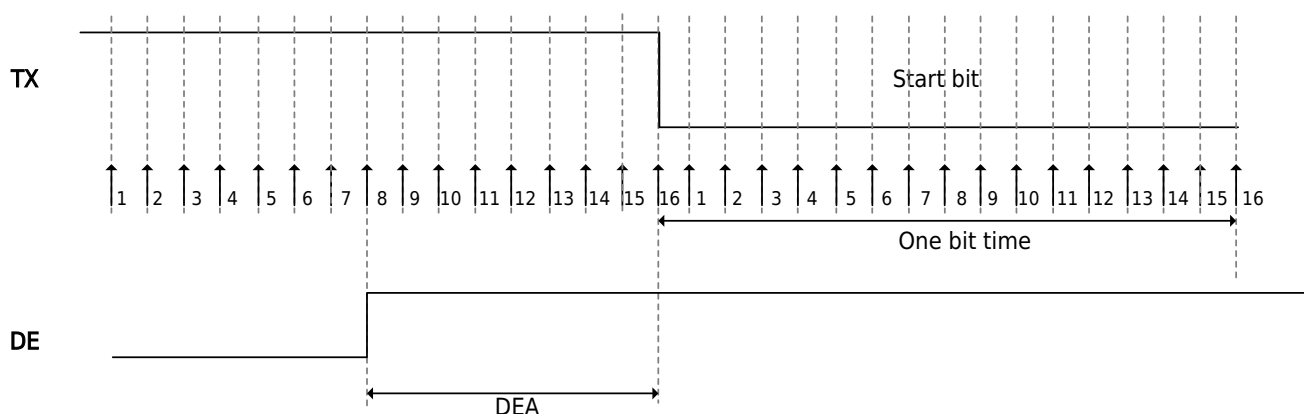


Figure 29-10 OVER8=0 DE Asserted

The de-assertion time programmed by the USART\_CR3.DED is the duration between the end of the last stop bit and the DE signal deactivation. The polarity of the DE signal can be configured using the USART\_CR3.DEP.

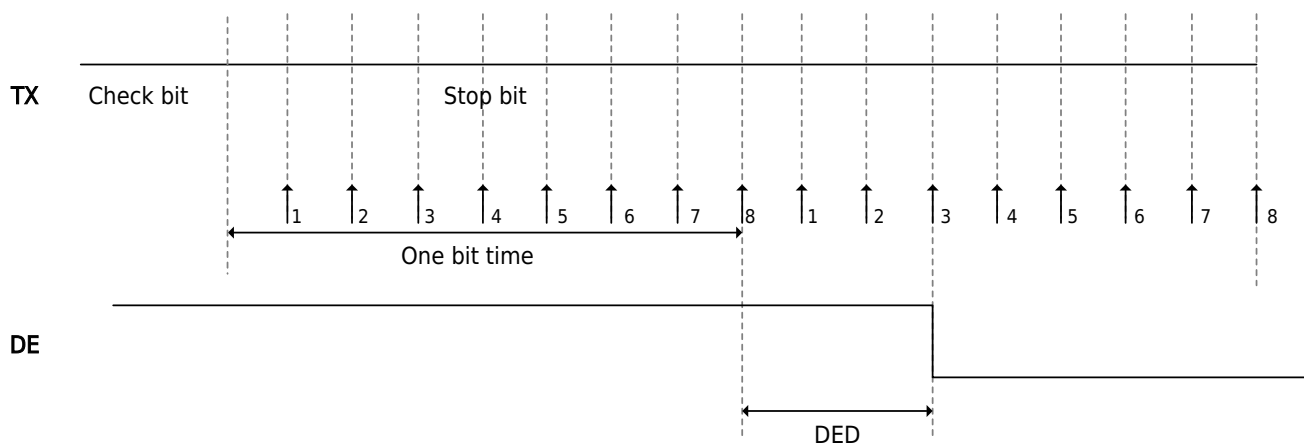


Figure 29-11 OVER8=1 DE De-assertion

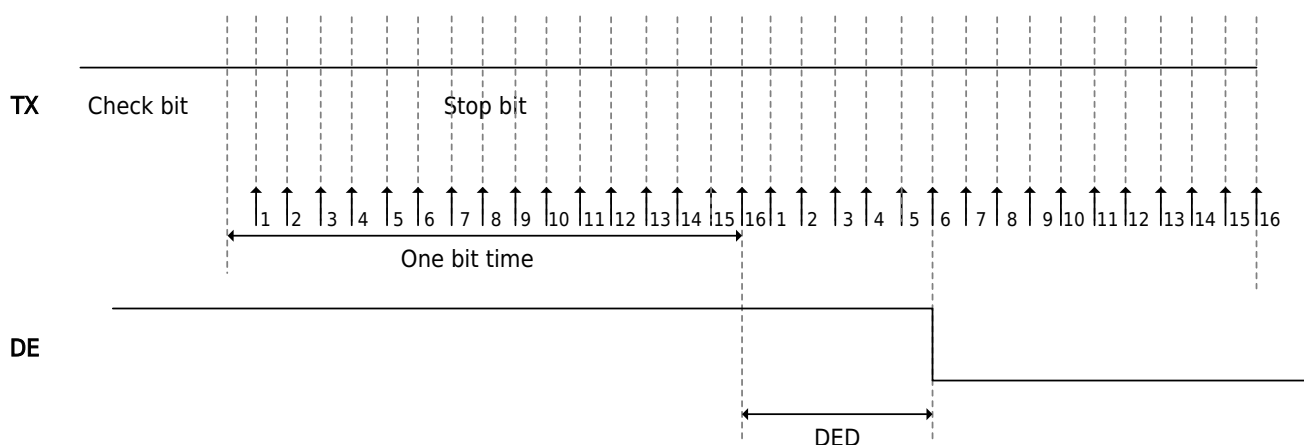


Figure 29-12 OVER8=0 DE De-assertion

### 29.3.2.10 Interrupt and Events

Table 29-5 UART Interrupt/Event Table

| Function Name           | Enable Bit (Interrupt only) | Flag Bit    | Used as Event Source |
|-------------------------|-----------------------------|-------------|----------------------|
| Receive Error Interrupt | RIE                         | ORE, FE, PE | Enable               |

| Function Name                          | Enable Bit (Interrupt only) | Flag Bit | Used as Event Source |
|----------------------------------------|-----------------------------|----------|----------------------|
| Receive Data Register Full Interrupt   | RIE                         | RXNE     | Enable               |
| Transmit Data Register Empty Interrupt | TXEIE                       | TXE      | Enable               |
| Transmit Completion Interrupt          | TCIE                        | TC       | Enable               |
| Transmit Completion Interrupt          | TENDIE                      | TEND     | Enable               |
| TIMEOUT Interrupt                      | RTOIE                       | RTOF     | Enable               |
| RX Line Wake-up Stop Mode Interrupt    | INTC_WKEN.RXWKEN            | -        | Disable              |

### 29.3.3 Multiprocessor Communication

#### 29.3.3.1 Function Introduction

Multiprocessor communication mode refers to a communication method where multiple processors share a common communication line. The processors are divided into Transmit Stations and Receive Stations, with each Receive Station having its own unique ID. The types of data transmitted by the Transmit Station include Receive Station ID and communication data. By adding the MPB bit in the Data Format, it distinguishes whether the current transmission is the Receive Station ID or communication data. When the MPB bit is 0, the current frame contains communication data; when the MPB bit is 1, the current frame carries the Receive Station ID. All Receive Stations can receive the ID sent by the Transmit Station and compare it with their own ID. If they match, the Receive Data is accepted; otherwise, the station enters Silent Mode (neither receiving data nor setting related Receive Flags) until the next ID is received.

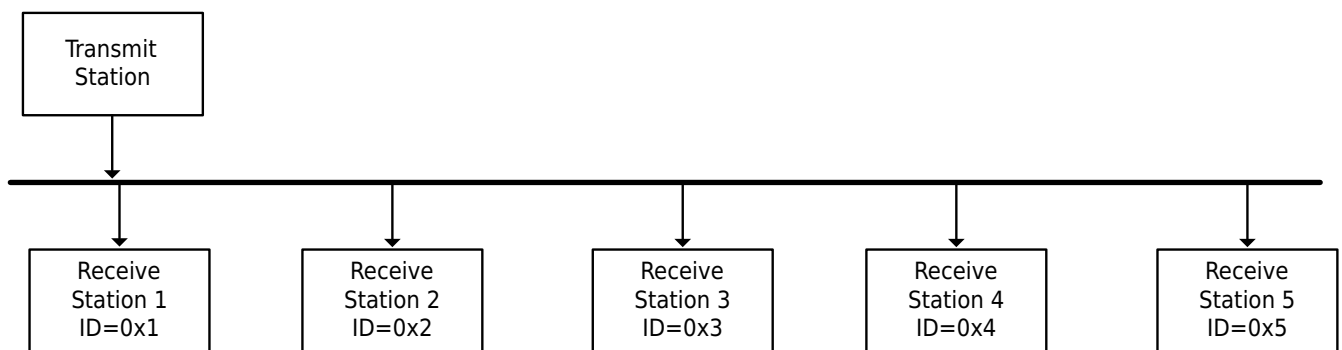


Figure 29-13 Multiprocessor Communication Legend

#### 29.3.3.2 Data Format

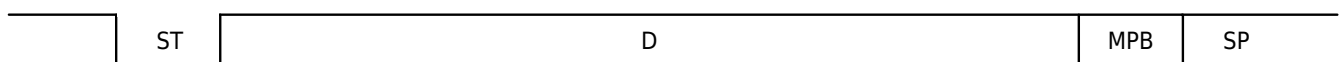


Figure 29-14 Multiprocessor Mode Data Format

- **Start Bit (ST)** : The start bit is always composed of one bit at low level.
- **Data Bit (D)** : The data bit can be configured as 8-bit or 9-bit.
- **Multiprocessor bit (MPB)** : The multiprocessor bit can be configured to transmit communication data or send IDs.



- **Stop Bit (SP)** : The stop bit is fixed at high level and can be configured as 1 bit or 2 bits.

### 29.3.3.3 Action Description

In multiprocessor mode, the check bit has no effect, and the multiprocessor bit is added. Other functions such as clock and interrupt remain the same as in UART Mode.

#### Transmit Station Action

- Step1.** Set the USARTn\_CR1 register to its reset value;
- Step2.** Set the required pins to be used;
- Step3.** Select the clock source via the USARTn\_CR2.CLKC bit;
- Step4.** Configure the USARTn\_CR1, USARTn\_CR2, and USARTn\_CR3 Registers;
- Step5.** Set USARTn\_PR to select the prescaler value, and configure the USARTn\_BRR register for communication baud rate (no setting required when the clock source is an external clock source);
- Step6.** Enable the transmitter (USARTn\_CR1.TE=1). If the transmit data register empty interrupt is required, set USARTn\_CR1.TXEIE=1.
- Step7.** Wait for the transmit data register to be empty, set USARTn\_TDR.MPID bit to 1 (Transmit ID), write the ID value to USARTn\_TDR, and transmit the ID.

**Note**

When the CTS function is enabled, data is transferred to the transmit shift register and transmission starts when the USARTn\_CTS input is at low level.

- Step8.** Set USARTn\_TDR.MPID bit to 0 (transmit data), write data to USARTn\_TDR to transmit data;

**Note**

When the CTS function is enabled, data is transferred to the transmit shift register and transmission starts when the USARTn\_CTS input is at low level.

- Step9.** If continuous data transmission is required, repeat step 8. If the ID needs to be changed before transmitting data, repeat steps 7 and 8.
- Step10.** Check whether the transmission is complete by verifying the USARTn\_SR.TC bit. For continuous data transmission using transmit interrupts, the last data can be written via the TI interrupt by setting USARTn\_CR1.TXEIE to 0 and USARTn\_CR1.TCIE to 1. A transmit completion interrupt will be generated after the last frame transmission ends.

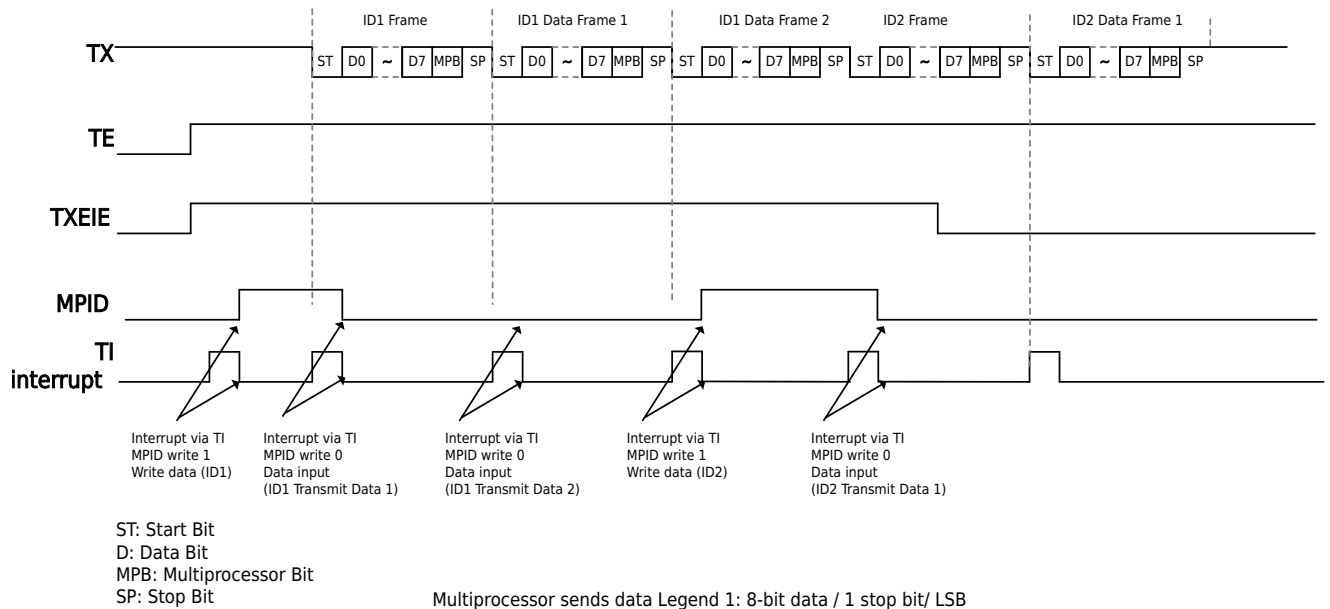


Figure 29-15 Multiprocessor Mode Transmit Data Diagram

### Receive Station Action

In multiprocessor mode, the receive station must ensure it receives every ID data and compares it with its own ID. If they match, it receives the data; otherwise, it enters silent mode (neither receiving data nor setting receive-related flags) until the next ID data is received. This function is implemented via the USARTn\_CR1.SLME bit.

When USARTn\_CR1.SLME=0, receive data normally.

When USARTn\_CR1.SLME=1, unless data with the MPB bit set to 1 (ID) is received, receive data is disabled, no RI interrupt occurs, and error flags FE and ORE are not set. When data with the MPB bit set to 1 (ID) is received, the USARTn\_CR1.SLME bit is automatically cleared, enabling normal receive data and interrupt generation.

Action steps:

- Step1.** Set the USARTn\_CR1 register to its reset value;
- Step2.** Set the required pins to be used;
- Step3.** Select the clock source via the USARTn\_CR1.CLKC bit;
- Step4.** Configure the USARTn\_CR1, USARTn\_CR2, and USARTn\_CR3 Registers;
- Step5.** Set USARTn\_PR to select the prescaler value, and configure the USARTn\_BRR register for communication baud rate (no setting required when the clock source is an external clock source);
- Step6.** USARTn\_CR1.RE=1, USARTn\_CR1.SLME=1 (waiting to receive ID). If using receive interrupt, set USARTn\_CR1.RIE=1.
- Step7.** When the start bit is detected, the receiver loads the data into the receive shift register and checks the USARTn\_SR.MPB bit.
- Step8.** If USARTn\_SR.MPB=1, the USARTn\_CR1.SLME bit is automatically cleared, Receive Data normally, and the software compares the received ID with its own ID.
  - a. If the ID matches, receive data normally, an interrupt occurs, and error detection is performed, which is the same as UART receive data.
  - b. If the ID does not match, the software writes the USARTn\_CR1.SLME bit to 1 again and repeats step 8.

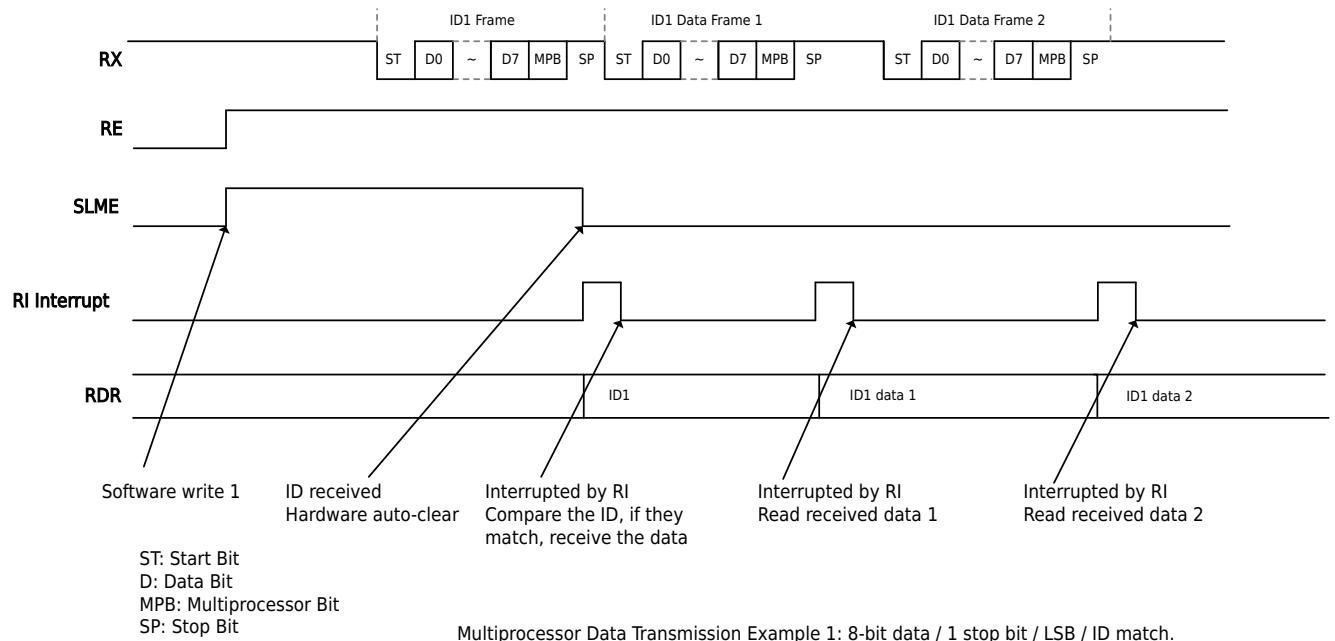


Figure 29-16 Multiprocessor Mode Receive Data Diagram 1

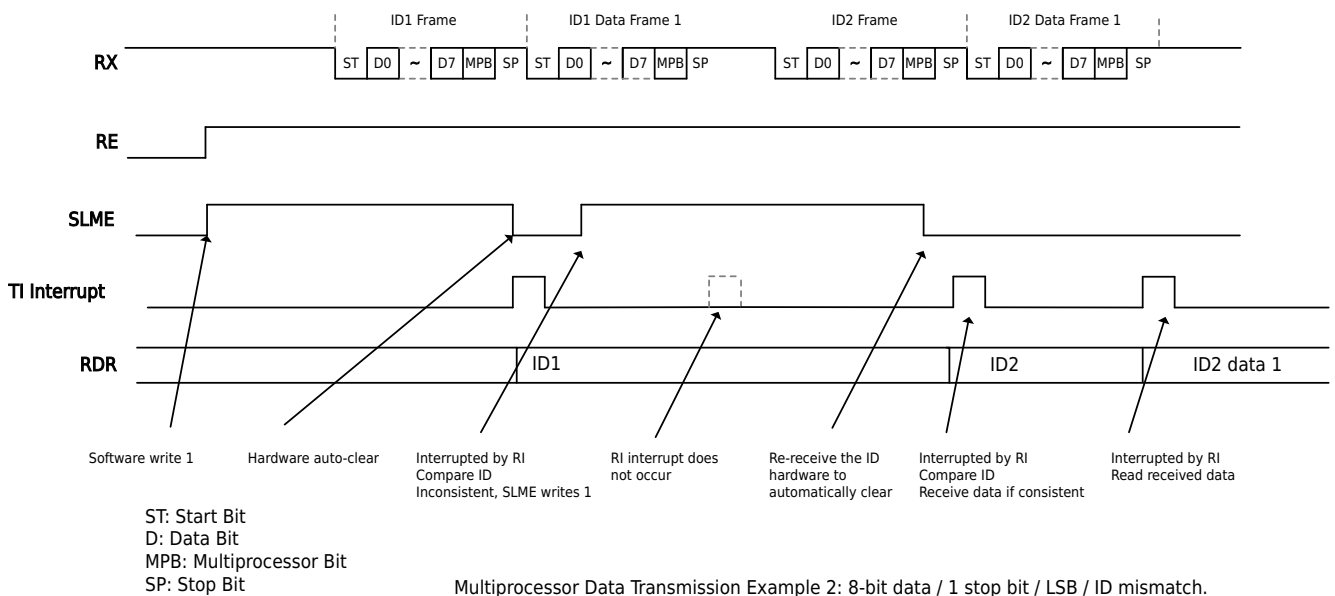


Figure 29-17 Multiprocessor Mode Receive Data Diagram 2

### 29.3.3.4 Interrupt and Events

In multiprocessor mode, the interrupt handling is identical to UART mode except for the absence of parity errors.

Table 29-6 Multiprocessor Mode Interrupt/Event Table

| Function Name                          | Enable Bit (Interrupt only) | Flag    | Used as Event Source |
|----------------------------------------|-----------------------------|---------|----------------------|
| Receive Error Interrupt                | RIE                         | ORE, FE | Enable               |
| Receive Data Register Full Interrupt   | RIE                         | RXNE    | Enable               |
| Transmit Data Register Empty Interrupt | TXEIE                       | TXE     | Enable               |
| Transmit Completion Interrupt          | TCIE                        | TC      | Enable               |

| Function Name                 | Enable Bit (Interrupt only) | Flag | Used as Event Source |
|-------------------------------|-----------------------------|------|----------------------|
| Transmit Completion Interrupt | TENDIE                      | TEND | Enable               |

### 29.3.4 UART-LIN

#### 29.3.4.1 Function Introduction

LIN stands for Local Interconnect Network, a low-speed (1-20kbps) serial communication protocol designed to reduce automotive network costs.

#### 29.3.4.2 LIN Data Format

A LIN frame consists of start bit + 8-bit data + 1 stop bit, with data transmitted and received in LSB format. The data behavior on the LIN Bus is as shown in the following figure.

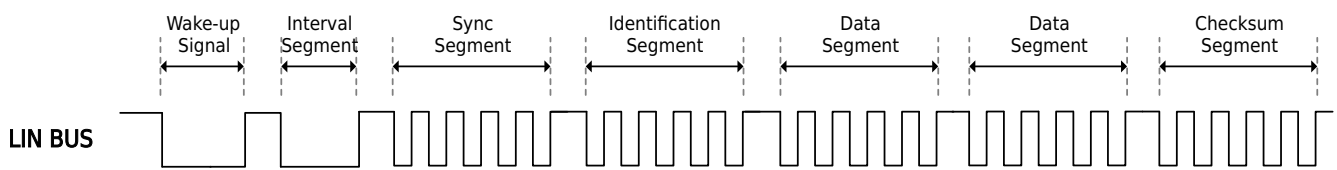


Figure 29-18 LIN Bus Data Behavior

#### 29.3.4.3 LIN Transmit

##### Wake-up Signal Frame Transmit

Transmit the corresponding signal frame by transmitting data 0x80.

##### Interval Segment Transmit

The synchronous interval segment indicates the start of a frame. This product supports hardware transmission of the LIN interval segment.

The width of the low level in the LIN interval segment is determined by the USARTn\_CR2.SBK register setting.

- When USARTn\_CR2.SBKM=0, writing 1 to USARTn\_CR2.SBK will cause the hardware to automatically send out the interval segment.
- When USARTn\_CR2.SBKM=1, after writing 1 to USARTn\_CR2.SBK, writing data 0x00 to the transmit data register USARTn\_TDR.TDR starts transmitting the interval segment.

After the interval segment transmission is completed, USARTn\_CR2.SBK is automatically cleared.

##### Synchronization Segment Transmit

Transmit data 0x55 to achieve the transmission of the synchronization segment.

##### Bus Error Detection

When USARTn\_CR2.BEE=1, the USARTn\_SR.BE flag is set if the hardware detects a mismatch between the data on the bus and the transmitted data. When USARTn\_CR2.BEIE=1, the corresponding interrupt is generated.

#### 29.3.4.4 LIN Receive

##### Wake-up Signal Frame Detection

When USARTn\_CR2.WKUPE=1, the hardware automatically detects the wake-up signal. Upon detecting a low-level width on the RX line greater than or equal to 2.5 bit data widths (130μs at 19.2kbps), the

USARTn\_SR.WKUP flag is set. When USARTn\_CR2.WKUPE=1, the corresponding interrupt is generated. Upon detecting a wake-up signal frame, clear the USARTn\_SR.WKUP flag and read the USARTn\_RDR.RDR register.

The USARTn\_CR2.WKUPE register should be set to 1 only when the system needs to wait for a wake-up signal; otherwise, it must be set to 0.

### Synchronous Interval Segment Detection

This product supports hardware-based synchronous interval segment detection. When the low-level width on the communication line is greater than or equal to the value set in the USARTn\_CR2.LBDL register and a break delimiter is detected, the USARTn\_SR.LBD flag is set. When USARTn\_CR2.LBDIE=1, the corresponding interrupt is generated.

Note that during synchronous interval segment detection, the USARTn\_SR.FE flag will be set. While clearing the USARTn\_SR.LBD flag, the USARTn\_SR.FE flag must also be cleared, and the USARTn\_RDR.RDR register should be read (the read value is 0x00).

### Synchronization Segment Detection and Baud Rate Measurement

Upon detecting the interval segment, the hardware automatically measures the frequency of the synchronization segment. The clock for measuring the counter is set by the USARTn\_PR.LBMPSC bit. The counter value is stored in the USARTn\_LBMC register. The baud rate of the master node can be obtained by dividing the selected counter clock frequency by the value in the USARTn\_LBMC register.

Please note that reading the USARTn\_LBMC register must be performed after the synchronization segment reception is completed.

### Receive Data Setting Procedure

- Step1.** Set the USARTn\_CR1 register to its reset value;
- Step2.** Configure the pins required for UART operation;
- Step3.** Select the clock source via the USARTn\_CR2.CLKC bit;
- Step4.** Configure the USARTn\_CR1 (RE=1, RIE=0), USARTn\_CR2, and USARTn\_CR3 registers.
- Step5.** (Optional) wake-up signal frame detection:
  - a. Set USARTn\_CR2.WKUPE=1 and wait for the wake-up signal;
  - b. Wake-up signal detected, USARTn\_CR2.WKUPE=0, waiting for synchronization segment interval segment.
- Step6.** (Optional) synchronization segment interval field detection:
  - a. Upon detecting a break in the sync interval, clear the USARTn\_SR.FE/LBD flag and verify that the received data is 0x00.
- Step7.** (Optional) Synchronization segment frequency measurement and communication Baud rate calculation:
  - a. To use the receive interrupt, set USARTn\_CR1.RIE=1.
  - b. Upon detecting the synchronization gap field and completing the reception of the synchronization segment, read the USARTn\_LBMC register to calculate the baud rate. If the received data is 0x55, it indicates the baud rate error is within the acceptable range. If the received data is not 0x55, user calibration of the baud rate is mandatory. The user should recalibrate the baud rate based on the USARTn\_LBMC register.
- Step8.** Receive identification segment (process is the same as UART receive data process);
- Step9.** Receive Data (process is the same as UART receive data process);
- Step10.** Receive checksum (process is the same as UART receive data process).

### 29.3.4.5 Interrupt and Events

Table 29-7 LIN Interrupt/Event Table

| Function Name                                       | Enable Bit (Interrupt only) | Flag Bit | Used as Event Source |
|-----------------------------------------------------|-----------------------------|----------|----------------------|
| Error Interrupt                                     | RIE                         | ORE, FE  | Enable               |
|                                                     | BEIE                        | BE       | Enable               |
| Wake-up Signal/Interval Segment Detection Interrupt | WKUPIE                      | WKUP     | Enable               |
|                                                     | LBDIE                       | LBD      | Enable               |
| Receive Data Register Full Interrupt                | RIE                         | RXNE     | Enable               |
| Transmit Data Register Empty Interrupt              | TXEIE                       | TXE      | Enable               |
| Transmit Completion Interrupt                       | TCIE                        | TC       | Enable               |
| Transmit Completion Interrupt                       | TENDIE                      | TEND     | Enable               |

### 29.3.5 Smart Card

#### 29.3.5.1 Connection Schematic Diagram

Supports the smart card communication protocol specified in ISO/IEC 7816-3. The figure below shows the connection schematic diagram for smart card mode.

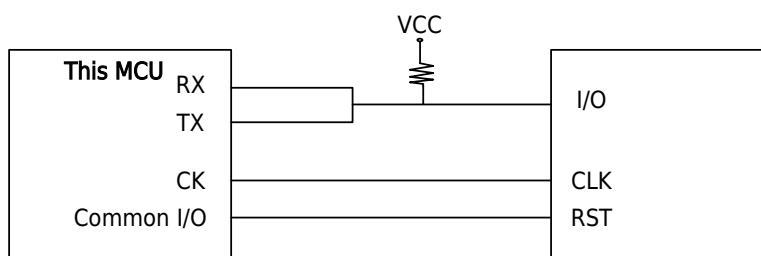


Figure 29-19 Smart Card Connection Schematic Diagram

#### 29.3.5.2 Clock

##### Internal Clock Source

In smart card mode, only the clock generated by the internal baud rate generator can be used as the clock source.

The basic number of clock periods for 1-bit data transmission is determined by the USARTn\_CR3.BCN setting.

The clock output in smart card mode is controlled by setting the USARTn\_CR2.CLKC bit in the register.

##### Maximum Baud Rate

When the internal clock source is selected, the baud rate generated by the internal baud rate generator is calculated using the following formula:

$$B = \frac{C}{2 \times BCN \times (DIV\_Integer + 1)}$$

Where:

- B: Baud rate, unit: Mbps
- C: When USARTn\_PR.ULBREN is 0, the clock set by USARTn\_PR.PSC bit (PCLK1, PCLK1/4, PCLK1/16, PCLK1/64), unit: MHz  
When USARTn\_PR.ULBREN is 1, the clock set by USARTn\_PR.PSC bit (PCLK1/128, PCLK1/256, PCLK1/512, PCLK1/1024), unit: MHz
- DIV\_Integer: USARTn\_BRR.DIV\_Integer setting value
- BCN: USARTn\_CR3.BCN register setting value

When C is PCLK1, DIV\_Integer=0, and BCN=0, the baud rate reaches the maximum baud rate of PCLK1/64 (Mbps).

### Sampling and Reception Tolerance

Upon detecting the falling edge of RX, the USART synchronizes the receive data with the internal basic clock to initiate data reception. The receive data will be sampled at the center of the data.

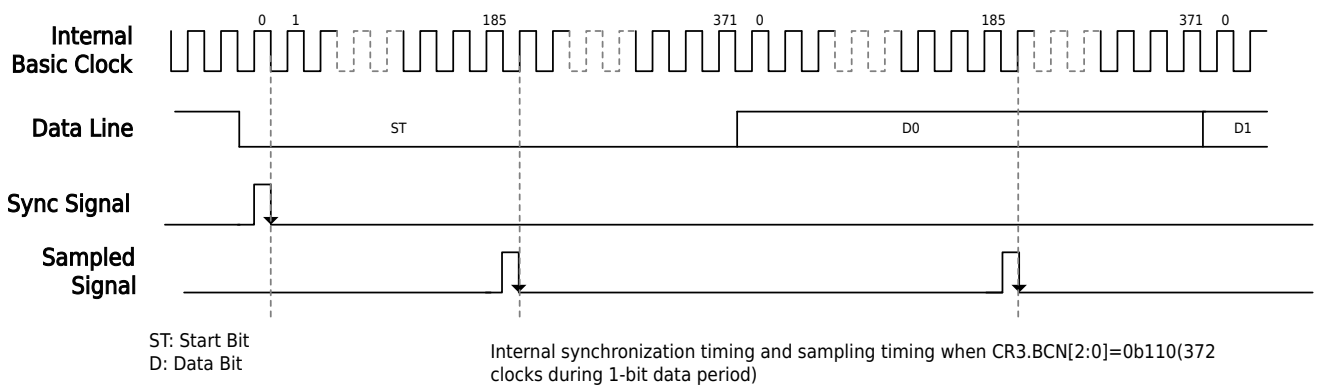


Figure 29-20 Smart Card Mode Synchronization Timing and Sample Timing Diagram

Receive tolerance calculation formula:

$$RM = \{0.5 \times (1 - \frac{1}{BCN}) - 9.5CFD\} \times 100\%$$

Where:

- RM: Reception Margin
- BCN: Number of clocks required for 1 bit of data transmission (set by USARTn\_CR3.BCN)
- CFD: Clock Frequency Deviation

#### 29.3.5.3 Data Format

In smart card mode, a frame of data consists of start bit, data bit and check bit.

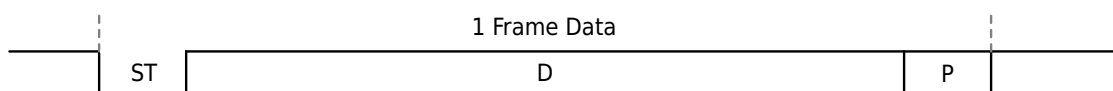


Figure 29-21 Smart Card Mode Data Format

- **Start Bit (ST)** : The start bit is always composed of one bit at low level.
- **Data Bit (D)** : The data bit is fixed as 8-bit data.
- **Check Bit (P)** : The check bit must be configured as 1-bit even parity.

#### 29.3.5.4 Initialization Setup Procedure for Smart Cards

- Step1.** Configure the USARTn\_CR1 register to the reset value;
- Step2.** Configure the required pins to be used;

- Step3.** Status register confirmed, USARTn\_SR register setting is reset to default value;
- Step4.** Configure the USARTn\_CR1, USARTn\_CR2, and USARTn\_CR3 Registers;
- Step5.** Set USARTn\_PR to select the prescaler value, and configure the USARTn\_BRR register for the communication baud rate.
- Step6.** USARTn\_CR2.CLKC bits configure clock control;
- Step7.** Configuration of USARTn\_CR1 register (TE, RE, RIE, TXEIE bits) : Except for self-test, do not simultaneously set both TE and RE to 1.

When switching from transmit mode to receive mode, or from receive mode to transmit mode, steps 1 to 7 above must be reconfigured.

### 29.3.5.5 Smart Card Mode Action Description

In smart card mode, the flag bit for the TI interrupt (transmit data empty interrupt) is the USARTn\_SR.TC bit. A TI interrupt is generated when USARTn\_SR.TC=1 and USARTn\_CR1.TXEIE=1.

#### Function Overview

During data transmission, there must be a protection time of at least 2etu (Elementary Time Unit) between two frames (from the end of the check bit to the start of the next frame's start bit).

If an error signal from the receiver is detected during data transmission, the data will be automatically retransmitted after 2etu.

When a parity error occurs in the received data, a low-level error signal is transmitted for 1etu. The error signal is sent with a timing of 10.5 etu after the start of reception.

#### Transmission Instructions

After a frame of data is transmitted, if an error signal from the receiver is detected, USARTn\_SR.FE is set to 1 (an error interrupt occurs if USARTn\_CR1.RIE=1). The USARTn\_SR.TC flag is not set, and the data is auto retransmitted. The USARTn\_SR.FE bit must be cleared before receiving the next frame's check bit.

After a frame of data is transmitted without errors, the USARTn\_SR.TC flag is set, and a TI interrupt occurs when USARTn\_CR1.TXEIE=1. Writing the data again enables continuous data transmission.

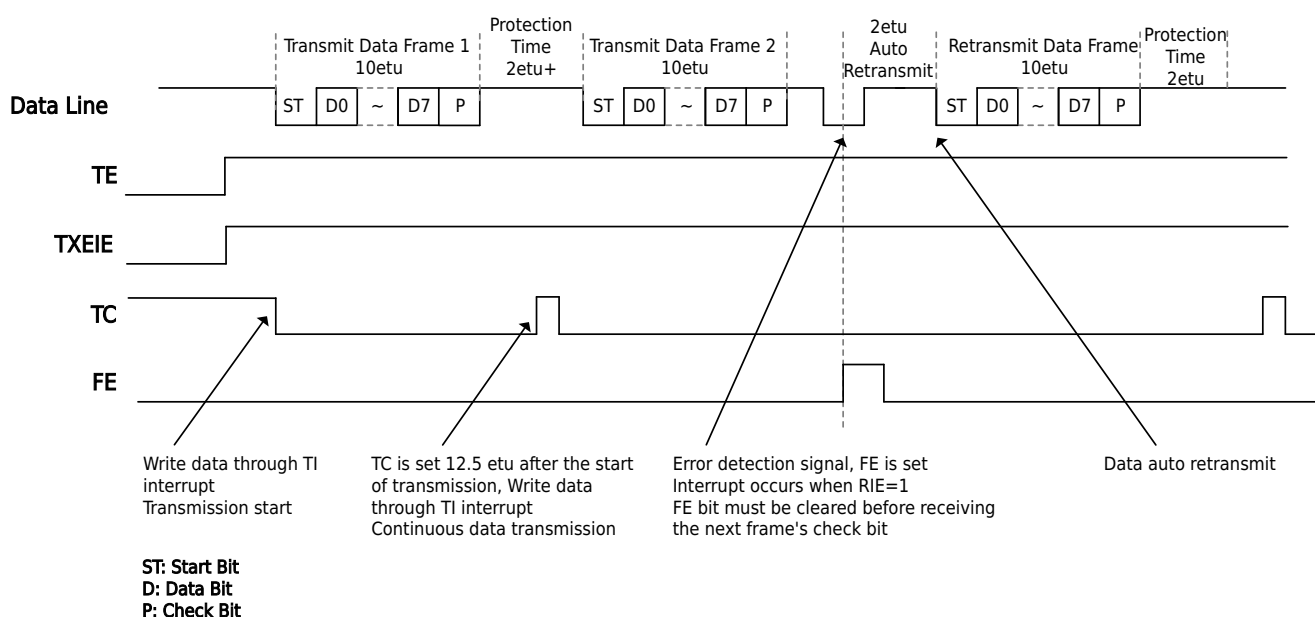


Figure 29-22 Smart Card Mode Transmit Data Diagram



## Receive Instructions

When receiving data, if a parity error is detected, USARTn\_SR.PE is set. When interrupt enable is active, an EI interrupt occurs. The USARTn\_SR.PE bit must be cleared before receiving the next frame's check bit.

When a parity error occurs, a low level of 1etu is transmitted as the error signal, requesting the sender to retransmit the data.

Normal receive data, the received data can be read via RI Interrupt, continuous reception.

Detect overflow error when receiving data.

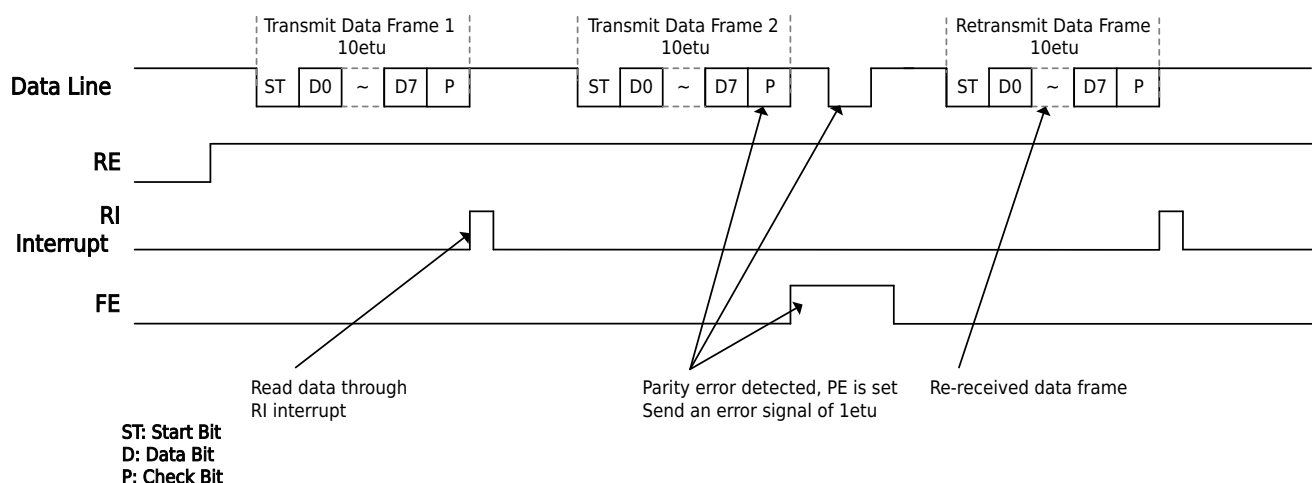


Figure 29-23 Smart Card Mode Receive Data Diagram

### 29.3.5.6 Interrupt and Events

Table 29-8 Smart Card Mode Interrupt/Event Table

| Function Name                 | Enable Bit (Interrupt only) | Flag Bit    | Used as Event Source |
|-------------------------------|-----------------------------|-------------|----------------------|
| Error Interrupt               | RIE                         | ORE, PE, FE | Enable               |
| Receive Data Full Interrupt   | RIE                         | RXNE        | Enable               |
| Transmit Data Empty Interrupt | TXEIE                       | TC          | Enable               |

## 29.3.6 Clock Synchronization Mode

### 29.3.6.1 Clock

The clock synchronization mode can select either the clock generated by the internal baud rate generator (internal clock source) or the clock input from the USARTn\_CK pin (external clock source) as the communication clock source.

#### Internal Clock Source

The synchronization clock is output from the USARTn\_CK pin. A frame of data outputs 8 clock pulses. When neither transmitting nor receiving data, the clock output remains fixed at high level.

#### External Clock Source

The external clock source inputs the clock from the USARTn\_CK pin as the communication clock.

## Maximum Baud Rate

When the internal clock source is selected, the baud rate generated by the internal baud rate generator is calculated using the following formula:

$$B = \frac{C}{4 \times (\text{DIV\_Integer} + 1)}$$

Where:

- B: Baud rate, unit: Mbps
- C: When USARTn\_PR.ULBREN is 0, the clock set by USARTn\_PR.PSC bit (PCLK1, PCLK1/4, PCLK1/16, PCLK1/64), unit: MHz  
When USARTn\_PR.ULBREN is 1, the clock set by USARTn\_PR.PSC bit (PCLK1/128, PCLK1/256, PCLK1/512, PCLK1/1024), unit: MHz
- DIV\_Integer: USARTn\_BRR.DIV\_Integer setting value

When the internal clock source is selected and C is PCLK1 with DIV\_Integer=1, the maximum baud rate is PCLK1/8 (Mbps).



### Notice

In synchronous mode, setting DIV\_Integer to 0 is prohibited.

When using an external clock source, the maximum frequency requirement for the external input clock is PCLK1/6 (MHz), so the maximum baud rate is PCLK1/6 (Mbps).

Please note that in synchronous mode, the maximum communication baud rate is not only determined by the PCLK1-based calculation method described above but also needs to refer to the maximum communication baud rate specified in the "Electrical Specifications" chapter of the Datasheet.

### 29.3.6.2 Data Format

In clock synchronization mode, each frame of data consists of a fixed 8 bits. The transmission and reception of one frame requires 8 synchronization clock pulses. During data transmission, data is sent on the falling edge of the synchronization clock. During data reception, data is sampled on the rising edge of the synchronization clock.

The synchronization clock remains fixed at high level when there is no data transmission. After the last bit is transmitted, the communication line maintains the value of the last bit.

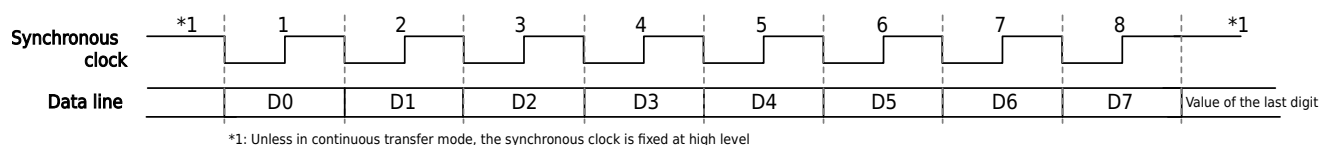


Figure 29-24 Clock Synchronization Mode Data Format

### 29.3.6.3 Modem Operation

Modem operations include the CTS function and RTS function. When USARTn\_CR3.RTSE=1, the RTS function is enabled. When USARTn\_CR3.CTSE=1, the CTS function is enabled.

#### CTS Function

The CTS function controls data transmission via the input of the USARTn\_CTS pin. Data can only be transmitted when the USARTn\_CTS pin input is at low level. If the USARTn\_CTS input transitions to high level during transmission, the ongoing data transmission is unaffected.

## RTS Function

The RTS function outputs a low level through the USARTn\_RTS pin to request the peer to transmit data.

The USARTn\_RTS pin outputs a low level when all of the following conditions are met:

- Receive enable (USARTn\_CR1.RE=1) and not currently receiving data
- There is no unread data in the USARTn\_RDR.RDR register (when USARTn\_CR1.RE=1)
- USARTn\_TDR.TDR update completed (when USARTn\_CR1.TE=1)
- No receiving errors

If all the above conditions cannot be met at the same time, USARTn\_RTS outputs a high level.

### 29.3.6.4 Transmitter

When USARTn\_CR1.TE bit is set to 1, the data in the transmit shift register is serially output at the USARTn\_TX pin, and the corresponding clock pulse is output at the USARTn\_CK pin.

The transmit data register USARTn\_TDR.TDR and the internal transmit shift register form a double buffer structure, which can transmit data continuously.

When transmitting data via the transmit data register empty interrupt or DMA, to ensure transmission accuracy, only one data write is enabled per request.

### Transmit Data Setting Procedure

- Step1.** Configure USARTn\_CR1, USARTn\_SR1 registers to reset values;
- Step2.** Configure the required pins to be used;
- Step3.** Select the clock source via the USARTn\_CR2.CLKC bit;
- Step4.** Configure the USARTn\_CR1, USARTn\_CR2, and USARTn\_CR3 registers;
- Step5.** Set USARTn\_PR to select the prescaler value, and configure the USARTn\_BRR register for communication baud rate (no setting required when the clock source is an external clock source);
- Step6.** Enable the transmitter (USARTn\_CR1.TE=1). If the transmit data register empty interrupt is required, set USARTn\_CR1.TXEIE=1.
- Step7.** Wait for the transmit data register to be empty, write communication data to USARTn\_TDR.TDR, data is transferred to the transmit shift register, transmission starts;



#### Note

When the CTS function is enabled, data is transferred to the transmit shift register and transmission starts when the USARTn\_CTS input is at low level.

- Step8.** To continuously transmit data, repeat step 7;
- Step9.** Check whether the transmission is complete by verifying the USARTn\_SR.TC bit. For continuous data transmission using transmit interrupts, the last data can be written via the TI interrupt by setting USARTn\_CR1.TXEIE to 0 and USARTn\_CR1.TCIE to 1. A transmit completion interrupt will be generated after the last frame transmission ends.

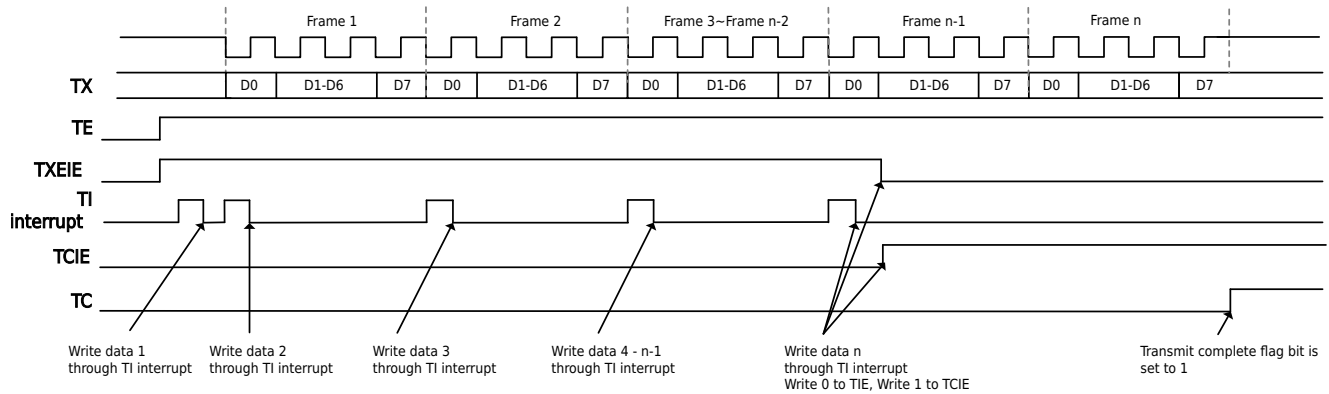


Figure 29-25 Data Transmission in Clock Synchronization Mode Diagram 1 (LSB/without CTS function)

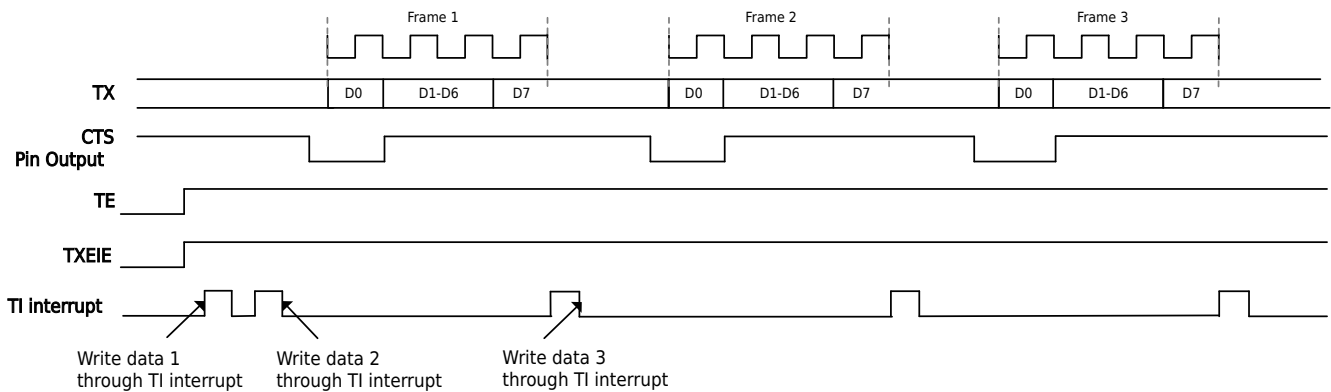


Figure 29-26 Data Transmission in Clock Synchronization Mode Diagram 2 (LSB/with CTS function)

### Transmitter Interrupt

The clock synchronous mode transmitter supports two interrupts, the transmit data register empty interrupt TI and the transmit completion interrupt TCI.

- TXEIE=1, when the value of USARTn\_TDR.TDR register is transferred to the transmit shift register, a TI interrupt is generated.
- TCIE=1, when the last bit of transmitted data is sent and the USARTn\_TDR.TDR register is not updated, a TCI interrupt is generated.

#### 29.3.6.5 Receiver

##### Receive Data Setting Procedure

- Step1.** Configure USARTn\_CR1, USARTn\_SR registers to reset values;
- Step2.** Configure the required pins to be used;
- Step3.** Select the clock source via the USARTn\_CR2.CLKC bit;
- Step4.** Configure the USARTn\_CR1, USARTn\_CR2, and USARTn\_CR3 registers;
- Step5.** Set USARTn\_PR to select the prescaler value, and configure the USARTn\_BRR register for communication baud rate (no setting required when the clock source is an external clock source);
- Step6.** Enable the receiver (USARTn\_CR1.RE=1). If receive interrupt is required, set USARTn\_CR1.RIE=1.



#### Note

When using the RTS function, USARTn\_RTS outputs a low level after RE=1.

- Step7.** Start receiving data in synchronization with the input synchronization clock or the internally generated synchronization clock, and receive data into the receive shift register.

- When an overflow error occurs, data is lost and the USARTn\_SR.ORE flag is set;
- When no error occurs, the received data is transferred to the USARTn\_RDR.RDR register, the USARTn\_SR.RXNE flag is set, and the currently received data is read before the last bit of the next frame is received. Repeating step 7 can realize the continuous data reception function.

**Note**

When using the RTS function, USARTn\_RTS outputs a low level after data is read.

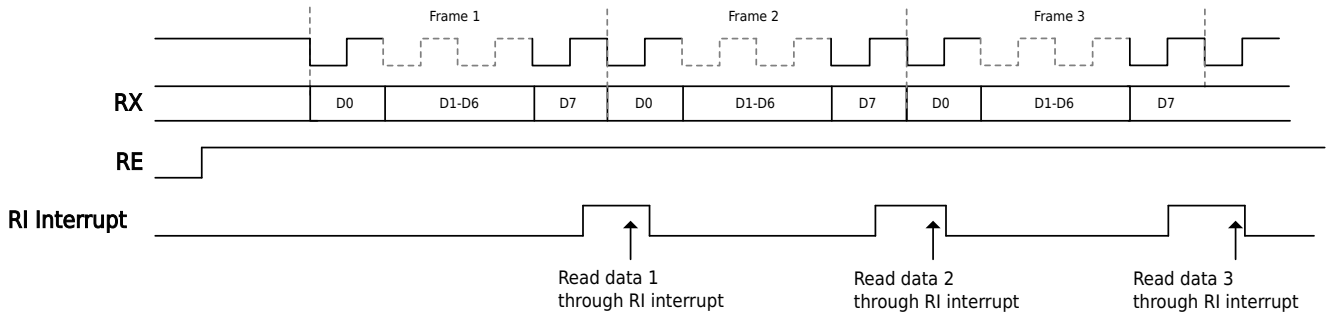


Figure 29-27 Data Reception in Clock Synchronization Mode Diagram 1 (LSB/without RTS function)

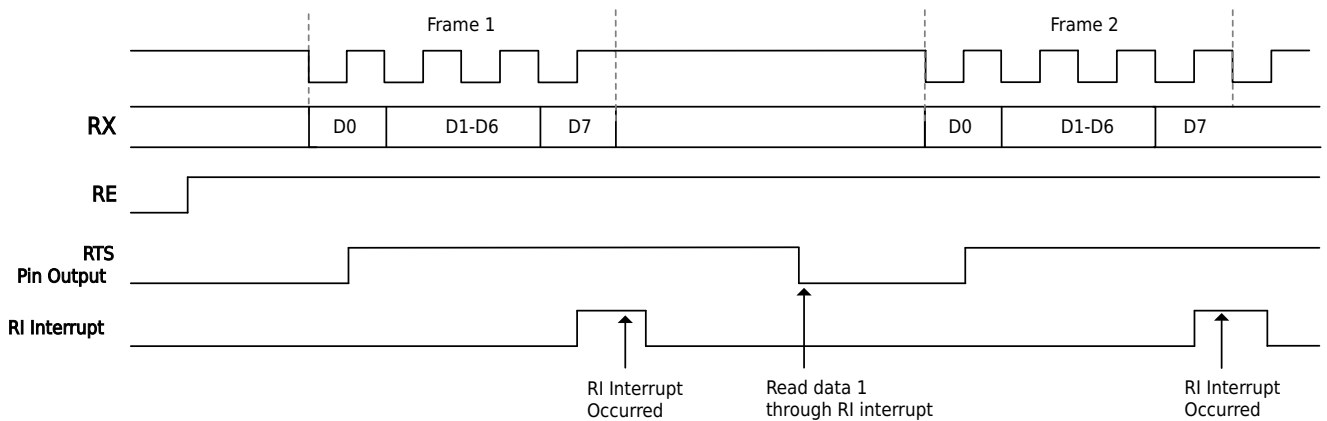


Figure 29-28 Data Reception in Clock Synchronization Mode Diagram 2 (LSB/with RTS function)

## Error Handling

When receiving data in clock synchronous mode, the reception error is an overflow error (USARTn\_SR.ORE). When a reception error occurs, data transmission and reception are halted. To resume data transmission, the error flags must be cleared by writing to the corresponding clear register.

The condition for an overflow error is that new data is received without reading the USARTn\_RDR.RDR value, so the data received in the previous frame should be read before the last bit of the current frame is received. When an overflow error occurs, the received data is lost and no RI interrupt is generated.

## Receiver Interrupt

The clock synchronous mode receiver supports two interrupts, the receive data register full interrupt RI and the receive error interrupt EI.

- When RIE=1, when data is transferred from the receive shift register to the receive data register, an RI interrupt is generated.
- When RIE=1, when a receive error (such as an overflow error) occurs during reception, an EI interrupt is generated.

### 29.3.6.6 Simultaneous Transmission and Reception

The USART synchronous clock mode supports full-duplex operation, enabling simultaneous data transmission and reception. To perform simultaneous transmission and reception, a command must set RE, TE, RIE, and TXEIE to 1. Other configuration procedures are the same as those for the transmitter and receiver individually.

### 29.3.6.7 Interrupt and Events

Table 29-9 Interrupt/Event Table in Synchronous Clock Mode

| Interrupt Name                         | Enable bit (Interrupt only) | Flag Bit | Used as Event Source |
|----------------------------------------|-----------------------------|----------|----------------------|
| Receive Error Interrupt                | RIE                         | ORE      | Enable               |
| Receive Data Register Full Interrupt   | RIE                         | RXNE     | Enable               |
| Transmit Data Register Empty Interrupt | TXEIE                       | TXE      | Enable               |
| Transmit Completion Interrupt          | TCIE                        | TC       | Enable               |
| Transmit Completion Interrupt          | TENDIE                      | TEND     | Enable               |

### 29.3.7 Digital Filter Function

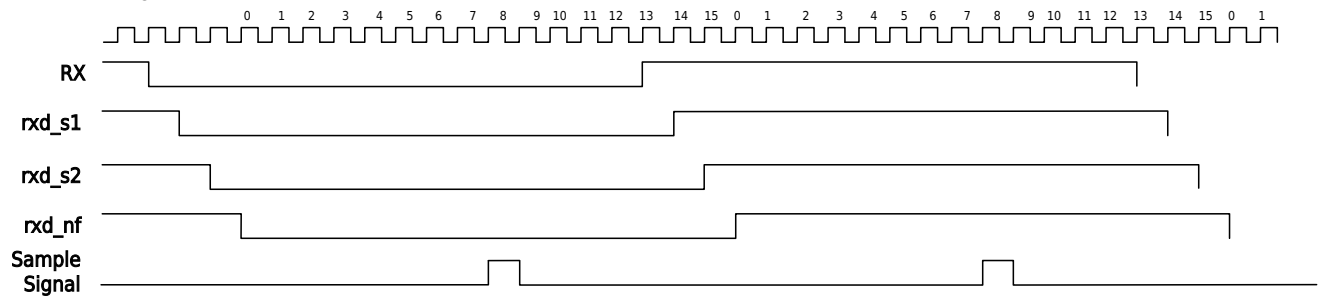
When USARTn\_CR1.NFE=1, the built-in digital filter function is enabled. The digital filter is only active in UART mode and can remove noise on the receive data line RX.

The built-in digital filter can filter out noise with a width less than 3/16 (USARTn\_CR1.OVER8=0) or 3/8 width (USARTn\_CR1.OVER8=1) of 1 bit.

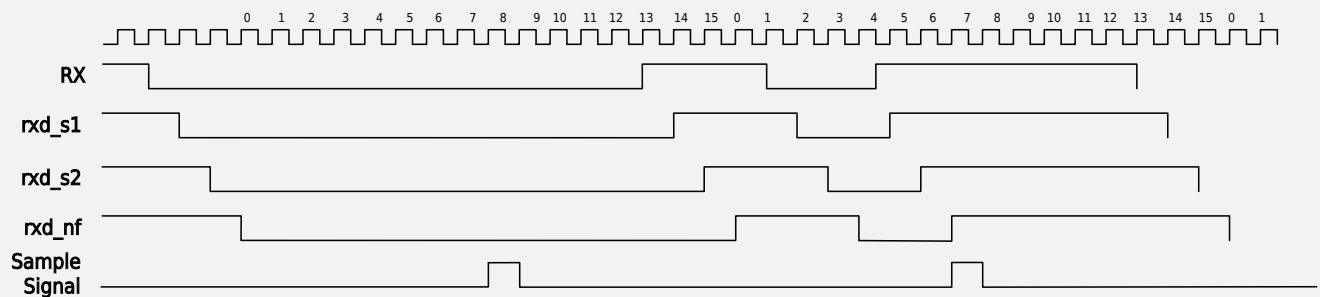
If the digital filter's clock stops and then starts again, the digital filter resumes operation from the state maintained when the clock stopped.

When USARTn\_CR.TE=0 and USARTn\_CR.RE=0, the Flip-Flop state inside the digital filter is reset to 1.

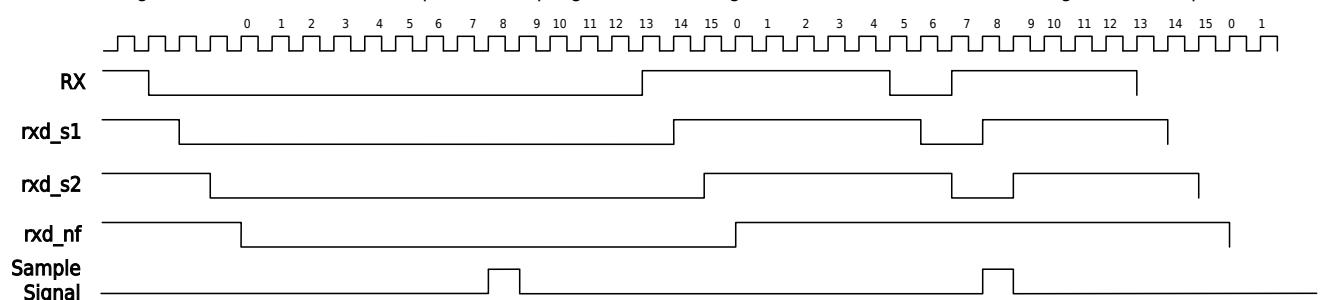
Enable Filtering, No Interference



Enable filtering, interference not filtered, no impact on sampling (when the sampling signal is valid, no interference signal is sampled)



Enable filtering, filter out interference, no impact on sampling (interference signals are filtered out, and correct signals are sampled)



Enable filtering, interference not filtered, affects sampling (interference not filtered out, interference signals are sampled)

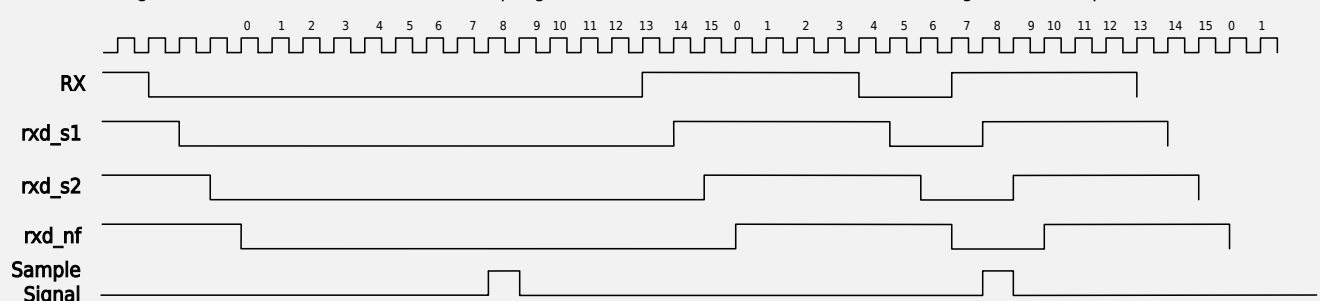


Figure 29-29 Digital Filter Function Diagram

- rxd\_s1: internal signal RX delayed by 1 sampling clocks
  - rxd\_s2: internal signal RX delayed by 2 sampling clocks
  - rxd\_nf: the actual sampled signal, determined by comparing the input signal RX, rxd\_s1, and rxd\_s2.
- Enabling the NF (Noise Filter) function performs two delayed operations on the input RX signal, generating three signals in total. Only when the input signal matches both delayed signals can the internal module perform signal toggling.

## 29.3.8 Interrupt

Table 29-10 USART Global Interrupt Summary Table

| Interrupt Name                                      | Symbol                    | Description                                                                                                                                                                                      |
|-----------------------------------------------------|---------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Error Interrupt                                     | USART_n_EI<br>n=1-10      | UART/Multiprocessor/Smart Card/Clock Synchronization Mode Receive Error Interrupt<br>LIN Error interrupt                                                                                         |
| Receive Data Register Full Interrupt                | USART_n_RI<br>n=1-10      | UART/Multiprocessor/LIN/Smart Card/Clock Synchronization Mode Receive Data Register Full Interrupt<br>LIN Error interrupt                                                                        |
| Transmit Data Register Empty Interrupt              | USART_n_TI<br>n=1-10      | UART/Multiprocessor/LIN/Smart Card/Clock Synchronization Mode Transmit Data Register Empty Interrupt                                                                                             |
| Transmit Completion Interrupt                       | USART_n_TCI<br>n=1-10     | UART/Multiprocessor/LIN/Smart Card/Clock Synchronization Mode Transmit Completion Interrupt (Writing transmit data to the Transmit Data Register clears the interrupt)                           |
| Transmit Completion Interrupt                       | USART_n_TENDI<br>n=1-10   | UART/Multiprocessor/LIN/Smart Card/Clock Synchronization Mode Transmit Completion Interrupt (Writing 1 to CR1.CTEND bit or writing transmit data to Transmit Data Register clears the interrupt) |
| Wake-up Signal/Interval Segment Detection Interrupt | USART_n_BRKWKPI<br>n=5/10 | LIN Wake-up Signal/Interval Segment Detection Interrupt                                                                                                                                          |
| UART Receive TIMEOUT Interrupt                      | USART_n_RTO<br>n=1-10     | UART Receive TIMEOUT Interrupt                                                                                                                                                                   |
| RX Line Wake-up Stop Mode Interrupt                 | USART_1_WUPI              | USART1 RX Line Wake-up Stop Mode Interrupt                                                                                                                                                       |



## 29.4 Register Description

This chapter provides a detailed description of the USART module control status and related registers. The function register bit is only valid when the corresponding function is configured for the channel; otherwise, it defaults to 0.

### 29.4.1 Register List

USART1\_NFC base address: 0x40055400

Table 29-11 USART1 Filter Control Register Offset

| Offset | Register   | Description                    | Width |
|--------|------------|--------------------------------|-------|
| 0x1C   | USART1_NFC | USART1 Filter Control Register | 32    |

- USART1 base address: 0x4001CC00
- USART2 base address: 0x4001D000
- USART3 base address: 0x4001D400
- USART4 base address: 0x4001D800
- USART5 base address: 0x4001DC00
- USART6 base address: 0x40020C00
- USART7 base address: 0x40021000
- USART8 base address: 0x40021400
- USART9 base address: 0x40021800
- USART10 base address: 0x40021C00

Table 29-12 Universal Synchronous Asynchronous Receiver/Transmitter (USART) Register Offset

| Offset | Register   | Description                              | Width |
|--------|------------|------------------------------------------|-------|
| 0x00   | USART_SR   | Status Register                          | 32    |
| 0x04   | USART_TDR  | Transmit Data Register                   | 16    |
| 0x06   | USART_RDR  | Receive Data Register                    | 16    |
| 0x08   | USART_BRR  | Baud Rate Register                       | 32    |
| 0x0C   | USART_CR1  | Control Register 1                       | 32    |
| 0x10   | USART_CR2  | Control Register 2                       | 32    |
| 0x14   | USART_CR3  | Control Register 3                       | 32    |
| 0x18   | USART_PR   | Prescaler Register                       | 32    |
| 0x1C   | USART_LBMC | LIN Baud Rate Measurement Count Register | 32    |

## 29.4.2 Status Register (USART\_SR)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |     |          |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |      |  |  |  |  |  |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-----|----------|----|----|----|----|---|---|---|---|---|---|---|---|---|---|--|------|--|--|--|--|--|
| 0x00   | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15  | 14       | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |  |      |  |  |  |  |  |
| Reset  | 0x000000C0   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |     |          |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |      |  |  |  |  |  |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | MPB | Reserved |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  | TEND |  |  |  |  |  |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RO  |          |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |      |  |  |  |  |  |

| Bit | Name | Function                            | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | Access |
|-----|------|-------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 10  | LBD  | LIN interval Segment Detection Flag | <ul style="list-style-type: none"> <li>When CR2.WKUPE=0, a low level on the RX line is detected with a width greater than or equal to the width set by the CR2.LBDL register, and a break delimiter is detected.</li> </ul> <p>LBD clear condition:</p> <ul style="list-style-type: none"> <li>Write 1 to the CR1.CLBD bit.</li> </ul> <div>  <b>Note</b><br/> RE=0 does not reset the LBD bit. </div>                                                                                                                                                                                                                                                                                                                                                                   | RO     |
| 9   | WKUP | LIN Wakeup Signal Detection Flag    | <p>Only valid in USART5/USART10:</p> <ul style="list-style-type: none"> <li>0b0: No Wakeup signal detected</li> <li>0b1: Wakeup signal detected</li> </ul> <p>WKUP set condition:</p> <ul style="list-style-type: none"> <li>When CR2.WKUPE=1, a low level on the RX line with a width greater than or equal to 2.5 bit durations (130μS at 19.2kbps) is detected.</li> </ul> <p>WKUP Clear condition:</p> <ul style="list-style-type: none"> <li>Write 1 to the CR1.CWKUP bit.</li> </ul> <div>  <b>Note</b><br/> RE=0 does not reset the WKUP bit.<br/><br/> After detecting the WKUP signal as 1, it indicates that this LIN node has exited Sleep mode. After setting register CR2.WKUPE=0, perform subsequent operations such as interframe space detection. </div> | RO     |
| 8   | RTOF | UART Receive TIMEOUT Flag Bit       | <ul style="list-style-type: none"> <li>0b0: No UART receive TIMEOUT</li> <li>0b1: UART receive TIMEOUT occurred</li> </ul> <p>RTOF set condition:</p> <ul style="list-style-type: none"> <li>From the detection of the STOP bit of the last frame data, no new Receive Data is detected after the set time has elapsed.</li> </ul> <p>RTOF clear condition:</p> <ul style="list-style-type: none"> <li>Write 1 to the CR1.CRTOF bit.</li> </ul> <div>  <b>Note</b><br/> RTOF is set to 1 by hardware, and is only set to 1 when both CR1.RE=1 and CR1.RTOE=1. When CR1.RE=0, the TIMEOUT function is enabled, but RTOF is not set to 1. </div>                                                                                                                         | RO     |

| Bit | Name | Function                             | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | Access |
|-----|------|--------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 7   | TXE  | Transmit Data Register Empty Flag    | <ul style="list-style-type: none"> <li>● 0b0: Data not transferred to shift register, transmit data register not empty</li> <li>● 0b1: Data transferred to shift register, transmit data register empty</li> </ul> <p>The TXE bit is set to 1 and cleared to 0 by hardware. The hardware clears TXE to 0 when data is not transferred to the shift register, and sets TXE to 1 when data is transferred to the shift register.</p> <div>  <b>Note</b><br/>           The TXE bit is valid in UART/Clock synchronization mode/LIN Mode.         </div>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | RO     |
| 6   | TC   | Transmit Completion Flag             | <ul style="list-style-type: none"> <li>● 0b0: Transmit data in progress</li> <li>● 0b1: Data transmission complete</li> </ul> <p><b>UART Mode, Multiprocessor Mode, LIN Mode, Clock Synchronization Mode</b></p> <ul style="list-style-type: none"> <li>● TC set condition:               <ul style="list-style-type: none"> <li>▶ TE=0: Transmission Disabled</li> <li>▶ When the last bit of a frame is transmitted, the value in the Transmit Data Register is not updated.</li> </ul> </li> <li>● TC clear condition:               <ul style="list-style-type: none"> <li>▶ When TE=1, write transmit data to the transmit data register.</li> </ul> </li> </ul> <p><b>Smart Card Mode</b></p> <ul style="list-style-type: none"> <li>● TC set condition:               <ul style="list-style-type: none"> <li>▶ TE=0: Transmission Disabled</li> <li>▶ After the last byte of data is transmitted, FE=0 after a specific time period, and the value in the transmit data register is not updated.</li> </ul> </li> </ul> <p>The specific timing for TC set is: 2.5 bit durations after the check bit is transmitted.</p> <ul style="list-style-type: none"> <li>● TC clear condition:               <ul style="list-style-type: none"> <li>▶ When TE=1, write transmit data to the transmit data register.</li> </ul> </li> </ul> <div>  <b>Note</b><br/>           When the TE bit changes from 0 to 1, TC remains at 1.         </div> | RO     |
| 5   | RXNE | Receive Data Register not Empty Flag | <ul style="list-style-type: none"> <li>● 0b0: No data received</li> <li>● 0b1: Ready to read received data</li> </ul> <p>The RXNE bit is set to 1 and cleared to 0 by hardware. The hardware sets RXNE to 1 when the Receive Data is ready to be read, and clears RXNE to 0 after the Receive Data is read.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | RO     |

| Bit | Name            | Function                    | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | Access |
|-----|-----------------|-----------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 4   | BE              | LIN Bus Error-Flag          | <p>Only valid in USART5/USART10:</p> <ul style="list-style-type: none"> <li>● 0b0: No Bus Error.</li> <li>● 0b1: Bus error occurred</li> </ul> <p>BE set condition:</p> <ul style="list-style-type: none"> <li>● During data transmission, a mismatch is detected between the bus data and the transmitted data at the 13/16 position of each bit (16-bit oversampling).</li> </ul> <p>BE clear condition:</p> <ul style="list-style-type: none"> <li>● Write 1 to the CR1.CBE bit.</li> </ul> <div>  <b>Note</b><br/>           TE=0 does not reset the BE bit.<br/><br/>           When BE=1, after the ongoing data transmission is completed, no new data transmission will start. The software must clear the BE flag and restart the transmission action.         </div> | RO     |
| 3   | ORE             | Receive Overflow Error Flag | <ul style="list-style-type: none"> <li>● 0b0: No receive overflow error</li> <li>● 0b1: Receive overflow error occurred</li> </ul> <p>ORE set condition:</p> <ul style="list-style-type: none"> <li>● New data frame received while the receive data register remains unread.</li> </ul> <p>ORE clear condition:</p> <ul style="list-style-type: none"> <li>● Write 1 to the CR1.CORE bit.</li> </ul> <div>  <b>Note</b><br/>           RE=0 does not reset the ORE bit.<br/><br/>           Data received before ORE=1 will be retained, while data received when ORE=1 will be discarded.<br/><br/>           When ORE=1, it cannot continue to receive data, and transmit data is also prohibited in clock synchronization mode.         </div>                           | RO     |
| 2   | <i>Reserved</i> | <i>Reserved</i>             | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | -      |

| Bit | Name | Function                      | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | Access |
|-----|------|-------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 1   | FE   | Receive Frame Error Flag      | <ul style="list-style-type: none"> <li>● 0b0: No receive frame error</li> <li>● 0b1: Receive frame error occurred</li> </ul> <b>UART Mode, Multiprocessor Mode, LIN Mode</b> <ul style="list-style-type: none"> <li>● FE set condition: <ul style="list-style-type: none"> <li>▶ The stop bit of the received data frame is at low level. For two stop bits, only the first stop bit is checked.</li> </ul> </li> <li>● FE clear condition: <ul style="list-style-type: none"> <li>▶ Write 1 to the CR1.CFE bit.</li> </ul> </li> </ul> <div>  <b>Note</b><br/> RE=0 does not reset the FE bit.<br/> When FE=1, the received data is retained but the RI interrupt does not occur. After FE=1, receive data is disabled.<br/> In LIN mode, this flag bit is generated upon receiving the interval segment and does not indicate a frame error; it is also generated when receiving the Synchronization Segment if the baud rate deviation is significant. </div> <b>Smart Card Mode</b> <ul style="list-style-type: none"> <li>● FE set condition: <ul style="list-style-type: none"> <li>▶ Sampled low level error signal flag.</li> </ul> </li> <li>● FE clear condition: <ul style="list-style-type: none"> <li>▶ Write 1 to the CR1.CFE bit.</li> </ul> </li> </ul> <div>  <b>Note</b><br/> RE=0 does not reset the FE bit. </div> | RO     |
| 0   | PE   | Receive Data Check Error Flag | <ul style="list-style-type: none"> <li>● 0b0: No receive data parity error</li> <li>● 0b1: Receive data parity error occurred</li> </ul> <b>PE set condition:</b> <ul style="list-style-type: none"> <li>● A Parity check error occurs when receiving data.</li> </ul> <b>PE clear condition:</b> <ul style="list-style-type: none"> <li>● Write 1 to the CR1.CPE bit.</li> </ul> <div>  <b>Note</b><br/> RE=0 does not reset the PE bit.<br/> When PE=1, the received data is retained but the RI interrupt does not occur. receive data is disabled after PE=1 is set. </div>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | RO     |

### 29.4.3 Transmit Data Register (USART\_TDR)

| Offset | Bit Position |    |    |    |    |    |   |      |     |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|---|------|-----|---|---|---|---|---|---|---|
| 0x04   | 15           | 14 | 13 | 12 | 11 | 10 | 9 | 8    | 7   | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x01FF       |    |    |    |    |    |   |      |     |   |   |   |   |   |   |   |
| Name   | Reserved     |    |    |    |    |    |   | MPID | TDR |   |   |   |   |   |   |   |
| Access |              |    |    |    |    |    |   | RW   | RW  |   |   |   |   |   |   |   |

| Bit   | Name     | Function                   | Description                                                                                                                                                                                                                                                                                                                                                                                                                             | Access |
|-------|----------|----------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 15:10 | Reserved | Reserved                   | -                                                                                                                                                                                                                                                                                                                                                                                                                                       | -      |
| 9     | MPID     | Multiprocessor Mode ID Bit | In Multiprocessor mode, the selection bit for transmitting communication data or transmitting ID: <ul style="list-style-type: none"> <li>0b0: Transmit data</li> <li>0b1: Transmit ID</li> </ul> <div>  <b>Note</b><br/>           The MPID bit is only valid in multiprocessor mode and must be set to the reset value in other modes.         </div> | RW     |
| 8:0   | TDR      | Transmit Data              | Transmit Data Register <div>  <b>Note</b><br/>           The most significant bit TDR[8] is only valid in UART Mode when the data length is set to 9 bits.         </div>                                                                                                                                                                            | RW     |

### 29.4.4 Receive Data Register (USART\_RDR)

| Offset | Bit Position |    |    |    |    |    |   |     |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|---|-----|---|---|---|---|---|---|---|---|
| 0x06   | 15           | 14 | 13 | 12 | 11 | 10 | 9 | 8   | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x0000       |    |    |    |    |    |   |     |   |   |   |   |   |   |   |   |
| Name   | Reserved     |    |    |    |    |    |   | RDR |   |   |   |   |   |   |   |   |
| Access |              |    |    |    |    |    |   | RO  |   |   |   |   |   |   |   |   |

| Bit  | Name     | Function       | Description                                                                                                                                                                                                                                                 | Access |
|------|----------|----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 15:9 | Reserved | Reserved       | -                                                                                                                                                                                                                                                           | -      |
| 8:0  | RDR      | Receiving Data | Receive Data Register <div>  <b>Note</b><br/>           The most significant bit RDR[8] is only valid in UART Mode when the data length is set to 9 bits.         </div> | RO     |

## 29.4.5 Baud Rate Register (USART\_BRR)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |             |    |    |    |          |              |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-------------|----|----|----|----------|--------------|---|---|---|---|---|---|---|---|---|---|
| 0x08   | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15          | 14 | 13 | 12 | 11       | 10           | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x0000FFFF   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |             |    |    |    |          |              |   |   |   |   |   |   |   |   |   |   |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | DIV_Integer |    |    |    | Reserved | DIV_Fraction |   |   |   |   |   |   |   |   |   |   |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RW          |    |    |    |          | RW           |   |   |   |   |   |   |   |   |   |   |

| Bit   | Name         | Function                      | Description                                                                                                                                                                                                                                                           | Access |
|-------|--------------|-------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:16 | Reserved     | Reserved                      | -                                                                                                                                                                                                                                                                     | -      |
| 15:8  | DIV_Integer  | Integer frequency division    | Integer frequency divider register<br><br> <b>Note</b><br>This bit can only be set when TE=0 & RE=0 (transmit/receive disabled).                                                     | RW     |
| 7     | Reserved     | Reserved                      | Reads as 1, Write as 1                                                                                                                                                                                                                                                | -      |
| 6:0   | DIV_Fraction | Fractional frequency division | Fractional frequency divider register<br><br> <b>Note</b><br>This bit can only be set when TE=0 & RE=0 (transmit/receive disabled), and the setting takes effect only when FBME=1. | RW     |

Table 29-21 Baud Rate Calculation Formula (Fractional Baud Rate is disable where FBME=0)

| Mode                               | Baud Rate Calculation Formula                                                | Error E calculation formula                                                                                             |
|------------------------------------|------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------|
| UART Mode/<br>Multi-processor Mode | $B = \frac{C}{8 \times (2 - \text{OVER8}) \times (\text{DIV\_Integer} + 1)}$ | $E = \left\{ \frac{C}{8 \times (2 - \text{OVER8}) \times (\text{DIV\_Integer} + 1) \times B} - 1 \right\} \times 100\%$ |
| Clock Synchronization Mode         | $B = \frac{C}{4 \times (\text{DIV\_Integer} + 1)}$                           | -                                                                                                                       |
| Smart Card Mode                    | $B = \frac{C}{2 \times \text{BCN} \times (\text{DIV\_Integer} + 1)}$         | $E = \left\{ \frac{C}{2 \times \text{BCN} \times (\text{DIV\_Integer} + 1) \times B} - 1 \right\} \times 100\%$         |

Table 29-22 Baud Rate Calculation Formula (Fractional Baud Rate is enable where FBME=1)

| Mode                               | Baud Rate Calculation Formula                                                                                               | Calculation formula for error E (%)                                                                                                                                    |
|------------------------------------|-----------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| UART Mode/<br>Multi-processor Mode | $B = \frac{C \times (128 + \text{DIV\_Fraction})}{8 \times (2 - \text{OVER8}) \times (\text{DIV\_Integer} + 1) \times 256}$ | $E = \left\{ \frac{C \times (128 + \text{DIV\_Fraction})}{8 \times (2 - \text{OVER8}) \times (\text{DIV\_Integer} + 1) \times 256 \times B} - 1 \right\} \times 100\%$ |



| Mode                       | Baud Rate Calculation Formula                                                                                       | Calculation formula for error E (%)                                                                                                                            |
|----------------------------|---------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Clock Synchronization Mode | $B = \frac{C \times (128 + \text{DIV\_Fraction})}{4 \times (\text{DIV\_Integer} + 1) \times 256}$                   | -                                                                                                                                                              |
| Smart Card Mode            | $B = \frac{C \times (128 + \text{DIV\_Fraction})}{2 \times \text{BCN} \times (\text{DIV\_Integer} + 1) \times 256}$ | $E = \left\{ \frac{C \times (128 + \text{DIV\_Fraction})}{2 \times \text{BCN} \times (\text{DIV\_Integer} + 1) \times 256 \times B} - 1 \right\} \times 100\%$ |

Where:

- B: Baud rate (Mbps)
- C: Clock (MHz) set by PR.ULBREN and PR.PSC bits
- BCN: CR3.BCN register setting value

## 29.4.6 Control Register 1 (USART\_CR1)

| Offset | Bit Position |     |      |    |          |       |    |      |       |     |       |      |          |     |     |       |          |    |          |     |    |        |       |      |     |      |    |    |       |      |    |    |
|--------|--------------|-----|------|----|----------|-------|----|------|-------|-----|-------|------|----------|-----|-----|-------|----------|----|----------|-----|----|--------|-------|------|-----|------|----|----|-------|------|----|----|
| 0x0C   | 31           | 30  | 29   | 28 | 27       | 26    | 25 | 24   | 23    | 22  | 21    | 20   | 19       | 18  | 17  | 16    | 15       | 14 | 13       | 12  | 11 | 10     | 9     | 8    | 7   | 6    | 5  | 4  | 3     | 2    | 1  | 0  |
| Reset  | 0x80000000   |     |      |    |          |       |    |      |       |     |       |      |          |     |     |       |          |    |          |     |    |        |       |      |     |      |    |    |       |      |    |    |
| Name   | SBS          | NFE | FBME | ML | Reserved | CTEND | MS | CLBD | CWKUP | CBE | CRTOF | CORE | Reserved | CFE | CPE | OVER8 | Reserved | M  | Reserved | PCE | PS | TENDIE | TXEIE | TCIE | RIE | SLME | TE | RE | RTOIE | RTOE |    |    |
| Access | RW           | RW  | RW   | RW |          | RW    | RW | RW   | RW    | RW  | RW    | RW   |          | RW  | RW  | RW    |          | RW |          | RW  | RW | RW     | RW    | RW   | RW  | RW   | RW | RW | RW    | RW   | RW | RW |

| Bit | Name | Function                                                    | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | Access |
|-----|------|-------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31  | SBS  | UART Mode Receive Data Start Bit Detection Mode Setting Bit | <p>In UART Mode, the start bit detection setting for receive data:</p> <ul style="list-style-type: none"> <li>0b0: Start bit detection method is low level on RX pin</li> <li>0b1: Start bit detection mode is falling edge on RX pin</li> </ul> <div>  <b>Note</b><br/>           This bit must retain the reset value in non-USART Mode.<br/><br/>           This bit can only be set when TE=0 &amp; RE=0 (transmit/receive disabled).         </div> | RW     |
| 30  | NFE  | Digital Filtering Enable Bit                                | <ul style="list-style-type: none"> <li>0b0: Disable</li> <li>0b1: Enable</li> </ul> <div>  <b>Note</b><br/>           This bit must retain the reset value in Clock Synchronization Mode and synchronization frame detection of LIN Mode.<br/><br/>           This bit can only be set when TE=0 &amp; RE=0 (transmit/receive disabled).         </div>                                                                                                | RW     |
| 29  | FBME | Fractional Baud Rate Function Enabled                       | <ul style="list-style-type: none"> <li>0b0: Disable</li> <li>0b1: Enable</li> </ul> <div>  <b>Note</b><br/>           This bit can only be set when TE=0 &amp; RE=0 (transmit/receive disabled).         </div>                                                                                                                                                                                                                                        | RW     |
| 28  | ML   | MSB/LSB Selection Bit                                       | <p>In UART Mode/Clock Synchronization Mode/Smart Card Mode, MSB/LSB selection bit:</p> <ul style="list-style-type: none"> <li>0b0: LSB mode</li> <li>0b1: MSB mode</li> </ul> <div>  <b>Note</b><br/>           This bit can only be set when TE=0 &amp; RE=0 (transmit/receive disabled).         </div>                                                                                                                                              | RW     |

| Bit   | Name            | Function                         | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | Access |
|-------|-----------------|----------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 27:26 | <i>Reserved</i> | <i>Reserved</i>                  | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | -      |
| 25    | CTEND           | TEND Clear Bit                   | Write 1 to clear TEND flag, write 0 has no effect; read always returns 0.                                                                                                                                                                                                                                                                                                                                                                                                                                               | RW     |
| 24    | MS              | Communication Mode Selection Bit | <ul style="list-style-type: none"> <li>● 0b0: UART Mode</li> <li>● 0b1: Clock synchronization mode</li> </ul> <div>  <b>Note</b><br/>           This bit can only be set when TE=0 &amp; RE=0 (transmit/receive disabled) . In Smart Card Mode, MS must be written to the reset value.         </div>                                                                                                                                  | RW     |
| 23    | CLBD            | LBD Clear Bit                    | Only valid in USART5/USART10:<br>Write 1 to clear LBD flag, write 0 has no effect; read always returns 0.                                                                                                                                                                                                                                                                                                                                                                                                               | RW     |
| 22    | CWKUP           | WKUP Clear Bit                   | Only valid in USART5/USART10:<br>Write 1 to clear WKUP flag, write 0 has no effect; read always returns 0.                                                                                                                                                                                                                                                                                                                                                                                                              | RW     |
| 21    | CBE             | BE Clear Bit                     | Only valid in USART5/USART10:<br>Write 1 to clear BE flag, write 0 has no effect; read always returns 0.                                                                                                                                                                                                                                                                                                                                                                                                                | RW     |
| 20    | CRTOF           | RTOF Clear Bit                   | Write 1 to clear RTOF flag, write 0 has no effect; read always returns 0.                                                                                                                                                                                                                                                                                                                                                                                                                                               | RW     |
| 19    | CORE            | ORE Flag Clear Bit               | Write 1 to clear ORE flag, write 0 has no effect; read always returns 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                | RW     |
| 18    | <i>Reserved</i> | <i>Reserved</i>                  | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | -      |
| 17    | CFE             | FE Flag Clear Bit                | Write 1 to clear FE flag, write 0 has no effect; read always returns 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                 | RW     |
| 16    | CPE             | PE Flag Clear Bit                | Write 1 to clear PE flag, write 0 has no effect; read always returns 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                 | RW     |
| 15    | OVER8           | UART Oversampling Mode           | UART oversampling mode setting, i.e., the number of base Clock periods during one Bit data transmission: <ul style="list-style-type: none"> <li>● 0b0: 16 bits</li> <li>● 0b1: 8 bits</li> </ul> <div>  <b>Note</b><br/>           This bit must retain the reset value in Smart Card Mode/Clock Synchronization Mode.<br/><br/>           This bit can only be set when TE=0 &amp; RE=0 (transmit/receive disabled).         </div> | RW     |
| 14:13 | <i>Reserved</i> | <i>Reserved</i>                  | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | -      |

| Bit | Name            | Function                                          | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | Access |
|-----|-----------------|---------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 12  | M               | Data Length Setting Bit                           | <p>In UART Mode, transmit/receive data length setting bit:</p> <ul style="list-style-type: none"> <li>● 0b0: 8 bits</li> <li>● 0b1: 9 bits</li> </ul> <div>  <b>Note</b><br/>           This bit must retain the reset value in LIN Mode/Smart Card Mode/Clock Synchronization Mode.<br/>           This bit can only be set when TE=0 &amp; RE=0 (transmit/receive disabled).         </div>                                                                         | RW     |
| 11  | <i>Reserved</i> | <i>Reserved</i>                                   | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | -      |
| 10  | PCE             | Check Enable Bit                                  | <p>In UART Mode, the parity check enable bit:</p> <ul style="list-style-type: none"> <li>● 0b0: No check</li> <li>● 0b1: Check</li> </ul> <div>  <b>Note</b><br/>           This bit must retain the reset value in LIN Mode/Multiprocessor Communication Mode/Clock Synchronization Mode, and must be set to 1 in Smart Card Mode.<br/>           This bit can only be set when TE=0 &amp; RE=0 (transmit/receive disabled).         </div>                          | RW     |
| 9   | PS              | Check Bit                                         | <p>In UART Mode, the parity check selection bit:</p> <ul style="list-style-type: none"> <li>● 0b0: Even parity</li> <li>● 0b1: Odd parity</li> </ul> <div>  <b>Note</b><br/>           This bit is valid only when PCE=1.<br/>           This bit must retain 0 in Smart Card Mode.<br/>           This bit can only be set when TE=0 &amp; RE=0 (transmit/receive disabled).         </div>                                                                        | RW     |
| 8   | TENDIE          | Transmit Completion Interrupt Enable Bit          | <ul style="list-style-type: none"> <li>● 0b0: TENDI interrupt request is disabled, no TENDI interrupt occurs</li> <li>● 0b1: TENDI interrupt request is enabled, TENDI interrupt occurred</li> </ul>                                                                                                                                                                                                                                                                                                                                                   | RW     |
| 7   | TXEIE           | Transmit data Register Empty Interrupt Enable Bit | <ul style="list-style-type: none"> <li>● 0b0: TI interrupt request is disabled, no TI interrupt occurs</li> <li>● 0b1: TI interrupt request is enabled, TI interrupt occurred</li> </ul> <div>  <b>Note</b><br/>           When TE=0 and TXEIE=1 is written, it is necessary to wait for the TI interrupt to occur when TE=1.<br/>           When TE=1 and writing TXEIE=1, it is necessary to wait until SR.TC=1 before the write can be performed.         </div> | RW     |

| Bit | Name  | Function                                      | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | Access |
|-----|-------|-----------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 6   | TCIE  | Transmit Completion Interrupt Enable Bit      | <ul style="list-style-type: none"> <li>0b0: TCI interrupt request is disabled, no TCI interrupt occurs</li> <li>0b1: TCI interrupt request enable, TCI interrupt occurred</li> </ul> <div>  <b>Note</b><br/> Read is valid in non-Smart Card Mode, reads as 0 in Smart Card Mode. </div>                                                                                                                                                                                                                                                                                                                                                | RW     |
| 5   | RIE   | Receive Interrupt, Error Interrupt Enable Bit | <ul style="list-style-type: none"> <li>0b0: Receive interrupt and error Interrupt requests are disabled; no RI and EI Interrupts occur</li> <li>0b1: Receive interrupt and error interrupt requests are enabled, RI and EI interrupts occurred</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                | RW     |
| 4   | SLME  | Silent Mode Enable Bit                        | <p>In multiprocessor operations, the silent mode enable bit:</p> <ul style="list-style-type: none"> <li>0b0: Normal mode</li> <li>0b1: Silent Mode</li> </ul> <p>When SLME=1, communication data with MPB bit 0 will not be read from the Receive Shift Register to the Receive Data Register, and the error flags ORE and FE bits will not be set. When an ID data with MPB=1 is received, the SLME is automatically cleared and starts the normal data reception Action.</p> <div>  <b>Note</b><br/> This bit is only valid in UART Multiprocessor Communication Mode. In other modes, this bit must retain the reset value. </div> | RW     |
| 3   | TE    | Transmitter Enable Bit                        | <ul style="list-style-type: none"> <li>0b0: Transmitter disable</li> <li>0b1: Transmitter enable</li> </ul> <div>  <b>Note</b><br/> In Clock Synchronization Mode, this bit can only be written as 1 when TE=0 &amp; RE=0 (transmit/receive disabled). </div>                                                                                                                                                                                                                                                                                                                                                                         | RW     |
| 2   | RE    | Receiver Enable bit                           | <ul style="list-style-type: none"> <li>0b0: Receiver disable</li> <li>0b1: Receiver enable</li> </ul> <div>  <b>Note</b><br/> In Clock Synchronization Mode, this bit can only be written as 1 when TE=0 &amp; RE=0 (transmit/receive disabled). </div>                                                                                                                                                                                                                                                                                                                                                                               | RW     |
| 1   | RTOIE | UART TIMEOUT Interrupt Enable Bit             | <ul style="list-style-type: none"> <li>0b0: UART TIMEOUT interrupt request is disabled, no RTOI interrupt occurs</li> <li>0b1: UART TIMEOUT interrupt request is enabled, RTOI interrupt occurred</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | RW     |

| Bit | Name | Function                            | Description                                                                                                                      | Access |
|-----|------|-------------------------------------|----------------------------------------------------------------------------------------------------------------------------------|--------|
| 0   | RTOE | UART TIMEOUT<br>Function Enable Bit | <ul style="list-style-type: none"><li>● 0b0: UART TIMEOUT function disable</li><li>● 0b1: UART TIMEOUT function enable</li></ul> | RW     |

## 29.4.7 Control Register 2 (USART\_CR2)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |      |     |          |       |      |      |          |       |      |      |       |     |     |        |     |    |    |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|------|-----|----------|-------|------|------|----------|-------|------|------|-------|-----|-----|--------|-----|----|----|---|
| 0x10   | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17   | 16  | 15       | 14    | 13   | 12   | 11       | 10    | 9    | 8    | 7     | 6   | 5   | 4      | 3   | 2  | 1  | 0 |
| Reset  | 0x00000600   |    |    |    |    |    |    |    |    |    |    |    |    |    |      |     |          |       |      |      |          |       |      |      |       |     |     |        |     |    |    |   |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    | SBKM | SBK | Reserved | LINEN | STOP | CLKC | Reserved | WKUPE | SBKL | LBDL | LBDIE | BEE | BIE | WKUPIE | MPE |    |    |   |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    | RW   | RW  |          | RW    | RW   | RW   |          | RW    | RW   | RW   | RW    | RW  | RW  | RW     | RW  | RW | RW |   |

| Bit   | Name     | Function                                                                                                                                 | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | Access |      |                               |     |     |                              |     |     |                              |     |     |                                                                                                                                          |     |     |                                                                                                                                       |    |
|-------|----------|------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|------|-------------------------------|-----|-----|------------------------------|-----|-----|------------------------------|-----|-----|------------------------------------------------------------------------------------------------------------------------------------------|-----|-----|---------------------------------------------------------------------------------------------------------------------------------------|----|
| 31:18 | Reserved | Reserved                                                                                                                                 | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | -      |      |                               |     |     |                              |     |     |                              |     |     |                                                                                                                                          |     |     |                                                                                                                                       |    |
| 17    | SBKM     | Interval Segment Transmission Mode                                                                                                       | <div>Only valid in USART5/USART10:<br/>For specific settings, refer to the SBK bit description.</div> <div> <b>Note</b><br/>This bit is enabled only when LINEN=1.</div>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | RW     |      |                               |     |     |                              |     |     |                              |     |     |                                                                                                                                          |     |     |                                                                                                                                       |    |
| 16    | SBK      | Interval Segment Transmission Enable Bit                                                                                                 | <div>Only valid in USART5/USART10</div> <table><tr><th>SBK</th><th>SBKM</th><th>Interval segment transmission</th></tr><tr><td>0b0</td><td>0b0</td><td>Do not send interval segment</td></tr><tr><td>0b0</td><td>0b1</td><td>Do not send interval segment</td></tr><tr><td>0b1</td><td>0b0</td><td>When SBK is written as 1, the interval segment is automatically transmitted. After transmission completes, SBK is automatically cleared.</td></tr><tr><td>0b1</td><td>0b1</td><td>Write data 0x00 to USART_TDR.TDR to start sending the interval segment. After transmission is complete, SBK is automatically cleared.</td></tr></table> <div> <b>Note</b><br/>This bit is valid only when LINEN=1.</div> | SBK    | SBKM | Interval segment transmission | 0b0 | 0b0 | Do not send interval segment | 0b0 | 0b1 | Do not send interval segment | 0b1 | 0b0 | When SBK is written as 1, the interval segment is automatically transmitted. After transmission completes, SBK is automatically cleared. | 0b1 | 0b1 | Write data 0x00 to USART_TDR.TDR to start sending the interval segment. After transmission is complete, SBK is automatically cleared. | RW |
| SBK   | SBKM     | Interval segment transmission                                                                                                            |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |        |      |                               |     |     |                              |     |     |                              |     |     |                                                                                                                                          |     |     |                                                                                                                                       |    |
| 0b0   | 0b0      | Do not send interval segment                                                                                                             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |        |      |                               |     |     |                              |     |     |                              |     |     |                                                                                                                                          |     |     |                                                                                                                                       |    |
| 0b0   | 0b1      | Do not send interval segment                                                                                                             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |        |      |                               |     |     |                              |     |     |                              |     |     |                                                                                                                                          |     |     |                                                                                                                                       |    |
| 0b1   | 0b0      | When SBK is written as 1, the interval segment is automatically transmitted. After transmission completes, SBK is automatically cleared. |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |        |      |                               |     |     |                              |     |     |                              |     |     |                                                                                                                                          |     |     |                                                                                                                                       |    |
| 0b1   | 0b1      | Write data 0x00 to USART_TDR.TDR to start sending the interval segment. After transmission is complete, SBK is automatically cleared.    |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |        |      |                               |     |     |                              |     |     |                              |     |     |                                                                                                                                          |     |     |                                                                                                                                       |    |
| 15    | Reserved | Reserved                                                                                                                                 | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | -      |      |                               |     |     |                              |     |     |                              |     |     |                                                                                                                                          |     |     |                                                                                                                                       |    |
| 14    | LINEN    | LIN function Enabled                                                                                                                     | <div>Only valid in USART5/USART10:<ul style="list-style-type: none"><li>0b0: Disable LIN Function</li><li>0b1: Enable LIN function</li></ul></div> <div> <b>Note</b><br/>This bit can only be set when TE=0 &amp; RE=0 (transmit/receive disabled).</div>                                                                                                                                                                                                                                                                                                                                                                                                                                                     | RW     |      |                               |     |     |                              |     |     |                              |     |     |                                                                                                                                          |     |     |                                                                                                                                       |    |

| Bit   | Name     | Function             | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | Access |
|-------|----------|----------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 13    | STOP     | Stop Bit Setting Bit | <p>In UART Mode, stop bit length setting bit:</p> <ul style="list-style-type: none"> <li>● 0b0: 1 stop bit</li> <li>● 0b1: 2 stop bits</li> </ul> <div>  <b>Note</b><br/>           This bit must retain the reset value in LIN Mode/ Smart Card Mode/Clock Synchronization Mode.<br/><br/>           This bit can only be set when TE=0 &amp; RE=0 (transmit/receive disabled).         </div>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | RW     |
| 12:11 | CLKC     | Clock Control Bit    | <p><b>UART Mode</b></p> <ul style="list-style-type: none"> <li>● 0b00: Clock source is the clock generated by the internal baud rate generator and doesnot output to the USARTn_CK pin. The USARTn_CK pin can be used as a general-purpose IO.</li> <li>● 0b01: Clock source is the clock generated by the internal baud rate generator and output to the USARTn_CK pin. The output clock frequency equals the baud rate.</li> <li>● 0b10/0b11: Clock source is external input clock, the input clock frequency is 16 times the baud rate (OVER8=0) or 8 times the baud rate (OVER8=1).</li> </ul> <p><b>Clock Synchronization Mode</b></p> <ul style="list-style-type: none"> <li>● 0b00/0b01: Clock source is the clock generated by the internal baud rate generator and output to the USARTn_CK pin.</li> <li>● 0b10/0b11: Clock source is external input clock, the input clock frequency equals baud rate.</li> </ul> <p><b>Smart Card Mode</b></p> <ul style="list-style-type: none"> <li>● 0b00: Clock source is the clock generated by the internal baud rate generator and doesnot output to the USARTn_CK pin. The USARTn_CK pin can be used as a general-purpose IO.</li> <li>● 0b01: Clock source is the clock generated by the internal baud rate generator and output to the USARTn_CK pin.</li> <li>● 0b10/0b11: Setting prohibited</li> </ul> <div>  <b>Note</b><br/>           This bit can only be set when TE=0 &amp; RE=0 (transmit/receive disabled).         </div> | RW     |
| 10:9  | Reserved | Reserved             | All bits read as 1, Write as 1                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | -      |



| Bit | Name  | Function                                                     | Description                                                                                                                                                                                                                                                                                                                                                                                                                                              | Access |
|-----|-------|--------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 8   | WKUPE | LIN Wake-up Signal Detection Enable Bit                      | <p>Only valid in USART5/USART10:</p> <ul style="list-style-type: none"> <li>● 0b0: Disable wake-up signal detection</li> <li>● 0b1: Enable wake-up signal detection</li> </ul> <p> <b>Note</b><br/>Set WKUPE to 1 only when the system requires wake-up signal detection. After detecting the wake-up signal, software must clear WKUPE.</p>                            | RW     |
| 7:6 | SBKL  | LIN Interval Low Level Width Setting Bit                     | <p>Only valid in USART5/USART10:</p> <ul style="list-style-type: none"> <li>● 0b00: 10 bits</li> <li>● 0b01: 11 bits</li> <li>● 0b10: 13 bits</li> <li>● 0b11: 14 bits</li> </ul> <p> <b>Note</b><br/>This bit must retain the reset value in non-UART-LIN Mode.<br/><br/>This bit can only be set when TE=0 &amp; RE=0 (transmit/receive disabled).</p>                | RW     |
| 5   | LBDL  | LIN Interval Low Level Width Detection Threshold Setting Bit | <p>Only valid in USART5/USART10:</p> <ul style="list-style-type: none"> <li>● 0b0: &gt;= 10 bits</li> <li>● 0b1: &gt;= 11 bits</li> </ul> <p> <b>Note</b><br/>This bit must retain the reset value in non-UART-LIN Mode.<br/><br/>This bit can only be set when TE=0 &amp; RE=0 (transmit/receive disabled).</p>                                                      | RW     |
| 4   | LBDIE | LIN Interval Detection Interrupt Enable Bit                  | <p>Only valid in USART5/USART10:</p> <ul style="list-style-type: none"> <li>● 0b0: LIN interval segment detection interrupt request is disable, no interrupt occurs.</li> <li>● 0b1: LIN interval segment detection interrupt request is enable, an interrupt occurred.</li> </ul> <p> <b>Note</b><br/>This bit must retain the reset value in non-UART-LIN Mode.</p> | RW     |

| Bit | Name   | Function                                          | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | Access |
|-----|--------|---------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 3   | BEE    | LIN Bus Error Detection Function Enable Bit       | <p>Only valid in USART5/USART10:</p> <ul style="list-style-type: none"> <li>● 0b0: Disable LIN bus error function detection function</li> <li>● 0b1: Enable LIN bus error function detection function</li> </ul> <div>  <b>Note</b><br/>           This bit must retain the reset value in non-UART-LIN Mode.<br/><br/>           This bit can only be set when TE=0 &amp; RE=0 (transmit/receive disabled).         </div> | RW     |
| 2   | BEIE   | LIN Bus Error Interrupt Enable Bit                | <p>Only valid in USART5/USART10:</p> <ul style="list-style-type: none"> <li>● 0b0: LIN Bus error interrupt request is disabled, no interrupt occurs.</li> <li>● 0b1: LIN bus error interrupt request is enabled, an interrupt occurred.</li> </ul> <div>  <b>Note</b><br/>           This bit must retain the reset value in non-UART-LIN Mode.         </div>                                                              | RW     |
| 1   | WKUPIE | LIN Wake-up Signal Detection Interrupt Enable Bit | <p>Only valid in USART5/USART10:</p> <ul style="list-style-type: none"> <li>● 0b0: LIN Wake-up signal detection interrupt request is disabled, no interrupt occurs.</li> <li>● 0b1: LIN Wake-up signal detection interrupt request is enabled, an interrupt has occurred.</li> </ul> <div>  <b>Note</b><br/>           This bit must retain the reset value in non-UART-LIN Mode.         </div>                          | RW     |
| 0   | MPE    | Multiprocessor Function Enable Bit                | <p>In UART Mode, the multiprocessor function enable bit:</p> <ul style="list-style-type: none"> <li>● 0b0: Disable</li> <li>● 0b1: Enable</li> </ul> <div>  <b>Note</b><br/>           This bit must retain the reset value in non-UART Mode.<br/><br/>           This bit can only be set when TE=0 &amp; RE=0 (transmit/receive disabled).         </div>                                                               | RW     |

### 29.4.8 Control Register 3 (USART\_CR3)

| Offset | Bit Position |     |          |     |    |    |    |          |     |    |    |    |          |    |    |    |     |    |    |    |      |    |      |          |      |   |      |       |          |   |   |   |
|--------|--------------|-----|----------|-----|----|----|----|----------|-----|----|----|----|----------|----|----|----|-----|----|----|----|------|----|------|----------|------|---|------|-------|----------|---|---|---|
| 0x14   | 31           | 30  | 29       | 28  | 27 | 26 | 25 | 24       | 23  | 22 | 21 | 20 | 19       | 18 | 17 | 16 | 15  | 14 | 13 | 12 | 11   | 10 | 9    | 8        | 7    | 6 | 5    | 4     | 3        | 2 | 1 | 0 |
| Reset  | 0x00000000   |     |          |     |    |    |    |          |     |    |    |    |          |    |    |    |     |    |    |    |      |    |      |          |      |   |      |       |          |   |   |   |
| Name   | DEP          | DEM | Reserved | DEA |    |    |    | Reserved | BCN |    |    |    | Reserved |    |    |    | DED |    |    |    | CTSE |    | RTSE | Reserved | SCEN |   | LOOP | HDSEL | Reserved |   |   |   |
| Access | RW           | RW  |          | RW  |    |    |    |          | RW  |    |    |    |          |    |    |    | RW  |    |    |    | RW   |    | RW   |          | RW   |   | RW   | RW    |          |   |   |   |

| Bit   | Name            | Function                 | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | Access |
|-------|-----------------|--------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31    | DEP             | DE Function Polarity     | Driver enable polarity mode:<br><ul style="list-style-type: none"> <li>0b0: DE signal is high when USART is in IDLE state</li> <li>0b1: DE signal is low when USART is in IDLE state</li> </ul> When USART TE or RE is enabled, this bit field cannot be written.                                                                                                                                                                                                                                                                                              | RW     |
| 30    | DEM             | DE Function Enabled      | Enable driver enable mode. This bit is used to activate external transceiver control via the DE signal.<br><ul style="list-style-type: none"> <li>0b0: DE function disable</li> <li>0b1: DE function enable</li> </ul> When USART TE or RE is enabled, writing to this bit field is prohibited.                                                                                                                                                                                                                                                                | RW     |
| 29    | <i>Reserved</i> | <i>Reserved</i>          | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | -      |
| 28:25 | DEA             | DE Assertion Enable Time | These bits define the time between DE (Driver Enable) signal activation and the Start bit, expressed in sample time units (1/8 or 1/16 bit time), configured by the OVER8 bit: maximum configurable value is 0x7 when OVER8 is set to 1, and 0xF when OVER8 is set to 0.<br>Minimum configuration is 0. When configured as 0, the DE signal occurs simultaneously with the Start Bit; when configured as 0x7 or 0xF, the DE signal occurs 7 or 15 time units before the Start Bit.<br>When USART TE or RE is enabled, writing to this bit field is prohibited. | RW     |
| 24    | <i>Reserved</i> | <i>Reserved</i>          | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | -      |
| 23:21 | BCN             | Basic Clock Count        | In Smart Card Mode, set the number of basic clock periods for 1 Bit data transmission:<br><ul style="list-style-type: none"> <li>0b000: 32</li> <li>0b001: 64</li> <li>0b010: Setting prohibited</li> <li>0b011: 128</li> <li>0b100: Setting prohibited</li> <li>0b101: 256</li> </ul>                                                                                                                                                                                                                                                                         | RW     |

| Bit   | Name     | Function                       | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | Access |
|-------|----------|--------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 23:21 | BCN      | Basic Clock Count              | <ul style="list-style-type: none"> <li>● 0b110: 372</li> <li>● 0b111: Setting prohibited</li> </ul> <div>  <b>Note</b><br/>           This bit must retain the reset value in non-Smart Card Mode.<br/><br/>           This bit can only be set when TE=0 &amp; RE=0 (transmit/receive disabled).<br/><br/>           USART5/USART10 does not support this function. Read always returns 0.         </div>                                                                                                                                                                                  | RW     |
| 20:14 | Reserved | Reserved                       | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | -      |
| 13:10 | DED      | DE Assertion Cancellation Time | <p>These bits define the interval between the end of the last stop bit in the transmitted message and the deactivation of the DE (Driver Enable) signal. It is expressed in sample time units (1/8 or 1/16 bit time), configured by the OVER8 bit: maximum configurable value is 0x7 when OVER8 is set to 1, and 0xF when OVER8 is set to 0.</p> <p>Minimum configurable value is 0. When configured as 0, the DE signal transition occurs 2 time units after the end bit completes; when configured as 0x7 or 0xF, the DE signal occurs 9 or 17 time units after the end bit completes.</p> <p>When USART TE or RE is enabled, writing to this bit field is prohibited.</p> | RW     |
| 9     | CTSE     | CTS Function Enable Bit        | <ul style="list-style-type: none"> <li>● 0b0: CTS Function Disable</li> <li>● 0b1: CTS Function Enable</li> </ul> <div>  <b>Note</b><br/>           This bit can only be set when TE=0 &amp; RE=0 (transmit/receive disabled).         </div>                                                                                                                                                                                                                                                                                                                                             | RW     |
| 8     | RTSE     | RTS Function Enable Bit        | <ul style="list-style-type: none"> <li>● 0b0: RTS Function Disable</li> <li>● 0b1: RTS Function Enable</li> </ul> <div>  <b>Note</b><br/>           This bit can only be set when TE=0 &amp; RE=0 (transmit/receive disabled).         </div>                                                                                                                                                                                                                                                                                                                                             | RW     |
| 7:6   | Reserved | Reserved                       | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | -      |

| Bit | Name     | Function                                     | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | Access |
|-----|----------|----------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 5   | SCEN     | Smart Card Mode Enable Bit                   | <ul style="list-style-type: none"> <li>0b0: Disable Smart Card Mode</li> <li>0b1: Enable Smart Card Mode</li> </ul> <div>  <b>Note</b><br/>           This bit must retain the reset value in non-Smart Card Mode.<br/>           This bit can only be set when TE=0 &amp; RE=0 (transmit/receive disabled).<br/>           USART5/USART10 does not support this function. Read always returns 0.         </div> | RW     |
| 4   | LOOP     | LIN Loop Mode Enable Bit                     | Only valid in USART5/USART10: <ul style="list-style-type: none"> <li>0b0: Normal Mode</li> <li>0b1: Loop Mode</li> </ul> <div>  <b>Note</b><br/>           This bit can only be set when TE=0 &amp; RE=0 (transmit/receive disabled).         </div>                                                                                                                                                             | RW     |
| 3   | HDSEL    | UART Single-wire Half-duplex Mode Enable Bit | <ul style="list-style-type: none"> <li>0b0: UART full-duplex mode</li> <li>0b1: UART half-duplex mode</li> </ul> <div>  <b>Note</b><br/>           This bit can only be set when TE=0 &amp; RE=0 (transmit/receive disabled).         </div>                                                                                                                                                                    | RW     |
| 2:0 | Reserved | Reserved                                     | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | -      |

## 29.4.9 Prescaler Register (USART\_PR)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |        |        |     |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|--------|--------|-----|
| 0x18   | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |        |        |     |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |        |        |     |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   | ULBREN | LBMPPC | PSC |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   | RW     | RW     | RW  |

| Bit  | Name     | Function                                                 | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | Access |
|------|----------|----------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:5 | Reserved | Reserved                                                 | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | -      |
| 4    | ULBREN   | Ultra-low baud rate enable bit                           | <p>When using the internal clock source, the ultra-low Baud rate function enable bit:</p> <ul style="list-style-type: none"> <li>● 0b0: Ultra-low baud rate function disable</li> <li>● 0b1: Ultra-low baud rate function enable</li> </ul> <p>And the prescaler division value is determined by PSC as follows:</p> <ul style="list-style-type: none"> <li>● 0b00: PCLK1/128</li> <li>● 0b01: PCLK1/256</li> <li>● 0b10: PCLK1/512</li> <li>● 0b11: PCLK1/1024</li> </ul> <p> <b>Note</b><br/>This bit can only be set when TE=0 &amp; RE=0 (transmit/receive disabled).</p> | RW     |
| 3:2  | LBMPPSC  | LIN Baud rate measurement counter clock source selection | <p>Only valid in USART5/USART10:</p> <ul style="list-style-type: none"> <li>● 0b00: PCLK1</li> <li>● 0b01: PCLK1/2</li> <li>● 0b10: PCLK1/4</li> <li>● 0b11: PCLK1/8</li> </ul> <p> <b>Note</b><br/>This bit can only be set when TE=0 &amp; RE=0 (transmit/receive disabled).</p>                                                                                                                                                                                                                                                                                            | RW     |
| 1:0  | PSC      | Prescaler Value                                          | <p>When using the internal clock source, the prescaler division ratio selection bit:</p> <ul style="list-style-type: none"> <li>● 0b00: PCLK1</li> <li>● 0b01: PCLK1/4</li> <li>● 0b10: PCLK1/16</li> <li>● 0b11: PCLK1/64</li> </ul> <p> <b>Note</b><br/>This bit can only be set when TE=0 &amp; RE=0 (transmit/receive disabled).</p>                                                                                                                                                                                                                                      | RW     |

## 29.4.10 LIN Baud Rate Measurement Count Register (USART\_LBMC)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |      |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0x1C   | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15   | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |      |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | LBMC |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |      |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit   | Name     | Function                                 | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | Access |
|-------|----------|------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:16 | Reserved | Reserved                                 | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | -      |
| 15:0  | LBMC     | LIN Baud Rate Measurement Result Counter | <p>Only valid in USART5/USART10:<br/>LIN slave node uses the clock source selected by the USARTn_PR.LBMPSC register to measure the frequency of the synchronization segment transmitted by the master node.<br/>The measured LIN communication baud rate is obtained by dividing the count clock frequency selected by USARTn_PR.LBMPSC by the value of LBMC.</p> <div>  <b>Note</b><br/>           The value of LBMC is meaningful only after being read upon completion of the synchronization segment.         </div> | RO     |

## 29.4.11 USART1 Filter Control Register (USART1\_NFC)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |            |   |            |  |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|------------|---|------------|--|
| 0x1C   | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1          | 0 |            |  |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |            |   |            |  |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   | USART1_NFE |   | USART1_NFS |  |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   | RW         |   | RW         |  |

| Bit  | Name       | Function                              | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | Access |
|------|------------|---------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:3 | Reserved   | Reserved                              | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | -      |
| 2    | USART1_NFE | USART1 filter enable control          | Controls the Analog Filter on the RX line in STOP mode:<br><ul style="list-style-type: none"> <li>0b0: Analog filter disable</li> <li>0b1: Analog filter enable. for the filter range, refer to the setting of USART1_NFS.</li> </ul>                                                                                                                                                                                                                                                       | RW     |
| 1:0  | USART1_NFS | USART1 Filter frequency range setting | Controls the filtering range of the analog filter on the RX line in STOP mode.<br><ul style="list-style-type: none"> <li>0b00: Filter width gear 1</li> <li>0b01: Filter width gear 2</li> <li>0b10: Filter width gear 3</li> <li>0b11: Filter width gear 4</li> </ul> For specific values of each setting, please refer to the relevant description in the "Electrical Specifications - Operating conditions - RX Filtering Characteristics in USART1 STOP Mode" chapter of the Datasheet. | RW     |



## 29.5 Precautions

### 29.5.1 UART Precautions

#### Transmitter

When the transmitter is disabled in UART mode (USARTn\_CR1.TE=0), the TX pin can be used as a general-purpose I/O, allowing output value and direction configuration. If the output is 0, it will cause a frame error on the receiver side, thereby triggering an interrupt in data transmission. If the output is 1, the receiver cannot detect the start bit, thus preventing data transmission from starting.

#### Receiver

In UART mode, when a frame error occurs, software can detect whether the subsequent RX line is at a low level to determine if the sender intends to interrupt the transmission.

If the start bit detection mode for receive data is set to low level detection, continuing to receive all low level data after clearing the error flag will cause the receive error to occur again.

### 29.5.2 Clock Synchronous Mode Precautions

1. When using an external input clock for data transmission, the USARTn\_TDR.TDR register must be updated before the clock is applied. After writing data to the register, the clock input must be delayed by at least one bit time.
2. When transmitting data continuously, the next frame must be updated before the last bit of the current frame is transmitted.

### 29.5.3 Other Precautions

To prevent the TX communication line from entering Hi-Z state when transmission is disabled, the following methods can be adopted:

- Communication line pull-up;
- Before the end of data transmission and before USARTn\_CR1.TE=0, configure the TX pin as a general-purpose IO output.
- Before transmitting data starts, after USARTn\_CR1.TE=1, configure the IO as TX function.

## 30 Inter-Integrated Circuit (I2C)

### 30.1 Introduction

I2C (Inter-Integrated Circuit) used as an interface between the microcontroller and the I2C serial bus. Provide multi-master mode function, which can control the protocol and arbitration of all I2C buses. Standard mode and fast mode are supported. SMBus is also supported.

### 30.2 Features

- I2C bus mode and SMBUS bus mode are optional. Master mode and slave mode are optional. Automatically ensure various preparation times, hold times, and bus idle times corresponding to the transfer rate
- Standard mode up to 100kbps, fast mode up to 400kbps
- The start condition, restart condition, and stop condition are automatically generated, and the start condition, restart condition, and stop condition of the bus can be detected
- Supports up to 128 slave mode addresses. Supports 7-bit address format and 10-bit address format. Broadcast call address, SMBus host address, SMBus device default address, SMBus alarm address can be detected
- Answer bits can be automatically determined when sent. Answer bits can be automatically sent when reception
- Handshake function
- Arbitration function
- Timeout function, can detect SCL clock stop for a long time
- SCL input and SDA input have built-in digital filters that are programmable to filter
- Communication error, receive data full, send data empty, one frame send end, address match unani-mously interrupt
- 2-stage transmit FIFO and 2-stage receive FIFO

## 30.3 Functional Description

### 30.3.1 I2C System Block Diagram

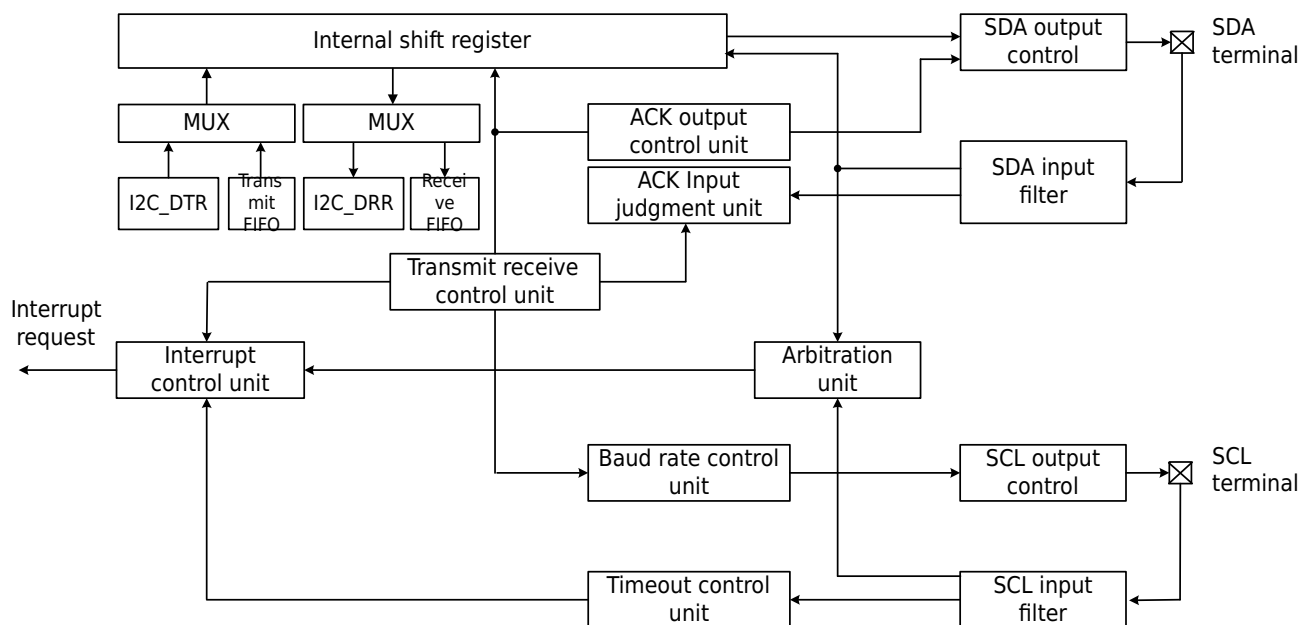


Figure 30-1 I2C System Block Diagram

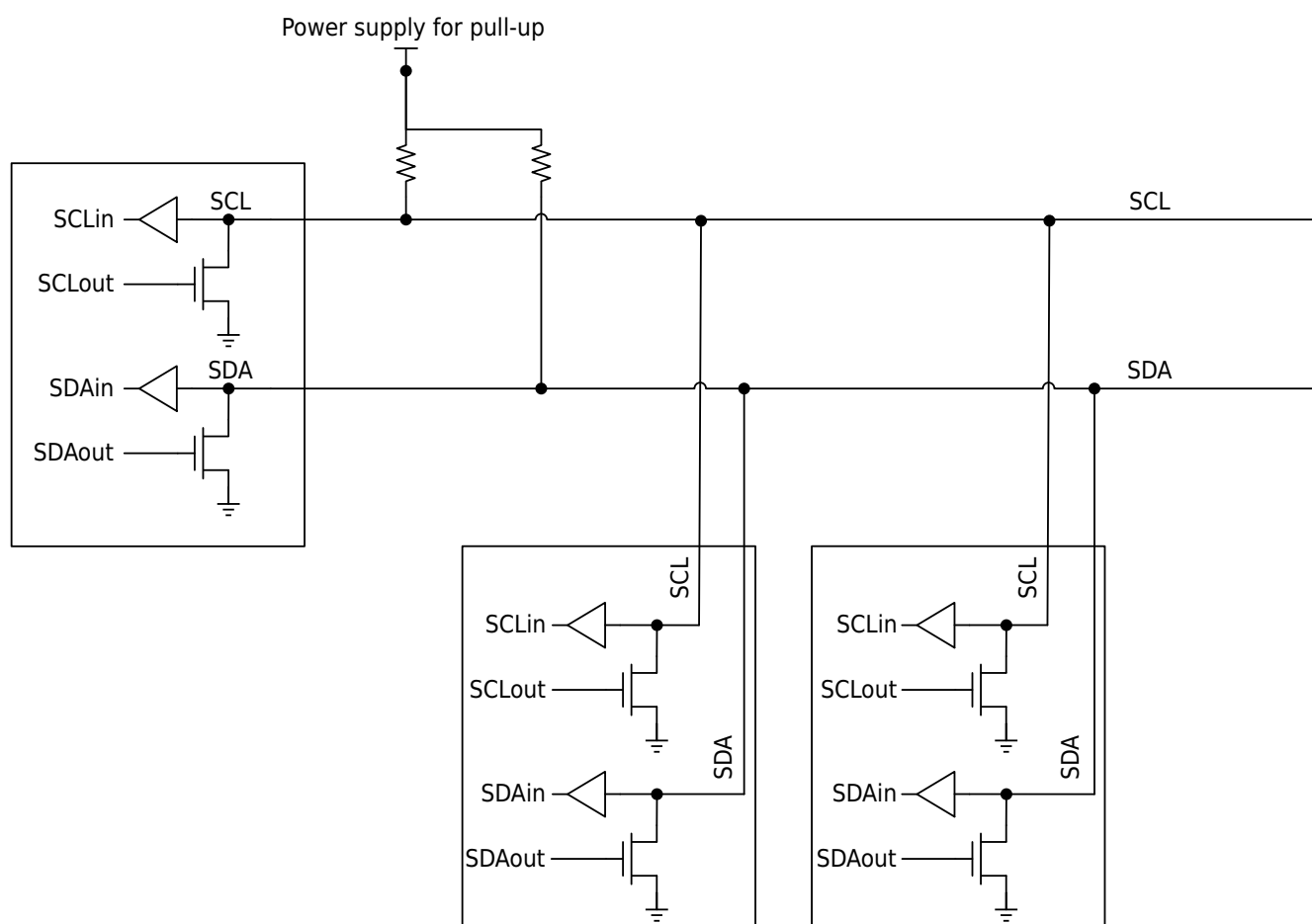


Figure 30-2 Example of I2C Bus Structure

Table 30-1 Input/Output Pin

| Pin Name | Input/Output | Function                         |
|----------|--------------|----------------------------------|
| SCL      | Input/Output | Input/output pin of serial clock |
| SDA      | Input/Output | Input/output pin of serial data  |

When the I2C bus is selected, the SCL/SDA input level is the Schmitt level (compatible with the CMOS level). When the SMBus is selected, please set the SCL/SDA input level to the CMOS level by setting PCRxy.CINSEL in "General-Purpose IO (GPIO) " "Register-General Control Register (PCRxy) ".

### 30.3.2 I2C Protocol

#### 30.3.2.1 Data Transmission

I2C bus consists of a clock line (SCL) and a data line (SDA). All connection devices must be drain open output. The SCL and SDA lines are connected to external pull-up resistors, and the resistance value depends on the system application.

In general, a complete communication process consists of the following four parts:

1. Start Condition
2. Address Transmission
3. Data Transmission
4. Stop Condition

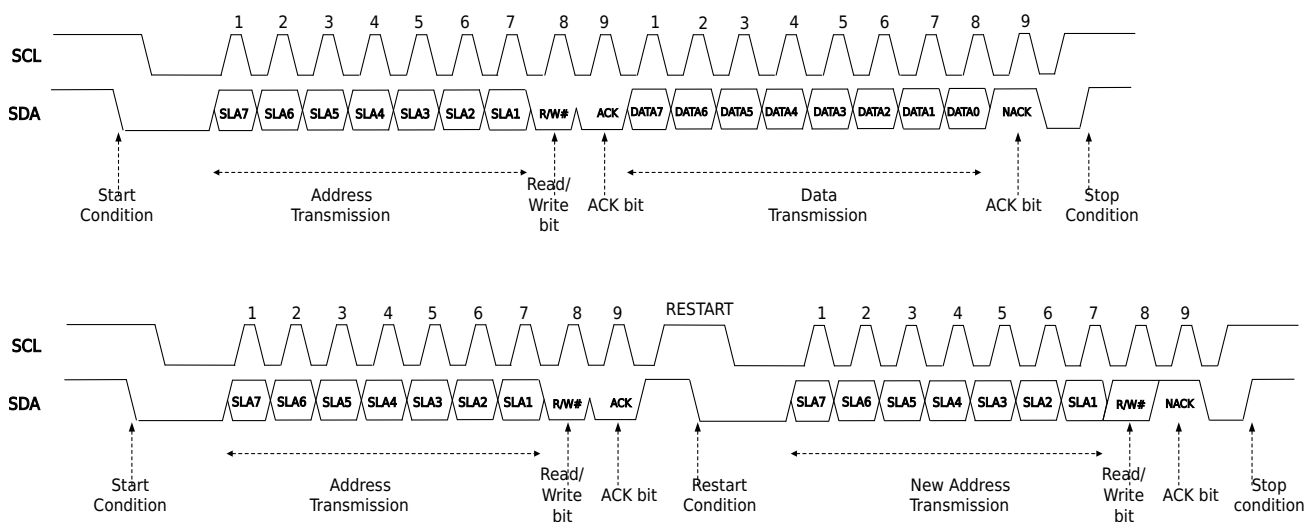


Figure 30-3 I2C Bus Timing Diagram

#### Start Condition

When the master on the bus does not drive the bus, the bus enters the idle state. Both SCL and SDA are high level. Devices on the bus in idle state can start the communication by sending the start condition.

If the START position is set to 1 in the I2C\_SR.BUSY flag of 0 (bus is idle), the start condition is issued. If a start condition is detected, the I2C\_SR.BUSY flag and I2C\_SR.STARTF flag are automatically set to 1 and the START bit is automatically set to 0. At this point, if the status of the SDA signal sent in the START bit 1 is the same as that of the SDA line, and the start condition is detected, it is considered that the start condition is correctly issued by the START bit, and the I2C\_SR.MSL bit and I2C\_SR.TRA bit are automatically set to 1 and then changed to the master send mode. In addition, I2C\_SR.EMPTFYF

automatically changes to 1 because the TRA bit is 1. Next write the slave address to the I2C\_DTR register and send the address.

### Address Transmission

The frame after the start condition or restart condition is an address frame that specifies the object address for master communication. The specified slave is always enable until the stop condition is sent.

The top 7 bits of the address frame are the slave addresses. The eighth bit of the address frame determines the direction of the data frame.

1. The 7-bit addressing mode is shown in the following figure [7-bit address format]

Master send mode, master send address frame 8th bit is 0;

Master receive mode, master send address frame 8th bit is 1.

2. The 10-bit addressing mode is shown in the following figure [10-bit address format]

Master send mode, master first frame send header sequence (0b11110XX0, where XX represents the top two digits of the 10-bit address), then the second frame send the low eight-bit slave address.

Master receive mode, master first frame send header sequence (0b11110XX0, where XX represents the top two digits of the 10-bit address), and then the second frame send the low eight-bit slave address. Next, a restart condition is sent, and then a frame header sequence is sent (0b11110XX1, where XX represents the top two digits of the 10-bit address).

#### 7-bit Address Format

|   |             |          |              |              |     |              |              |   |
|---|-------------|----------|--------------|--------------|-----|--------------|--------------|---|
| S | SLA (7-bit) | R/<br>W# | ACK/<br>NACK | DATA (8-bit) | ACK | DATA (8-bit) | ACK/<br>NACK | P |
|---|-------------|----------|--------------|--------------|-----|--------------|--------------|---|

#### 10-bit Address Format

|   |                         |    |     |             |     |              |     |              |              |   |
|---|-------------------------|----|-----|-------------|-----|--------------|-----|--------------|--------------|---|
| S | 0b11110+SL<br>A (2-bit) | W# | ACK | SLA (8-bit) | ACK | DATA (8-bit) | ACK | DATA (8-bit) | ACK/<br>NACK | P |
|---|-------------------------|----|-----|-------------|-----|--------------|-----|--------------|--------------|---|

|   |                         |    |     |             |     |    |                         |   |     |              |     |              |              |   |
|---|-------------------------|----|-----|-------------|-----|----|-------------------------|---|-----|--------------|-----|--------------|--------------|---|
| S | 0b11110+SL<br>A (2-bit) | W# | ACK | SLA (8-bit) | ACK | Sr | 0b11110+SL<br>A (2-bit) | R | ACK | DATA (8-bit) | ACK | DATA (8-bit) | ACK/<br>NACK | P |
|---|-------------------------|----|-----|-------------|-----|----|-------------------------|---|-----|--------------|-----|--------------|--------------|---|

Figure 30-4 Data Format of I2C Bus

- S: Indicates the start condition
- SLA: Indicates the slave address
- RW#: Indicates the direction of transmission and reception. Send data from slave to master when RW# is 1; When RW# is 0 the data is sent from the master to the slave.
- Sr: Indicates the restart condition
- DATA: Indicates the data sent and received
- P: Indicates the stop condition

### Data Transmission

After address matching is consistent, the master on the bus transmits data frame by frame according to the direction defined by R/W.

The data transmitted after all address frames are treated as data frames. Even a low 8-bit address in a 10-bit address format is considered a data frame.

The length of the data frame is 8 bits. The low-level SDA of the SCL changes, the high-level SDA of the SCL remains, and a bit of data is sent per clock period. The 9th clock after the data frame is an answer bit and is a handshake signal transmitted to the receiver.

If data is received from the slave on the bus and does not respond to the master in the ninth clock period, the slave must send NACK. If the master receives data on the bus, NACK is sent in the 9th period, NACK is received from the slave, and data is stopped from the slave.

Whether the master or the slave sends NACK, the data transfer interrupts. The master can do any of the following:

1. Transmit stop condition to release bus;
2. Send the restart condition to start a new communication.

#### ● Master transmitting data

In master transmit mode, the master outputs SCL clock and transmit data, the slave receives data and returns an answer. The sequence of master sending data is shown in the following figure.

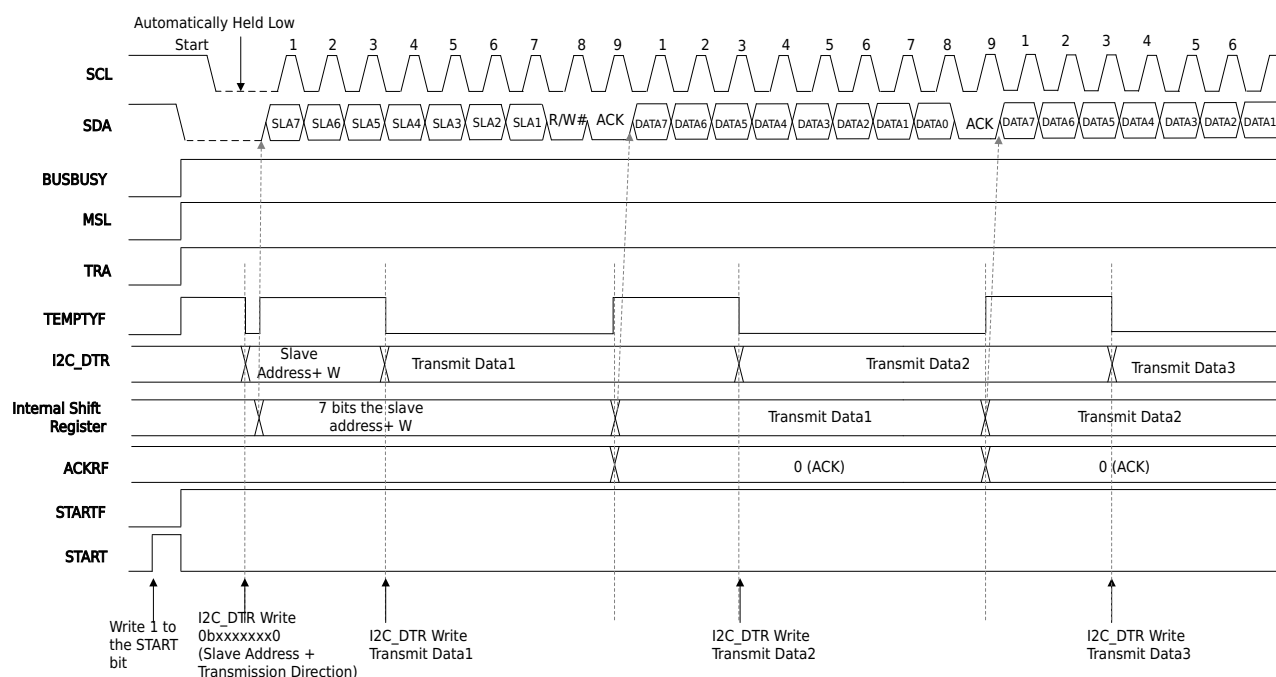


Figure 30-5 Sequence Diagram of Master Transmitting Data in 7-bit Address Format (Example)

#### ● Master receiving data

In master receiving mode, the master outputs an SCL clock, receives slave data, and returns an answer. The running sequence of receiving data by the master is shown in the following figure.

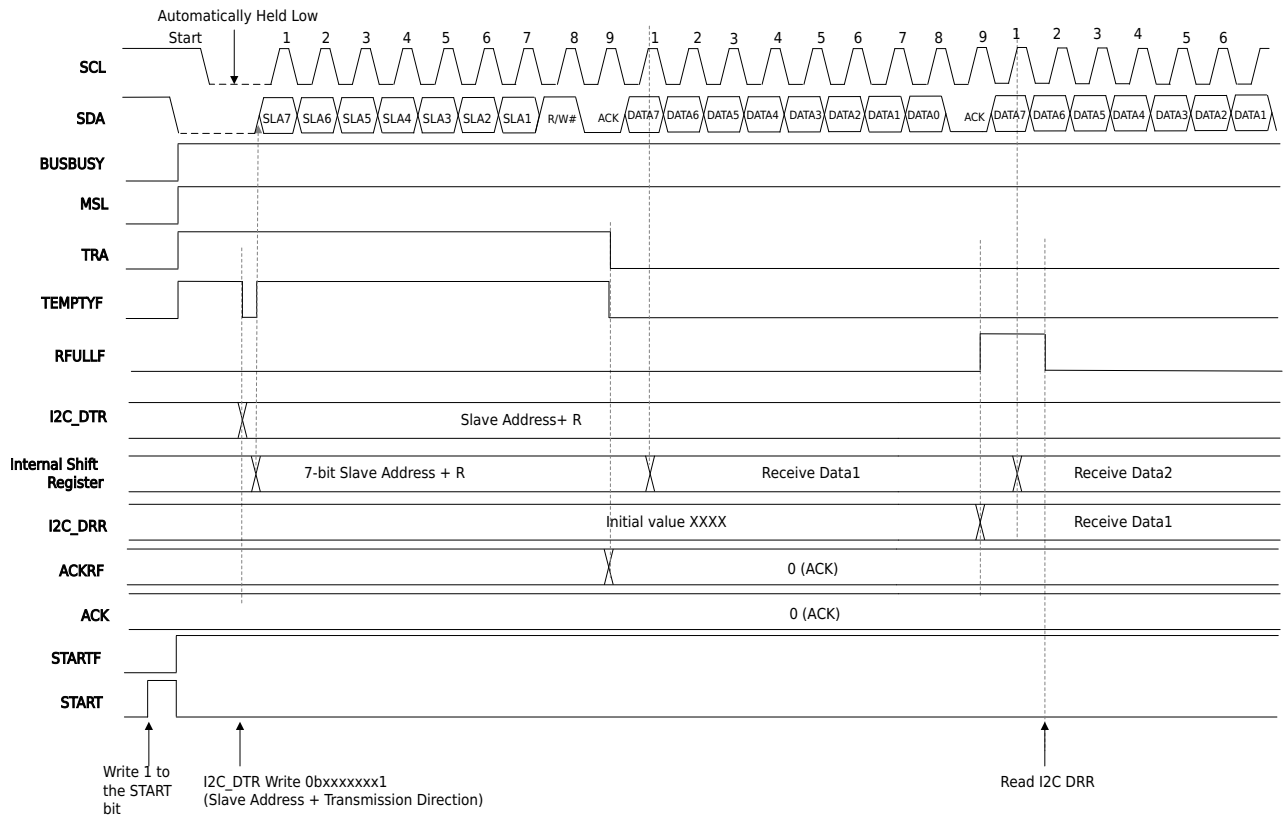


Figure 30-6 Sequence Diagram of Master Receiving Data in 7-bit Address Format (Example)

- Slave transmitting data

In slave send mode, the SCL clock from the master is received. This product sends data for the slave and receives a reply from the master. The sequence of data sent from the slave is shown in the following figure.

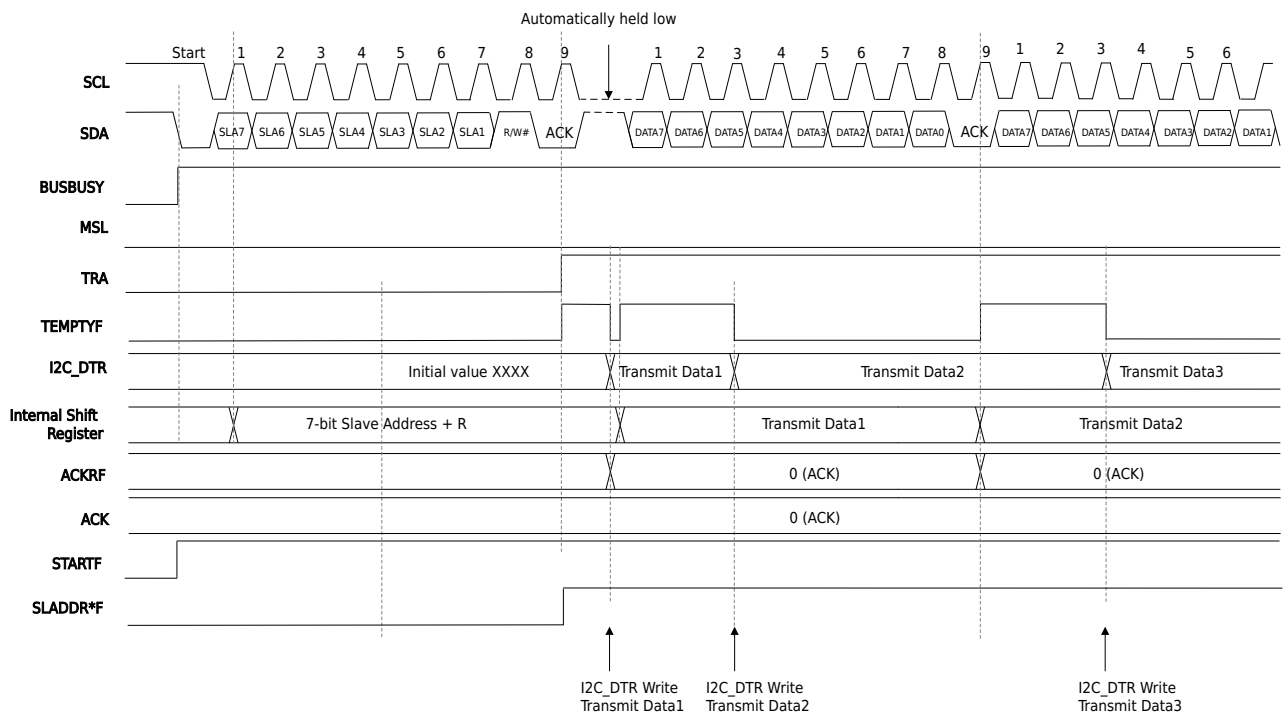


Figure 30-7 Sequence Diagram of Slave Transmitting Mode in 7-bit Address Format (Example)

### ● Slave receiving data

In slave receiving mode, the SCL clock and data from the master are received, and the response is returned after receiving the data. The sequence of data received from the slave is shown in the following figure.

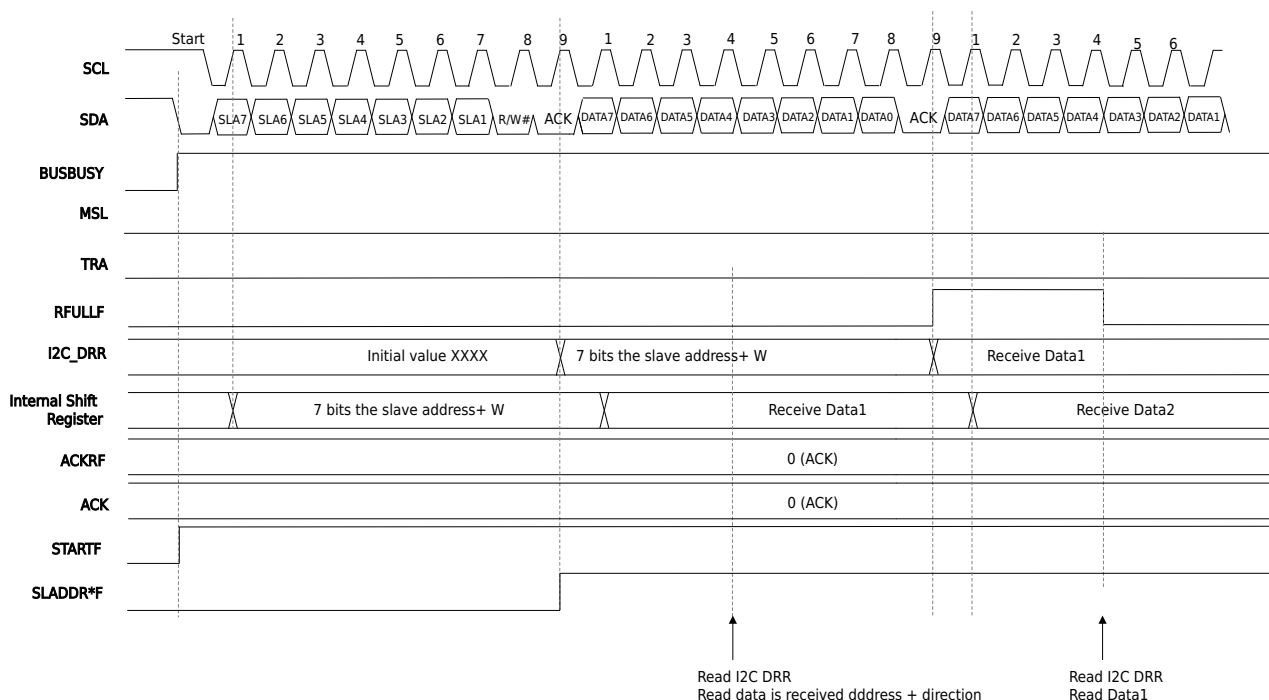


Figure 30-8 Sequence Diagram of Slave Receiving Data in 7-bit Address Format (Example)

### Stop Condition

A stop condition is sent via the I2C\_CR1.STOP bit.

When the I2C\_SR.BUSY flag is 1 (bus is busy) and the I2C\_SR.MSL bit is 1 (master mode), the STOP bit is set to 1 to issue a stop condition.

### Restart Condition

Through I2C\_CR1.RSTART bit generation restart condition.

The RSTART bit 1 generates a restart condition when the I2C\_SR.BUSY flag is 1 (busy bus) and the I2C\_SR.MSL bit is 1 (master mode).

By restarting the condition, the master can switch transmit/receive modes without releasing BUS authority. Communication can also be established with another slave without releasing BUS authority.

#### 30.3.2.2 SCL Clock Synchronization

When using the I2C bus in multi-master mode, there is a possibility that the SCL clock may conflict due to competition with other masters. If the SCL clock conflicts, the master needs to synchronize with the SCL clock and synchronize the SCL clock bit by bit. When the rising edge of the SCL line is detected and the high level set by I2C\_CCR.SHIGHW register is counted, if the SCL line is lowered due to the SCL clock output of other masters, the increment count of the high level width is stopped when the falling edge of the SCL line is detected, and the increment count of the low level width set by I2C\_CCR.SLOWW is started while the SCL line is driven to low level. When the counting of the low-level width ends, the low-level drive of the SCL line is terminated and the SCL line is released. At this time, if the low level width of the SCL clock of other masters is greater than the low-level width set by SLOWW, the low-level width of the SCL clock is prolonged. When other masters end the low-level output, release the SCL line



and the SCL clock rises. Therefore, in case of SCL clock output conflict, the high level width of SCL clock synchronizes with short clock, and the low level width synchronizes with long clock.

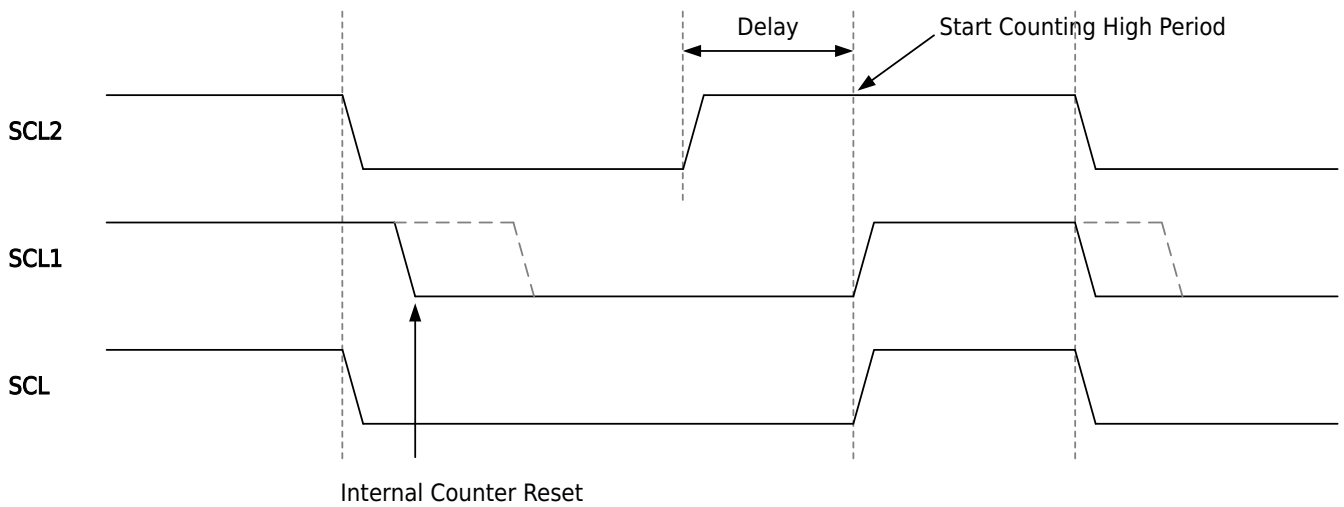


Figure 30-9 SCL Synchronous Timing

### 30.3.2.3 Arbitration

I2C bus is a real multi-master bus that allows multiple masters to connect.

If two or more masters try to control the bus at the same time, the SCL clock synchronization process determines the bus clock. The low period of the bus clock depends on the longest low-level clock and the high period depends on the shortest high-level clock. The results of arbitration are determined by the data collected at high levels. When the SDA output sent is a high-level output (the SDA pin is in a high impedance state) and the SDA line is detected to be at a low-level, arbitration fails. The I2C\_SR.AROLF bit will be set to "1" by hardware. If a master arbitration fails, it is immediately transferred to the slave receiving mode. At this point, if the slave addresses matches, including broadcast addresses, the slave mode continues to run.

### 30.3.2.4 Handshake

The handshake is realized by SCL clock synchronization mechanism during data transfer. After transmitting a frame of data (including ACK bits), the slave maintains the SCL clock line at low level. In this case, the low level of the SCL clock causes the master to enter the waiting state until the SCL line is released from the slave.

### Slave Transmitting Mode

In the transmitting mode (I2C\_SR.TRA bit=1), if the internal shift register is empty and the transmitting data (I2C\_DTR register) is not written, the SCL line is automatically kept low in the low level interval between the 9th clock and the 1st clock of the next transmission. The action timing is shown in the figure below.

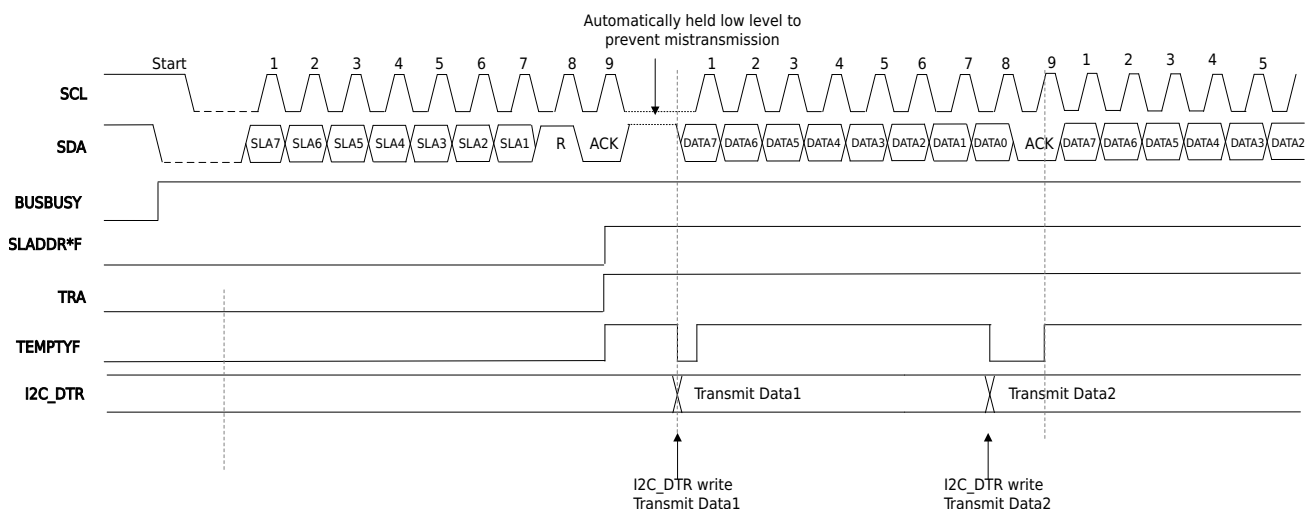


Figure 30-10 Transmitting Timing Diagram of Slave (1)

After the I2C\_SR.NACKF flag is changed to 1 or the last sent data is written to the I2C\_DTR register, wait until the I2C\_SR.TENDF flag changes to 1 while the I2C\_SR.EMPTFYF flag is 1. When the IC\_SR.NACKF flag or TENDF flag is 1, keep the SCL line low after the ninth clock drops. At this point, the communication must be ended by reading I2C\_DRR register, thereby releasing the SCL line.

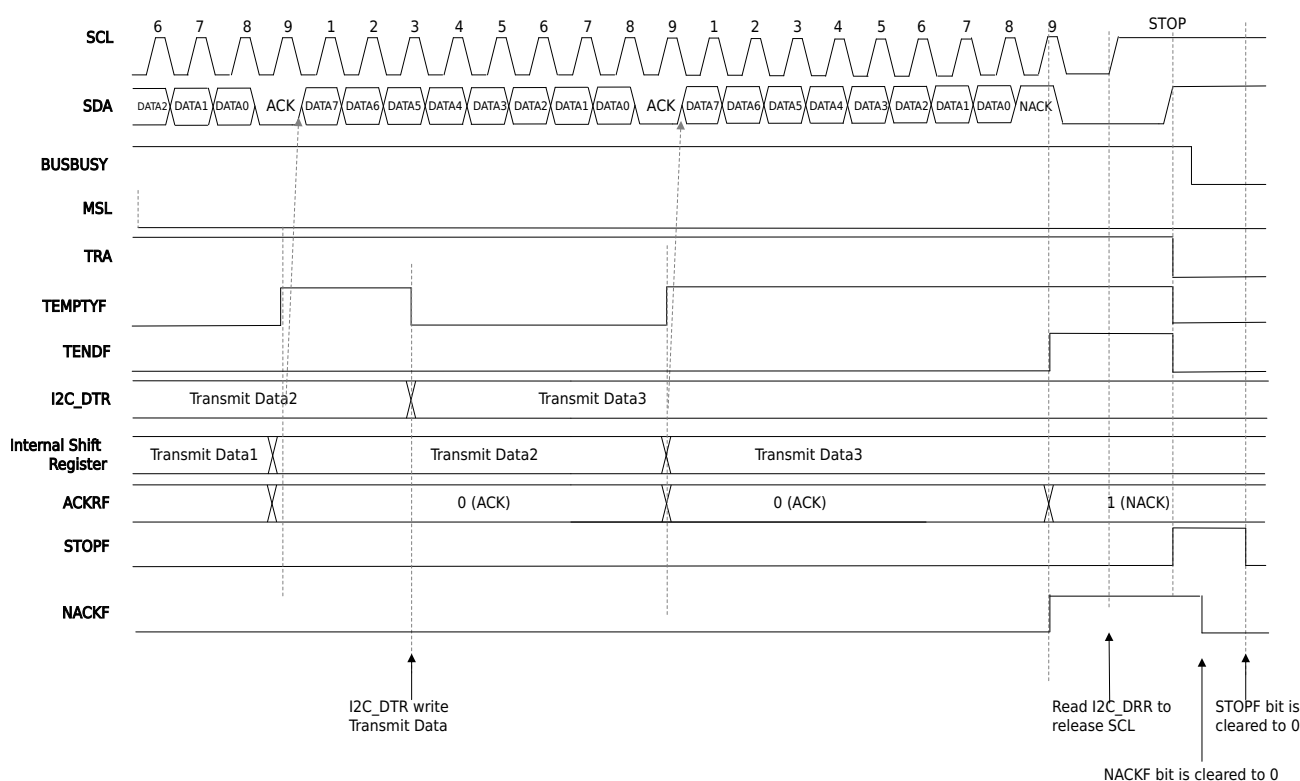


Figure 30-11 Transmitting Timing Diagram of Slave (2)

### Slave Receiving Mode

If a response processing delay occurs in receiving mode (I2C\_SR.TRA bit=0) and when the received data is full (I2C\_SR.RFULLF flag=1) due to the delay of at least one transmission frame read received data (I2C\_DRR register), etc., the SCL line is automatically maintained low between the 8th SCL and the 9th SCL clock, as shown in the following figure.

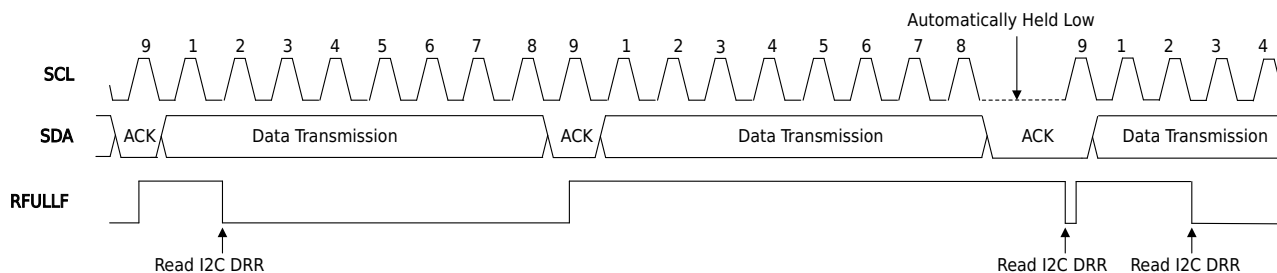


Figure 30-12 Slave Receiving Timing Diagram

### Fast ACK/NACK

In SMBUS communication, the data packet error code (PEC) of SMBUS is calculated or the received data is checked using the built-in CRC operator. During a PEC code check, transmits an ACK or NACK based on matching in the last byte. This requires the SCL to be kept low at the falling edge of the eighth clock of the last byte of the received SCL. This satisfies the software processing time. The software writes I2C\_CR1.ACK bit to remove SCL low level according to the calculation results. Fast ACK/NACK is controlled through I2C\_CR3.FACKEN bit, action sequence is shown in the following figure.

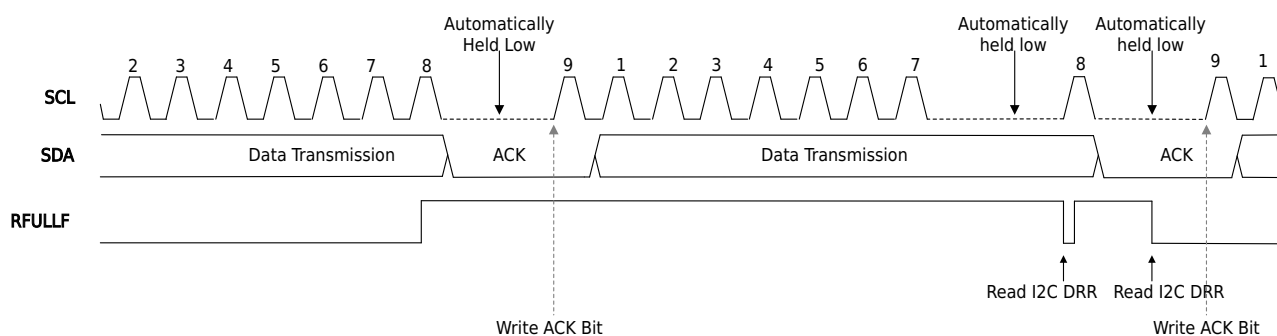


Figure 30-13 Fast ACK/NACK Timing Diagram

## 30.3.3 Address Matching

As a slave, two kinds of addresses other than broadcast address and master notification address can be set, and 7-bit address or 10-bit address format can be set for slave address.

### 30.3.3.1 Slave Address Matching

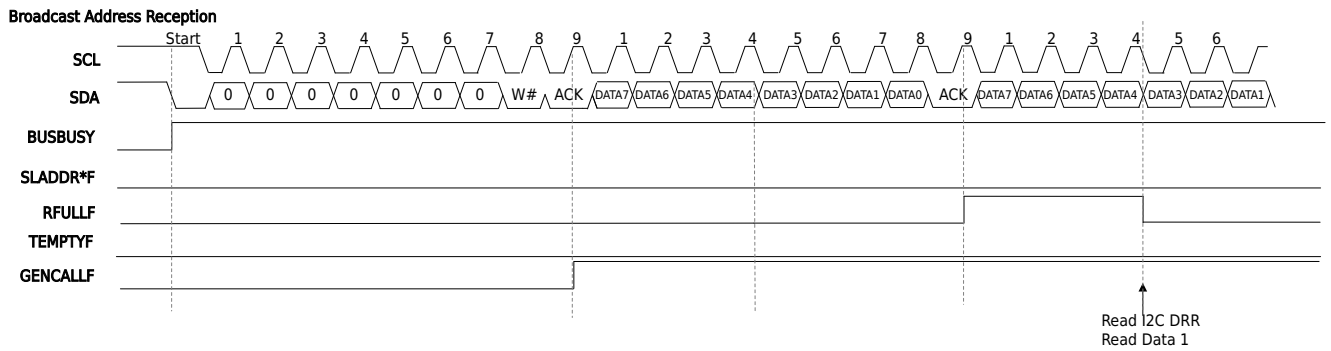
The I2C bus can set two kinds of slave addresses, and has corresponding slave address detection functions. When SLADDR1EN and SLADDR0EN are "1", the subordinate addresses set by I2C\_SLR1 and I2C\_SLR0 register can be detected.

If the set slave address matching are consistent, the corresponding SLADDR1F and SLADDR0F are set to 1 at the rising edge of the 9th clock of the SCL clock, and then the I2C\_SR.RFULLF flag or the I2C\_SR.EMPTIF flag is set to 1 according to the subsequent RW# bit. As a result, either receiving full interrupt or sending empty interrupt can be generated, and can determine which slave address is specified by confirming the I2C\_SR.SLADDR1F, SLADDR0F flags.

The timings of I2C\_SR.SLADDR1F and SLADDR0F flags changing to 1 are shown in the figure below.



The operation after the broadcast address matches is the same as the ordinary slave receiving operation.



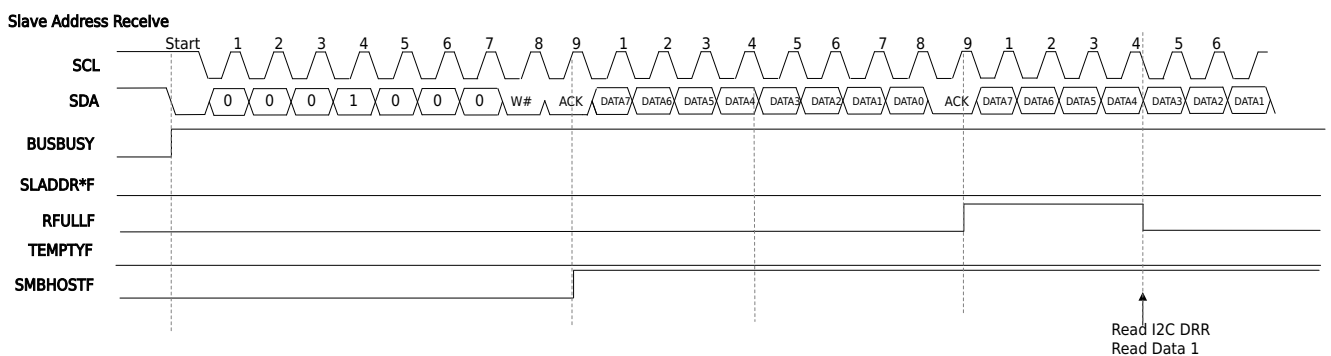
### Figure 30-16 Broadcast Address Matching Timing Diagram

### 30.3.3.3 SMBus Master Address Matching

This product has the master address detection function when SMBus runs. If the I2C\_CR1.SMBHOSTEN bit is set to 1 when the I2C\_CR1.SMBUS bit is 1, the master address (0b0001000) can be detected in slave receive mode (I2C\_SR.MSL bit TRA is 0b00).

If the SMBUS master address is detected, the I2C\_SR.SMBHOSTF flag is set to 1 at the rising edge of the 9th clock of SCL clock.

Even if the bit following the SMBUS master address (0b0001000) is the Rd bit (RW# bit receives 1), the SMBUS master address can be detected. The running of SMBUS master address detection is the same as that of normal slave mode.



### Figure 30-17 SMBus Master Address Matching Timing Diagram

#### 30.3.3.4 SMBus Alarm Response Address Matching

This product has the function of alarm response address detection when SMBus is running. If the I2C\_CR1.SMBARLERTEN bit is set to 1 when the I2C\_CR1.SMBUS bit is 1, the SMBUS alarm response address (0b0001100) can be detected in slave receive mode (I2C\_SR.MSL bit TRA is 0b00).

If the SMBUS alarm response address is detected, the I2C\_SR.SMBALRTF flag is set to 1 at the rising edge of the 9th clock of the SCL clock.

The operation of SMBUS alarm response address detection is the same as that of normal slave mode.

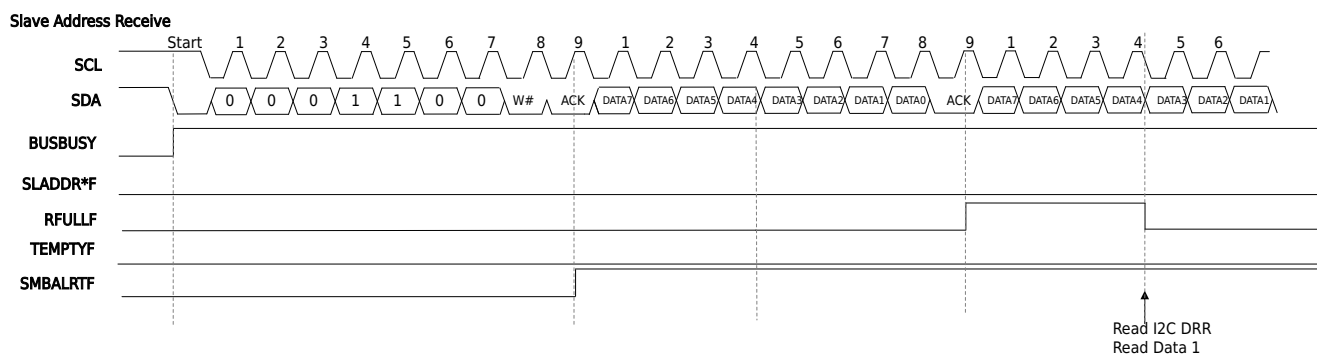


Figure 30-18 SMBus Alarm Response Address Matching Timing Diagram

### 30.3.3.5 SMBus Default Address Matching

This product has the default address detection function when SMBus runs. If the I2C\_CR1.SMBDEFAULTEN bit is set to 1 when the I2C\_CR1.SMBUS bit is 1, the SMBUS default address (0b1100001) can be detected in slave receive mode (I2C\_SR.MSL bit TRA is 0b00).

If the SMBUS default address is detected, the I2C\_SR.SMBDEFAULTF flag is set to 1 on the rising edge of the 9th clock of SCL clock.

The SMBUS default address detection runs the same as the normal slave mode.

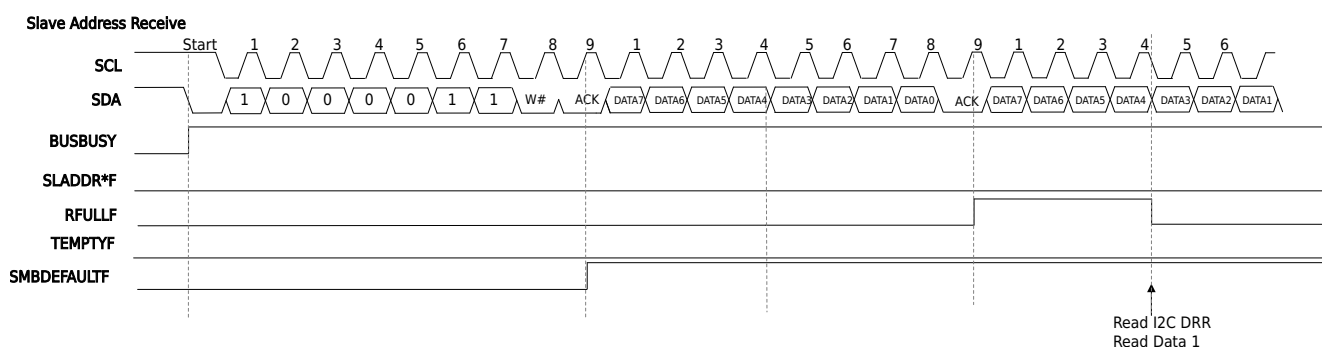


Figure 30-19 SMBus Default Address Timing Diagram

## 30.3.4 SMBus Action

This I2C interface can perform communication based on SMBus (Ver.2.0). When performing SMBus communication, I2C\_CR1.SMBUS bit must be set to 1. The transmission speed must be set within the SMBus specification range of 10kbps to 100kbps by setting the I2C\_CCR register.

### 30.3.4.1 SMBus Timeout Measurement

#### SCL Level Timeout Measurement

When working in slave mode, after the address match is consistent, it is detected that the low level or high level of the SCL line is fixed for more than a certain time, and the abnormal state of the bus is detected.

When working in the master mode, after the start condition is generated, it is detected that the low level or high level of SCL is fixed for more than a certain time, and the abnormal state of the bus is detected.

The timeout detection function monitors the state of the SCL line, and counts the time of high level or low level through the internal counter. If the SCL line changes (rising/falling), the internal counter is reset, otherwise it continues to count. If the internal counter counts up to the TOUTHIGH/TOUTLOW set value while the SCL line does not change, a timeout is detected and an abnormal state of the bus is notified.

For the counting of the internal counter, it can select whether to count at the low level or high level of the SCL line by setting the HTMOUT and LTMOUT bits, or to count at both low and high levels. If both the HTMOUT and LTMOUT bits are set to 0, no internal counting will be performed.

### Timeout Measurement of Slave

The slave device of SMBus communication needs to measure the interval shown below (timeout interval:  $T_{\text{LOW:SEXT}}$ ):

- Interval from start condition to stop condition

When timeout is measured by the slave device, use the start condition to detect interrupt and stop condition to detect interrupt, and use the chip timer to measure the time from detection to start condition to detection to stop condition. This timeout measurement time must be within the SMBus specification clock low level accumulation time [slave device]  $T_{\text{LOW:SEXT}}$ : 25ms (max).

If the timer measures longer than the timeout of the SMBus specification clock low level detection  $T_{\text{TIMEOUT}}$ : 25ms (min), the slave needs to release the bus.

### Timeout Measurement of Master

The main control device of SMBus communication needs to measure the interval shown below (timeout interval:  $T_{\text{LOW:MEXT}}$ ):

- Interval from start condition to answer bit
- Interval from answer to next answer
- Interval from answer bit to stop condition

When the master performs timeout measurement, use the start condition to detect interrupt, stop condition to detect interrupt, send end interrupt or receive data full interrupt, and use the chip timer to measure the time between intervals. The timeout time must be within the accumulated time [master]  $T_{\text{LOW:MEXT}}$ : 10ms (max) of the clock low level in the SMBus specification, and the cumulative result of all  $T_{\text{LOW:MEXT}}$  from the start condition to the stop condition must be within  $T_{\text{LOW:SEXT}}$ : 25ms (max).

If the time measured by the timer exceeds the cumulative time of the clock low level in the SMBus specification [master device]  $T_{\text{LOW:MEXT}}$ : 10ms (max), or the cumulative result of each measurement time exceeds the timeout of the clock low level detection in the SMBus specification  $T_{\text{TIMEOUT}}$ : 25ms (min), the master needs to terminate the processing. When sending a master, it must immediately abort the transmitting (write I2C\_DTR register). Abort the master transaction by transmitting a stop condition.

#### 30.3.4.2 Packet Error Code (PEC)

In communication, use the CPU to calculate the CRC, transmits the packet error code (PEC) of the SMBus, or checks the received data.

#### 30.3.5 Reset

It has the function of resetting the communication module. There are two types of reset, one is communication reset: all registers inside I2C (except I2C\_FSTR[3:0]) and internal state reset; the other is internal state reset: I2C\_SR (including BUSY bit), internal shift register and internal state machine are reset.

After reset, the I2C\_CR1.SWRST bit must be set to 0.

Because any kind of reset will remove the output state of the SCL pin /SDA pin and change it to a high-impedance state, it can also be used to remove the unexpected shutdown state of the bus.

The reset in the slave mode causes disparity with the master device, so try to avoid using it.

**Notice**

During reset (I2C\_CR1.PE bit and I2C\_CR1.SWRST bit are 0b01), the bus status such as the start condition cannot be monitored.

### 30.3.6 Programmable Digital Filtering

The status of SCL pin and SDA pin is taken internally via analog filter circuit and digital filter. A block diagram of the digital filter circuit is shown below.

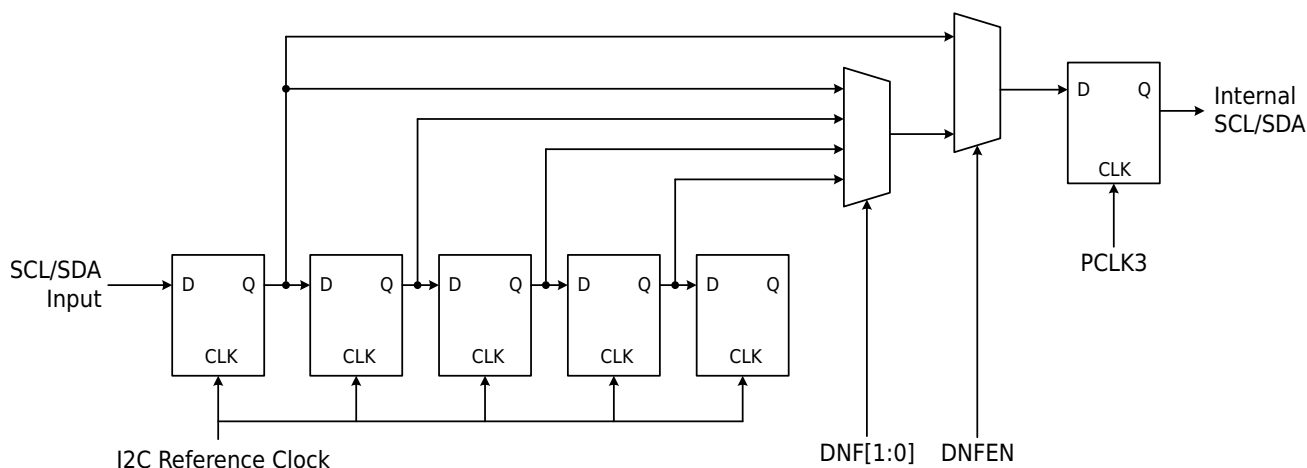


Figure 30-20 Digital Filtering Circuit Block Diagram

The internal digital filter circuit is composed of a four-segment series-connected trigger circuit and matching detection circuit.

The number of active segments of the digital filter is selected through the I2C\_FLTR.DNF bit. According to the selected active segment number, the noise elimination capability is 1- 4 I2C periods.

The input signal of the SCL pin (or the input signal of the SDA pin) is sampled on the falling edge of the I2C internal clock. If the output of the trigger circuit with the number of active segments set by the I2C\_FLTR.DNF bit all matches, the level is used as the internal signal to transmit, otherwise it keeps the original value.

### 30.3.7 Interrupt and Event

I2C has four interrupts and event outputs for triggering the activation of other peripheral circuits for user selection. Including: the occurrence of communication errors (arbitration failure detection, NACK detection, timeout detection, start condition detection, stop condition detection), end of receiving, transmitting data empty, end of transmitting.

The interrupt list is as follows.

Table 30-2 Interrupt List

| Name    | Interrupt Source                        | Interrupt Flag | Interrupt Condition     |
|---------|-----------------------------------------|----------------|-------------------------|
| I2C_EEI | Communication Error/Communication Event | ARLOF          | ARLOF=1&ARLOIE=1        |
|         |                                         | SLADDR0F       | SLADDR0F=1&SLADDR0IE=1  |
|         |                                         | SLADDR1F       | SLADDR1F=1&SLADDR1IE=1  |
|         |                                         | SMBALRTF       | SMBALRTF=1&SMBALRTIE=1  |
|         |                                         | SMBHOSTF       | SMBHOSTF=1&SMBHOSTFIE=1 |



| Name    | Interrupt Source                        | Interrupt Flag | Interrupt Condition          |
|---------|-----------------------------------------|----------------|------------------------------|
| I2C_EEI | Communication Error/Communication Event | SMBDEFAULTF    | SMBDEFAULTF=1&SMBDEFAULTIE=1 |
|         |                                         | GENCALLF       | GENCALLF=1&GENCALLIE=1       |
|         |                                         | NACKF          | NACKF=1&NACKIE=1             |
|         |                                         | TMOUTF         | TMOUTF=1&TMOUTIE=1           |
|         |                                         | STARTF         | STARTF=1&STARTIE=1           |
|         |                                         | STOPF          | STOPF=1&STOPIE=1             |
| I2C_RXI | Receive Data Full                       | RFULLF         | RFULLF=1&RFULLIE=1&-FEN      |
|         | Receive FIFO Data is not Empty          | RFREQF         | RFREQF=1&RFREQIE=1&FEN       |
| I2C_TXI | Transmit Data empty                     | EMPTYF         | EMPTYF=1&EMPTYIE=1           |
| I2C_TEI | End of Transmit                         | TENDF          | TENDF=1&TENDIE=1             |

The following table shows the output of the event signal.

Table 30-3 Event Signal Output List

| Name    | Event Source                            | Event Conditions |
|---------|-----------------------------------------|------------------|
| I2C_EEI | Communication Error/Communication Event | ARLOF=1          |
|         |                                         | SLADDR0F=1       |
|         |                                         | SLADDR1F=1       |
|         |                                         | SMBALRTF=1       |
|         |                                         | SMBHOSTF=1       |
|         |                                         | SMBDEFAULTF=1    |
|         |                                         | GENCALLF=1       |
|         |                                         | NACKF=1          |
|         |                                         | TMOUTF=1         |
|         |                                         | STARTF=1         |
|         |                                         | STOPF=1          |
| I2C_RXI | Receive Data Full                       | RFULLF=1         |
| I2C_TXI | Transmit Data Empty                     | EMPTYF=1         |
| I2C_TEI | End of Transmit                         | TENDF=1          |

## 30.4 Operation Example

### 30.4.1 I2C Initialization Process

When start transmitting or receiving data, it must follow the steps below to initialize:

**Step1.** I2C\_CR1.PE bit is set to 0;

**Step2.** Set I2C\_CR1.SWRST to 1, communication reset;

- Step3.** I2C\_CR1.PE bit set to 1, internal state reset;
- Step4.** Set slave address format and address;
- Step5.** Set baud rate;
- Step6.** Set control register function and interrupt as needed;
- Step7.** I2C\_CR1.SWRST bit is set to 0 to cancel internal state reset;
- Step8.** The initialization ends. Can transmit and receive data.

## 30.5 Register Description

### 30.5.1 Register List

- I2C1 base address: 0x4004E000
- I2C2 base address: 0x4004E400
- I2C3 base address: 0x4008E800
- I2C4 base address: 0x4008EC00
- I2C5 base address: 0x4008F000
- I2C6 base address: 0x4008F400

Table 30-4 Inter-Integrated Circuit (I2C) Register Offset

| Offset | Register    | Description                            | Width |
|--------|-------------|----------------------------------------|-------|
| 0x00   | I2C_CR1     | I2C Control Register 1                 | 32    |
| 0x04   | I2C_CR2     | I2C Control Register 2                 | 32    |
| 0x08   | I2C_CR3     | I2C Control Register 3                 | 32    |
| 0x0C   | I2C_CR4     | I2C Control Register 4                 | 32    |
| 0x10   | I2C_SLR0    | I2C Slave Address Register 0           | 32    |
| 0x14   | I2C_SLR1    | I2C Slave Address Register 1           | 32    |
| 0x18   | I2C_SLTR    | I2C SCL Level Timeout Control Register | 32    |
| 0x1C   | I2C_SR      | I2C Status Register                    | 32    |
| 0x20   | I2C_CLR     | I2C Status Clear Register              | 32    |
| 0x24   | I2C_DTR     | I2C Data Transmitting Register         | 8     |
| 0x28   | I2C_DRR     | I2C Data Receiving Register            | 8     |
| 0x2C   | I2C_CCR     | I2C Clock Control Register             | 32    |
| 0x30   | I2C_FLTR    | I2C Filter Control Register            | 32    |
| 0x34   | I2C_FSTR    | I2C FIFO Status Register               | 32    |
| 0x38   | I2C_SLVADRR | I2C Slave Address Receive Register     | 32    |

## 30.5.2 I2C Control Register 1 (I2C\_CR1)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |       |          |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |     |      |       |         |      |          |           |              |           |       |    |    |    |    |    |    |    |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-------|----------|----|----|----|----|---|---|---|---|---|---|---|---|---|---|--|-----|------|-------|---------|------|----------|-----------|--------------|-----------|-------|----|----|----|----|----|----|----|
| 0x00   | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15    | 14       | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |  |     |      |       |         |      |          |           |              |           |       |    |    |    |    |    |    |    |
| Reset  | 0x00000040   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |       |          |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |     |      |       |         |      |          |           |              |           |       |    |    |    |    |    |    |    |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | SWRST | Reserved |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  | ACK | STOP | START | RESTART | GCEN | Reserved | SMBHOSTEN | SMBDEFAULTEN | SMBALRTEN | SMBUS | PE |    |    |    |    |    |    |
|        |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |       |          |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |     |      |       |         |      |          |           |              |           |       |    |    |    |    |    |    |    |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RW    |          |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  | RW  | RW   | RW    | RW      | RW   |          | RW        | RW           | RW        | RW    | RW | RW | RW | RW | RW | RW | RW |

| Bit   | Name     | Function                                                                                                          | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | Access |    |               |     |     |                                                                                             |     |     |                                                                                                                   |    |
|-------|----------|-------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|----|---------------|-----|-----|---------------------------------------------------------------------------------------------|-----|-----|-------------------------------------------------------------------------------------------------------------------|----|
| 31:16 | Reserved | Reserved                                                                                                          | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | -      |    |               |     |     |                                                                                             |     |     |                                                                                                                   |    |
| 15    | SWRST    | Software Reset                                                                                                    | <div><div><div><div>● 0b0: Release reset</div><div>● 0b1: Software reset</div></div><div>By combining with the PE bit, select internal status reset or communication reset:</div><table><tr><th>SWRST</th><th>PE</th><th>Reset Content</th></tr><tr><td>0b1</td><td>0b0</td><td>Communication reset: re-set all registers (except I2C_FSTR[3:0]) and internal status of I2C</td></tr><tr><td>0b1</td><td>0b1</td><td>Internal status reset: I2C_SR (including BUSY bit), internal shift register and internal status machine are reset</td></tr></table></div></div> | SWRST  | PE | Reset Content | 0b1 | 0b0 | Communication reset: re-set all registers (except I2C_FSTR[3:0]) and internal status of I2C | 0b1 | 0b1 | Internal status reset: I2C_SR (including BUSY bit), internal shift register and internal status machine are reset | RW |
| SWRST | PE       | Reset Content                                                                                                     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |        |    |               |     |     |                                                                                             |     |     |                                                                                                                   |    |
| 0b1   | 0b0      | Communication reset: re-set all registers (except I2C_FSTR[3:0]) and internal status of I2C                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |        |    |               |     |     |                                                                                             |     |     |                                                                                                                   |    |
| 0b1   | 0b1      | Internal status reset: I2C_SR (including BUSY bit), internal shift register and internal status machine are reset |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |        |    |               |     |     |                                                                                             |     |     |                                                                                                                   |    |
| 14:11 | Reserved | Reserved                                                                                                          | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | -      |    |               |     |     |                                                                                             |     |     |                                                                                                                   |    |
| 10    | ACK      | Send Answer                                                                                                       | <div><div><div>● 0b0: The answer bit sends "0" (sends ACK)</div><div>● 0b1: The answer bit sends "1" (sends NACK)</div></div></div>                                                                                                                                                                                                                                                                                                                                                                                                                                  | RW     |    |               |     |     |                                                                                             |     |     |                                                                                                                   |    |
| 9     | STOP     | Stop Condition Generation Bit                                                                                     | <div><div><div><div>● 0b0: No stop condition is generated</div><div>● 0b1: Generate stop condition</div></div><div>This bit can be set and cleared by software.</div><div>Hardware clear condition:</div><div><div>● Stop condition detected</div><div>● Failure to arbitrate</div><div>● Start condition detected</div><div>● Communication reset</div></div></div></div>                                                                                                                                                                                           | RW     |    |               |     |     |                                                                                             |     |     |                                                                                                                   |    |

| Bit | Name            | Function                                           | Description                                                                                                                                                                                                                                                                                                                                                               | Access |
|-----|-----------------|----------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 8   | START           | Start Condition Generating Bit                     | <ul style="list-style-type: none"> <li>● 0b0: No start condition generated</li> <li>● 0b1: Start condition generated</li> </ul> This bit can be set and cleared by software.<br>Hardware clear condition: <ul style="list-style-type: none"> <li>● Start condition detected</li> <li>● Failure of arbitration</li> <li>● Communication reset</li> </ul>                   | RW     |
| 7   | RESTART         | Repeated Start Condition Generating Bit            | <ul style="list-style-type: none"> <li>● 0b0: No repeated start condition generated</li> <li>● 0b1: Repeated start condition generated</li> </ul> This bit can be set and cleared by software.<br>Hardware clear condition: <ul style="list-style-type: none"> <li>● Start condition detected</li> <li>● Failure of arbitration</li> <li>● Communication reset</li> </ul> | RW     |
| 6   | GCEN            | Broadcast Call Enabled                             | <ul style="list-style-type: none"> <li>● 0b0: Broadcast address detection disabled</li> <li>● 0b1: Broadcast address detection enabled</li> </ul>                                                                                                                                                                                                                         | RW     |
| 5   | <i>Reserved</i> | <i>Reserved</i>                                    | -                                                                                                                                                                                                                                                                                                                                                                         | -      |
| 4   | SMBHOSTEN       | Allow Matching of SMBUS Master Address Bit         | <ul style="list-style-type: none"> <li>● 0b0: SMBUS master address match disabled</li> <li>● 0b1: SMBUS master address match enabled</li> </ul>                                                                                                                                                                                                                           | RW     |
| 3   | SMBDEFAULTEN    | Allow Matching of SMBUS Default Address Bit        | <ul style="list-style-type: none"> <li>● 0b0: SMBUS default address match disabled</li> <li>● 0b1: SMBUS default address match enabled</li> </ul>                                                                                                                                                                                                                         | RW     |
| 2   | SMBALRTEN       | Allow Matching of SMBUS Alarm Response Address Bit | <ul style="list-style-type: none"> <li>● 0b0: SMBUS alarm response address disabled</li> <li>● 0b1: SMBUS alarm response address enabled</li> </ul>                                                                                                                                                                                                                       | RW     |
| 1   | SMBUS           | SMBUS/I2C Bus Mode Selection Bit                   | <ul style="list-style-type: none"> <li>● 0b0: I2C bus mode</li> <li>● 0b1: SMBUS mode</li> </ul>                                                                                                                                                                                                                                                                          | RW     |
| 0   | PE              | I2C Function Enabled                               | By combining with the SWRST bit, select internal status reset or communication reset: <ul style="list-style-type: none"> <li>● 0b0: I2C function disabled</li> <li>● 0b1: I2C function enabled</li> </ul>                                                                                                                                                                 | RW     |

## 30.5.3 I2C Control Register 2 (I2C\_CR2)

| Offset | Bit Position |    |    |    |    |    |    |    |           |          |    |    |    |    |    |    |    |         |          |        |         |          |        |          |         |         |          |        |        |           |           |         |    |    |  |  |  |  |  |  |  |  |  |  |  |  |
|--------|--------------|----|----|----|----|----|----|----|-----------|----------|----|----|----|----|----|----|----|---------|----------|--------|---------|----------|--------|----------|---------|---------|----------|--------|--------|-----------|-----------|---------|----|----|--|--|--|--|--|--|--|--|--|--|--|--|
| 0x04   | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23        | 22       | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14      | 13       | 12     | 11      | 10       | 9      | 8        | 7       | 6       | 5        | 4      | 3      | 2         | 1         | 0       |    |    |  |  |  |  |  |  |  |  |  |  |  |  |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |           |          |    |    |    |    |    |    |    |         |          |        |         |          |        |          |         |         |          |        |        |           |           |         |    |    |  |  |  |  |  |  |  |  |  |  |  |  |
| Name   | Reserved     |    |    |    |    |    |    |    | SMBALRTIE | Reserved |    |    |    |    |    |    |    | TMOUTIE | Reserved | NACKIE | RFREQIE | Reserved | ARLOIE | Reserved | EMPTYIE | RFULLIE | Reserved | STOPIE | TENDIE | SLADDR1IE | SLADDR0IE | STARTIE |    |    |  |  |  |  |  |  |  |  |  |  |  |  |
|        |              |    |    |    |    |    |    |    |           |          |    |    |    |    |    |    |    |         |          |        |         |          |        |          |         |         |          |        |        |           |           |         |    |    |  |  |  |  |  |  |  |  |  |  |  |  |
|        |              |    |    |    |    |    |    |    |           |          |    |    |    |    |    |    |    |         |          |        |         |          |        |          |         |         |          |        |        |           |           |         |    |    |  |  |  |  |  |  |  |  |  |  |  |  |
|        |              |    |    |    |    |    |    |    |           |          |    |    |    |    |    |    |    |         |          |        |         |          |        |          |         |         |          |        |        |           |           |         |    |    |  |  |  |  |  |  |  |  |  |  |  |  |
| Access |              |    |    |    |    |    |    |    | RW        |          |    |    |    |    |    |    |    | RW      |          | RW     | RW      |          | RW     |          | RW      | RW      |          | RW     | RW     | RW        | RW        | RW      | RW | RW |  |  |  |  |  |  |  |  |  |  |  |  |

| Bit   | Name         | Function                                                           | Description                                                                                               | Access |
|-------|--------------|--------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------|--------|
| 31:24 | Reserved     | Reserved                                                           | -                                                                                                         | -      |
| 23    | SMBALRTIE    | SMBUS Alarm Response Address Matching Consistent Interrupt Enabled | <ul style="list-style-type: none"> <li>0b0: Interrupt disabled</li> <li>0b1: Interrupt enabled</li> </ul> | RW     |
| 22    | SMBHOSTIE    | SMBUS Master Address Matching Consistent Interrupt Enabled         | <ul style="list-style-type: none"> <li>0b0: Interrupt disabled</li> <li>0b1: Interrupt enabled</li> </ul> | RW     |
| 21    | SMBDEFAULTIE | SMBUS Default Address Matching Consistent Interrupt Enabled        | <ul style="list-style-type: none"> <li>0b0: Interrupt disabled</li> <li>0b1: Interrupt enabled</li> </ul> | RW     |
| 20    | GENCALLIE    | Broadcast Call Address Matching Consistent Interrupt Enabled       | <ul style="list-style-type: none"> <li>0b0: Interrupt disabled</li> <li>0b1: Interrupt enabled</li> </ul> | RW     |
| 19:15 | Reserved     | Reserved                                                           | -                                                                                                         | -      |
| 14    | TMOUTIE      | Timeout Interrupt Enabled                                          | <ul style="list-style-type: none"> <li>0b0: Interrupt disabled</li> <li>0b1: Interrupt enabled</li> </ul> | RW     |
| 13    | Reserved     | Reserved                                                           | -                                                                                                         | -      |
| 12    | NACKIE       | NACK Interrupt Enabled                                             | <ul style="list-style-type: none"> <li>0b0: Interrupt disabled</li> <li>0b1: Interrupt enabled</li> </ul> | RW     |
| 11    | RFREQIE      | FIFO Receive Request Interrupt Enabled                             | <ul style="list-style-type: none"> <li>0b0: Interrupt disabled</li> <li>0b1: Interrupt enabled</li> </ul> | RW     |
| 10    | Reserved     | Reserved                                                           | -                                                                                                         | -      |

| Bit | Name            | Function                                                   | Description                                                                                                   | Access |
|-----|-----------------|------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------|--------|
| 9   | ARLOIE          | Arbitration Failure Interrupt Enabled                      | <ul style="list-style-type: none"> <li>● 0b0: Interrupt disabled</li> <li>● 0b1: Interrupt enabled</li> </ul> | RW     |
| 8   | <i>Reserved</i> | <i>Reserved</i>                                            | -                                                                                                             | -      |
| 7   | EMPTYIE         | Transmit Data Empty Interrupt Enabled                      | <ul style="list-style-type: none"> <li>● 0b0: Interrupt disabled</li> <li>● 0b1: Interrupt enabled</li> </ul> | RW     |
| 6   | RFULLIE         | Receive Data Full Interrupt Enabled                        | <ul style="list-style-type: none"> <li>● 0b0: Interrupt disabled</li> <li>● 0b1: Interrupt enabled</li> </ul> | RW     |
| 5   | <i>Reserved</i> | <i>Reserved</i>                                            | -                                                                                                             | -      |
| 4   | STOPIE          | Stop Condition Interrupt Enabled                           | <ul style="list-style-type: none"> <li>● 0b0: Interrupt disabled</li> <li>● 0b1: Interrupt enabled</li> </ul> | RW     |
| 3   | TENDIE          | Send a Frame of Data End Interrupt Enabled                 | <ul style="list-style-type: none"> <li>● 0b0: Interrupt disabled</li> <li>● 0b1: Interrupt enabled</li> </ul> | RW     |
| 2   | SLADDR1IE       | Slave Address 1 Match Consistent Interrupt Enabled         | <ul style="list-style-type: none"> <li>● 0b0: Interrupt disabled</li> <li>● 0b1: Interrupt enabled</li> </ul> | RW     |
| 1   | SLADDROIE       | Slave Address 0 Match Consistent Interrupt Enabled         | <ul style="list-style-type: none"> <li>● 0b0: Interrupt disabled</li> <li>● 0b1: Interrupt enabled</li> </ul> | RW     |
| 0   | STARTIE         | Start Condition/Repeated Start Condition Interrupt Enabled | <ul style="list-style-type: none"> <li>● 0b0: Interrupt disabled</li> <li>● 0b1: Interrupt enabled</li> </ul> | RW     |

## 30.5.4 I2C Control Register 3 (I2C\_CR3)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |        |          |   |   |        |        |         |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|--------|----------|---|---|--------|--------|---------|---|---|
| 0x08   | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8      | 7        | 6 | 5 | 4      | 3      | 2       | 1 | 0 |
| Reset  | 0x00000006   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |        |          |   |   |        |        |         |   |   |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   | FACKEN | Reserved |   |   | HTMOUT | LTMOUT | TMOUTEN |   |   |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   | RW     |          |   |   | RW     | RW     | RW      |   |   |

| Bit  | Name     | Function                                    | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | Access |
|------|----------|---------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:8 | Reserved | Reserved                                    | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | -      |
| 7    | FACKEN   | Time Point Selection of RFULLF Flag Bit Set | <ul style="list-style-type: none"> <li>0b0: When the 9th clock of SCL clock rises, the RFULLF flag is "1" (at the falling edge of the 8th clock, the SCL line does not remain at a low level)</li> <li>0b1: When the 8th clock of SCL clock rises, the RFULLF flag is "1" (at the falling edge of the 8th clock, the SCL line remains at a low level)</li> </ul> <div>  <b>Note</b><br/> Remove the held low level by writing the ACK bit. </div> | RW     |
| 6:3  | Reserved | Reserved                                    | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | -      |
| 2    | HTMOUT   | High Level Timeout Detection Enabled        | <ul style="list-style-type: none"> <li>0b0: Timeout detection disabled when SCL is high level</li> <li>0b1: Timeout detection enabled when SCL is high level</li> </ul>                                                                                                                                                                                                                                                                                                                                                              | RW     |
| 1    | LTMOUT   | Low Level Timeout Detection Enabled         | <ul style="list-style-type: none"> <li>0b0: Timeout detection disabled when SCL is low level</li> <li>0b1: Timeout detection enabled when SCL is low level</li> </ul>                                                                                                                                                                                                                                                                                                                                                                | RW     |
| 0    | TMOUTEN  | Timeout Function Enable Bit                 | <ul style="list-style-type: none"> <li>0b0: SCL level timeout detection function disabled</li> <li>0b1: SCL level timeout detection function enabled</li> </ul>                                                                                                                                                                                                                                                                                                                                                                      | RW     |



## 30.5.5 I2C Control Register 4 (I2C\_CR4)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |               |          |         |          |   |   |   |   |   |   |   |   |  |  |  |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---------------|----------|---------|----------|---|---|---|---|---|---|---|---|--|--|--|
| 0x0C   | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11            | 10       | 9       | 8        | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |  |  |  |
| Reset  | 0x00301B07   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |               |          |         |          |   |   |   |   |   |   |   |   |  |  |  |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | BUSFREE_CLREN | Reserved | BUSWAIT | Reserved |   |   |   |   |   |   |   |   |  |  |  |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RW            |          | RW      |          |   |   |   |   |   |   |   |   |  |  |  |

| Bit   | Name          | Function                 | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | Access |
|-------|---------------|--------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:22 | Reserved      | Reserved                 | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | -      |
| 21:20 | Reserved      | Reserved                 | All bits read as 1, Write as 1                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | -      |
| 19:13 | Reserved      | Reserved                 | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | -      |
| 12    | BUSFREE_CLREN | Bus End Status Clear Bit | <ul style="list-style-type: none"> <li>0b0: Disable the I2C bus STOP flag to clear the SMBALRTF, SMBHOSTF, SMBDEFAULTF, GENCALLF, MSL, SLADDR1F, SLADDR0F, and STARTF bits in the I2C_SLVADDR slave receive address register and the I2C_SR status register.</li> <li>0b1: Enable the I2C bus STOP flag to clear the SMBALRTF, SMBHOSTF, SMBDEFAULTF, GENCALLF, MSL, TENDF, SLADDR1F, SLADDR0F, and STARTF bits in the I2C_SLVADDR slave receive address register and the I2C_SR status register.</li> </ul> | RW     |
| 11    | Reserved      | Reserved                 | Reads as 1, Write as 1                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | -      |
| 10    | BUSWAIT       | Bus Wait Bit             | <ul style="list-style-type: none"> <li>0b0: When I2C_DRR is full and the internal shift register is empty, it does not maintain a low level between the 9th clock and the 1st clock of the next transmission, and continues to receive the next data</li> <li>0b1: When I2C_DRR is full and the internal shift register is empty, it maintains a low level between the 9th clock and the 1st clock of the next transmission. The low level is released by reading the I2C_DRR register.</li> </ul>           | RW     |
| 9:8   | Reserved      | Reserved                 | All bits read as 1, Write as 1                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | -      |
| 7:3   | Reserved      | Reserved                 | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | -      |
| 2:0   | Reserved      | Reserved                 | All bits read as 1, Write as 1                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | -      |

## 30.5.6 I2C Slave Address Register 0 (I2C\_SLR0)

| Offset | Bit Position |    |         |          |    |    |    |    |    |    |    |    |    |    |    |          |          |    |           |          |    |         |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|---------|----------|----|----|----|----|----|----|----|----|----|----|----|----------|----------|----|-----------|----------|----|---------|---|---|---|---|---|---|---|---|---|---|
| 0x10   | 31           | 30 | 29      | 28       | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16       | 15       | 14 | 13        | 12       | 11 | 10      | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00001000   |    |         |          |    |    |    |    |    |    |    |    |    |    |    |          |          |    |           |          |    |         |   |   |   |   |   |   |   |   |   |   |
| Name   | Reserved     |    | MASK0EN | MSLADDR0 |    |    |    |    |    |    |    |    |    |    |    | ADDRMOD0 | Reserved |    | SLADDR0EN | Reserved |    | SLADDR0 |   |   |   |   |   |   |   |   |   |   |
|        |              |    |         |          |    |    |    |    |    |    |    |    |    |    |    |          |          |    |           |          |    |         |   |   |   |   |   |   |   |   |   |   |
| Access |              |    | RW      | RW       |    |    |    |    |    |    |    |    |    |    |    | RW       |          | RW |           | RW       |    |         |   |   |   |   |   |   |   |   |   |   |

| Bit   | Name          | Function                                      | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | Access |
|-------|---------------|-----------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:27 | Reserved      | Reserved                                      | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | -      |
| 26    | MASK0EN       | Slave Address 0 Mask Enable Bit               | <ul style="list-style-type: none"> <li>0b0: The slave address register 0 mask setting disabled</li> <li>0b1: The slave address register 0 mask setting enabled</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                             | RW     |
| 25:24 | MSLADDR0[9:8] | 10-bit Slave Address High Mask Address        | Set the high mask address of the slave address 0: <ul style="list-style-type: none"> <li>When MASK0EN=0, the mask set by MSLADDR0[9:8] is invalid.</li> <li>When MASK0EN=1, the mask set by MSLADDR0[9:8] is valid. Its valid range is positively correlated with the value set by ADDRMOD0, namely:               <ul style="list-style-type: none"> <li>When ADDRMOD0=0, MSLADDR0[9:8] bits disabled;</li> <li>When ADDRMOD0=1, MSLADDR0[9:8] is the mask address of the upper two bits of the 10-bit slave address.</li> </ul> </li> </ul>                                         | RW     |
| 23:16 | MSLADDR0[7:0] | 7-bit Address/10-bit Address Low Mask Address | Set the low mask address of the slave address 0: <ul style="list-style-type: none"> <li>When MASK0EN=0, the mask set by MSLADDR0[7:0] is invalid.</li> <li>When MASK0EN=1, the mask set by MSLADDR0[7:0] is valid. Its valid range is positively correlated with the value set by ADDRMOD0, namely:               <ul style="list-style-type: none"> <li>When ADDRMOD0=0, MSLADDR0[7:1] is the 7-bit slave address mask, and MSLADDR0[0] bit is disabled;</li> <li>When ADDRMOD0=1, MSLADDR0[7:0] is the lower 8-bit address mask of the 10-bit slave address.</li> </ul> </li> </ul> | RW     |
| 15    | ADDRMOD0      | 7-bit/10-bit Address Format Selection Bit     | <ul style="list-style-type: none"> <li>0b0: Select 7-bit address format</li> <li>0b1: Select 10-bit address format</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                         | RW     |
| 14:13 | Reserved      | Reserved                                      | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | -      |

| Bit   | Name          | Function                                   | Description                                                                                                                                                                                                                                            | Access |
|-------|---------------|--------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 12    | SLADDR0EN     | Slave Address 0 Enabled Bit                | <ul style="list-style-type: none"> <li>0b0: Slave address register 0 setting value disabled</li> <li>0b1: Slave address register 0 setting value enabled</li> </ul>                                                                                    | RW     |
| 11:10 | Reserved      | Reserved                                   | -                                                                                                                                                                                                                                                      | -      |
| 9:8   | SLADDR0[9:8]  | High Bit of a 10-bit Slave Address.        | Set the slave address: <ul style="list-style-type: none"> <li>When ADDRMOD0=0, the SLADDR0[9:8] setting is invalid;</li> <li>When ADDRMOD0=1, SLADDR0[9:8] is the upper two bits of the 10-bit slave address.</li> </ul>                               | RW     |
| 7:0   | SLADDR0 [7:0] | Lower Bits of 7-bit Address/10-bit Address | Set the slave address: <ul style="list-style-type: none"> <li>When ADDRMOD0=0, SLADDR0[7:1] is the 7-bit slave address, and SLADDR0[0] is invalid;</li> <li>When ADDRMOD0=1, SLADDR0 [7:0] is the lower 8 bits of the 10-bit slave address.</li> </ul> |        |

## 30.5.7 I2C Slave Address Register 1 (I2C\_SLR1)

| Offset | Bit Position |    |         |          |    |    |    |    |    |    |    |          |          |           |          |         |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |
|--------|--------------|----|---------|----------|----|----|----|----|----|----|----|----------|----------|-----------|----------|---------|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|--|
| 0x14   | 31           | 30 | 29      | 28       | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20       | 19       | 18        | 17       | 16      | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |  |
| Reset  | 0x00000000   |    |         |          |    |    |    |    |    |    |    |          |          |           |          |         |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |
| Name   | Reserved     |    | MASK1EN | MSLADDR1 |    |    |    |    |    |    |    | ADDRMOD1 | Reserved | SLADDR1EN | Reserved | SLADDR1 |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |
|        |              |    |         |          |    |    |    |    |    |    |    |          |          |           |          |         |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |
| Access |              |    | RW      | RW       |    |    |    |    |    |    |    | RW       |          | RW        |          | RW      |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |

| Bit   | Name          | Function                                      | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | Access |
|-------|---------------|-----------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:27 | Reserved      | Reserved                                      | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | -      |
| 26    | MASK1EN       | Slave Address 1 Mask Enable Bit               | <ul style="list-style-type: none"> <li>0b0: The slave address register 1 mask setting disabled</li> <li>0b1: The slave address register 1 mask setting enabled</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                           | RW     |
| 25:24 | MSLADDR1[9:8] | 10-bit Slave Address High Mask Address        | Set the high mask address of the slave address 1: <ul style="list-style-type: none"> <li>When MASK1EN=0, the mask set by MSLADDR1[9:8] is invalid.</li> <li>When MASK1EN=1, the mask set by MSLADDR1[9:8] is valid. Its valid range is positively correlated with the value set by ADDRMOD 1, namely:               <ul style="list-style-type: none"> <li>When ADDRMOD1=0, MSLADDR1[9:8] bits disabled;</li> <li>When ADDRMOD1=1, MSLADDR1[9:8] is the mask address of the upper two bits of the 10-bit slave address.</li> </ul> </li> </ul>                                      | RW     |
| 23:16 | MSLADDR1[7:0] | 7-bit address/10-bit Address Low Mask Address | Set the low mask address of the slave address 1: <ul style="list-style-type: none"> <li>When MASK1EN=0, the mask set by MSLADDR1[7:0] is invalid.</li> <li>When MASK1EN=1, the mask set by MSLADDR1[7:0] is valid. Its valid range is positively correlated with the value set by ADDRMOD1, namely:               <ul style="list-style-type: none"> <li>When ADDRMOD1=0, MSLADDR1[7:1] is the 7-bit slave address mask, and MSLADDR1[0] bit is invalid;</li> <li>When ADDRMOD1= 1, MSLADDR1[7:0] is the low 8-bit address mask of the 10-bit slave address.</li> </ul> </li> </ul> | RW     |
| 15    | ADDRMOD1      | 7-bit/10-bit Address Format Selection Bit     | <ul style="list-style-type: none"> <li>0b0: Select 7-bit address format</li> <li>0b1: Select 10-bit address format</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                       | RW     |
| 14:13 | Reserved      | Reserved                                      | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | -      |

| Bit   | Name         | Function                                 | Description                                                                                                                                                                                                                                         | Access |
|-------|--------------|------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 12    | SLADDR1EN    | Slave Address 1 Enable Bit               | <ul style="list-style-type: none"> <li>0b0: Slave address register 1 setting value disabled</li> <li>0b1: Slave address register 1 setting value enabled</li> </ul>                                                                                 | RW     |
| 11:10 | Reserved     | Reserved                                 | -                                                                                                                                                                                                                                                   | -      |
| 9:8   | SLADDR1[9:8] | High Bit of a 10-bit Slave Address.      | Set the slave address: <ul style="list-style-type: none"> <li>When ADDRMOD1=0, the SLADDR1[9:8] setting is invalid;</li> <li>When ADDRMOD1=1, the SLADDR1[9:8] is the upper two bits of the 10-bit slave address.</li> </ul>                        | RW     |
| 7:0   | SLADDR1[7:0] | Low Bits of 7-bit Address/10-bit Address | Set the slave address: <ul style="list-style-type: none"> <li>When ADDRMOD1=0, SLADDR1[7:1] is the 7-bit slave address, and SLADDR1[0] is invalid;</li> <li>When ADDRMOD1=1, SLADDR1[7:0] is the low 8 bits of the 10-bit slave address.</li> </ul> |        |

### 30.5.8 I2C SCL Level Timeout Control Register (I2C\_SLTR)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |         |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0x18   | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15      | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0xFFFFFFFF   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |         |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | TOUTHIGH     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | TOUTLOW |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access | RW           |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RW      |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit   | Name     | Function                      | Description                                                     | Access |
|-------|----------|-------------------------------|-----------------------------------------------------------------|--------|
| 31:16 | TOUTHIGH | SCL high level timeout period | SCL high level timeout time=TOUTHIGH*I2C reference clock period | RW     |
| 15:0  | TOUTLOW  | SCL low level timeout period  | SCL low level timeout time=TOUTLOW*I2C reference clock period   | RW     |

## 30.5.9 I2C Status Register (I2C\_SR)

| Offset | Bit Position |          |      |        |         |        |         |          |          |             |          |          |     |      |     |          |
|--------|--------------|----------|------|--------|---------|--------|---------|----------|----------|-------------|----------|----------|-----|------|-----|----------|
| 0x1C   | 31           | 30       | 29   | 28     | 27      | 26     | 25      | 24       | 23       | 22          | 21       | 20       | 19  | 18   | 17  | 16       |
| Reset  | 0x05000000   |          |      |        |         |        |         |          |          |             |          |          |     |      |     |          |
| Name   | RFREQ        | Reserved | TFST | RFFULL | RFEMPTY | TFFULL | TFEMPTY | SMBALRTF | SMBHOSTF | SMBDEFAULTF | GENCALLF | Reserved | TRA | BUSY | MSL | Reserved |
| Access | RO           |          | RO   | RO     | RO      | RO     | RO      | RO       | RO       | RO          | RO       |          | RW  | RO   | RW  | RO       |

| Bit   | Name     | Function                                                  | Description                                                                                                                                                                                                                                                                                                                                                                                                                                            | Access |
|-------|----------|-----------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31    | RFREQ    | Receive FIFO request flag bit                             | <ul style="list-style-type: none"> <li>0b0: Current receive FIFO without enable data</li> <li>0b1: Current receive FIFO with enable data</li> </ul>                                                                                                                                                                                                                                                                                                    | RO     |
| 30    | Reserved | Reserved                                                  | -                                                                                                                                                                                                                                                                                                                                                                                                                                                      | -      |
| 29:28 | TFST     | Transmit FIFO status flag bit                             | <ul style="list-style-type: none"> <li>0b00: Current transmit FIFO is empty</li> <li>0b01: Current transmit FIFO with 1 Byte enable data</li> <li>0b10: Current transmit FIFO with 2 Bytes enable data</li> <li>0b11: Reserved</li> </ul>                                                                                                                                                                                                              | RO     |
| 27    | RFFULL   | Receive FIFO full flag bit                                | <ul style="list-style-type: none"> <li>0b0: Current receive FIFO not full</li> <li>0b1: Current receive FIFO full</li> </ul>                                                                                                                                                                                                                                                                                                                           | RO     |
| 26    | RFEMPTY  | Receive FIFO empty flag bit                               | <ul style="list-style-type: none"> <li>0b0: Current receive FIFO not empty</li> <li>0b1: Current receive FIFO empty</li> </ul>                                                                                                                                                                                                                                                                                                                         | RO     |
| 25    | TFFULL   | Transmit FIFO full flag bit                               | <ul style="list-style-type: none"> <li>0b0: Current transmit FIFO not full</li> <li>0b1: Current transmit FIFO full</li> </ul>                                                                                                                                                                                                                                                                                                                         | RO     |
| 24    | TFEMPTY  | Transmit FIFO empty flag bit                              | <ul style="list-style-type: none"> <li>0b0: Current transmit FIFO not empty</li> <li>0b1: Current transmit FIFO is empty</li> </ul>                                                                                                                                                                                                                                                                                                                    | RO     |
| 23    | SMBALRTF | SMBUS alarm response address matching consistent flag bit | <ul style="list-style-type: none"> <li>0b0: SMBUS alarm response address not matched</li> <li>0b1: Master address detected</li> </ul> <p>Setting conditions:</p> <ul style="list-style-type: none"> <li>Received address matches 0b0001100</li> </ul> <p>Clearing Conditions:</p> <ul style="list-style-type: none"> <li>SMBALRTFCLR writes "1"</li> <li>Stop condition detected</li> <li>Communication reset</li> <li>Internal state reset</li> </ul> | RO     |

| Bit | Name        | Function                                            | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | Access |
|-----|-------------|-----------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 22  | SMBHOSTF    | SMBUS master address matching consistent flag bit   | <ul style="list-style-type: none"> <li>● 0b0: No match to SMBUS master address</li> <li>● 0b1: Match to SMBUS master address</li> </ul> Setting conditions: <ul style="list-style-type: none"> <li>● Received address matches 0b0001000</li> </ul> Clearing Conditions: <ul style="list-style-type: none"> <li>● SMBHOSTFCLR writes "1"</li> <li>● Stop condition detected</li> <li>● Communication reset</li> <li>● Internal state reset</li> </ul>                                                         | RO     |
| 21  | SMBDEFAULTF | SMBUS default address matching consistent flag bit  | <ul style="list-style-type: none"> <li>● 0b0: No match with the SMBUS default address.</li> <li>● 0b1: A match with the SMBUS default address.</li> </ul> Setting conditions: <ul style="list-style-type: none"> <li>● Received address matches 0b1100001</li> </ul> Clearing Conditions: <ul style="list-style-type: none"> <li>● SMBDEFAULTFCLR Write "1"</li> <li>● Stop condition detected</li> <li>● Communication reset</li> <li>● Internal state reset</li> </ul>                                     | RO     |
| 20  | GENCALLF    | Broadcast call address matching consistent flag bit | <ul style="list-style-type: none"> <li>● 0b0: No match with the broadcast call address.</li> <li>● 0b1: A match with the broadcast call address.</li> </ul> Setting conditions: <ul style="list-style-type: none"> <li>● The received slave address matches the broadcast call address (All "0")</li> </ul> Clearing Conditions: <ul style="list-style-type: none"> <li>● GENCALLFCLR Write "1"</li> <li>● Stop condition detected</li> <li>● Communication reset</li> <li>● Internal state reset</li> </ul> | RO     |
| 19  | Reserved    | Reserved                                            | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | -      |

| Bit | Name     | Function                       | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | Access |     |                    |     |     |                    |     |     |                     |     |     |                     |     |     |                      |    |
|-----|----------|--------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|-----|--------------------|-----|-----|--------------------|-----|-----|---------------------|-----|-----|---------------------|-----|-----|----------------------|----|
| 18  | TRA      | Transmit receive selection bit | <div><div><div><div>● 0b0: Receive data</div><div>● 0b1: Transmit data</div></div><div>The operating mode of I2C is represented by a combination of MSL bits:</div><table><tr><th>MSL</th><th>TRA</th><th>I2C operating mode</th></tr><tr><td>0b0</td><td>0b0</td><td>Slave receive mode</td></tr><tr><td>0b0</td><td>0b1</td><td>Slave transmit Mode</td></tr><tr><td>0b1</td><td>0b0</td><td>Master receive mode</td></tr><tr><td>0b1</td><td>0b1</td><td>Master transmit mode</td></tr></table></div><div><div>This bit can be set and cleared by software.</div><div>Hardware set conditions:</div><div><div>● Start condition detected</div><div>● In master mode, the transmitted RW bit is 0</div><div>● In slave mode, the address matches and the received RW bit is 1</div></div><div>Hardware clear condition</div><div><div>● Stop condition detected</div><div>● In master mode, the transmitted RW bit is 1</div><div>● In slave mode, the address matches and the received RW bit is 0</div><div>● Communication reset</div><div>● Internal state reset</div></div></div></div> | MSL    | TRA | I2C operating mode | 0b0 | 0b0 | Slave receive mode | 0b0 | 0b1 | Slave transmit Mode | 0b1 | 0b0 | Master receive mode | 0b1 | 0b1 | Master transmit mode | RW |
| MSL | TRA      | I2C operating mode             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |        |     |                    |     |     |                    |     |     |                     |     |     |                     |     |     |                      |    |
| 0b0 | 0b0      | Slave receive mode             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |        |     |                    |     |     |                    |     |     |                     |     |     |                     |     |     |                      |    |
| 0b0 | 0b1      | Slave transmit Mode            |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |        |     |                    |     |     |                    |     |     |                     |     |     |                     |     |     |                      |    |
| 0b1 | 0b0      | Master receive mode            |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |        |     |                    |     |     |                    |     |     |                     |     |     |                     |     |     |                      |    |
| 0b1 | 0b1      | Master transmit mode           |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |        |     |                    |     |     |                    |     |     |                     |     |     |                     |     |     |                      |    |
| 17  | BUSY     | Bus busy flag bit              | <div><div><div>● 0b0: Idle state, no communication on the bus</div><div>● 0b1: Occupied state, the bus is communicating</div></div><div>Setting conditions:</div><div><div>● Start condition detected on bus</div></div><div>Clearing Conditions:</div><div><div>● Bus stop condition detected</div><div>● Communication reset</div></div></div>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | RO     |     |                    |     |     |                    |     |     |                     |     |     |                     |     |     |                      |    |
| 16  | MSL      | Master-slave selection bit     | <div><div><div>● 0b0: Slave mode</div><div>● 0b1: Master mode</div></div><div>The operating mode of I2C is represented by a combination of TRA bits.</div><div>This bit can be set and cleared by software.</div><div>Hardware set conditions:</div><div><div>● Start condition detected when START bit is 1</div></div><div>Hardware clear condition</div><div><div>● Stop condition detected</div><div>● Failure to arbitrate</div><div>● Communication reset</div><div>● Internal state reset</div></div></div>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | RW     |     |                    |     |     |                    |     |     |                     |     |     |                     |     |     |                      |    |
| 15  | Reserved | Reserved                       | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | -      |     |                    |     |     |                    |     |     |                     |     |     |                     |     |     |                      |    |



| Bit | Name     | Function                     | Description                                                                                                                                                                                                                                                                                                                                                                                                                                  | Access |
|-----|----------|------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 14  | TMOUTF   | Timeout flag bit             | <ul style="list-style-type: none"> <li>0b0: SCL level timeout not detected</li> <li>0b1: SCL level timeout</li> </ul> Setting conditions: <ul style="list-style-type: none"> <li>During the period set by I2C_SLTR, SCL is not inverted</li> </ul> Clearing conditions: <ul style="list-style-type: none"> <li>TMOUTFCLR writes "1"</li> <li>Communication reset</li> <li>Internal state reset</li> </ul>                                    | RO     |
| 13  | Reserved | Reserved                     | -                                                                                                                                                                                                                                                                                                                                                                                                                                            | -      |
| 12  | NACKF    | NACK flag bit                | <ul style="list-style-type: none"> <li>0b0: NACK not received</li> <li>0b1: NACK received</li> </ul> Setting conditions: <ul style="list-style-type: none"> <li>Receive NACK in transmit mode</li> </ul> Clearing Conditions: <ul style="list-style-type: none"> <li>NACKFCLR writes "1"</li> <li>Communication reset</li> <li>Internal state reset</li> </ul>                                                                               | RO     |
| 11  | Reserved | Reserved                     | -                                                                                                                                                                                                                                                                                                                                                                                                                                            | -      |
| 10  | ACKRF    | Receive acknowledge bit      | <ul style="list-style-type: none"> <li>0b0: The received acknowledge bit is "0" (receive ACK)</li> <li>0b1: The received acknowledge bit is "1" (receive NACK)</li> </ul> Setting conditions: <ul style="list-style-type: none"> <li>Receive NACK in transmit mode</li> </ul> Clearing conditions: <ul style="list-style-type: none"> <li>Receive ACK in transmit mode</li> <li>Communication reset</li> <li>Internal state reset</li> </ul> | RO     |
| 9   | ARLOF    | Arbitration failure flag bit | <ul style="list-style-type: none"> <li>0b0: No arbitration failure occurred</li> <li>0b1: Arbitration failure</li> </ul> Setting conditions: <ul style="list-style-type: none"> <li>Failure to arbitrate</li> </ul> Clearing conditions: <ul style="list-style-type: none"> <li>ARLOFCLR writes "1"</li> <li>Communication reset</li> <li>Internal state reset</li> </ul>                                                                    | RO     |
| 8   | Reserved | Reserved                     | -                                                                                                                                                                                                                                                                                                                                                                                                                                            | -      |

| Bit | Name            | Function                     | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | Access |
|-----|-----------------|------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 7   | EMPTYF          | Transmit data empty flag bit | <ul style="list-style-type: none"> <li>0b0: I2C_DTR register full</li> <li>0b1: I2C_DTR register empty</li> </ul> Setting conditions: <ul style="list-style-type: none"> <li>I2C_DTR data is transferred to the internal shift register</li> <li>TRA bit set to 1</li> </ul> Clearing Conditions: <ul style="list-style-type: none"> <li>Write I2C_DTR</li> <li>TRA bit clear 0</li> <li>Communication reset</li> <li>Internal state reset</li> </ul>                                                                                           | RO     |
| 6   | RFULLF          | Receive data full flag bit   | <ul style="list-style-type: none"> <li>0b0: I2C_DRR register empty</li> <li>0b1: I2C_DRR register full</li> </ul> Setting conditions: <ul style="list-style-type: none"> <li>The received data is transferred from the internal shift register to I2C_DRR</li> </ul> Clearing conditions: <ul style="list-style-type: none"> <li>Read I2C_DRR</li> <li>RFULLFCLR writes "1"</li> <li>Communication reset</li> <li>Internal state reset</li> </ul>                                                                                               | RO     |
| 5   | <i>Reserved</i> | <i>Reserved</i>              | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | -      |
| 4   | STOPF           | Stop condition flag bit      | <ul style="list-style-type: none"> <li>0b0: No STOP condition has been detected on the bus.</li> <li>0b1: A STOP condition has been detected on the bus.</li> </ul> Setting conditions: <ul style="list-style-type: none"> <li>Stop condition detected</li> </ul> Clearing conditions: <ul style="list-style-type: none"> <li>STOPFCLR writes "1"</li> <li>Communication reset</li> <li>Internal state reset</li> </ul>                                                                                                                         | RO     |
| 3   | TENDF           | Transmit data end flag bit   | <ul style="list-style-type: none"> <li>0b0: The internal shift register is still transmitting data.</li> <li>0b1: The internal shift register has finished transmitting data.</li> </ul> Setting conditions: <ul style="list-style-type: none"> <li>Under the condition of EMPTYF=1, the 9th rising edge of SCL is "1"</li> </ul> Clearing conditions: <ul style="list-style-type: none"> <li>Stop condition detected</li> <li>Write I2C_DTR</li> <li>TENDFCLR writes "1"</li> <li>Communication reset</li> <li>Internal state reset</li> </ul> | RO     |
| 2   | SLADDR1F        | Slave address register 1     | <ul style="list-style-type: none"> <li>0b0: No consistent address was detected in the slave address register 1.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                      | RO     |

| Bit | Name     | Function                                              | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | Access |
|-----|----------|-------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 2   | SLADDR1F | matching consistent mark bit                          | <ul style="list-style-type: none"> <li>0b1: A consistent address is detected in the slave address register 1.</li> </ul> <p>Setting conditions:</p> <ul style="list-style-type: none"> <li>When I2C_SLR1.ADDRM0D1=0, the received slave address is matched with the I2C_SLR1.SLADDR1[7:1].</li> <li>When I2C_SLR1.ADDRM0D1=1, the first byte address of the received 10-bit slave address matches 0b11110+I2C_SLR1.SLADDR1[9:8] and the second byte address matches I2C_SLR1.SLADDR1[7:0]</li> </ul> <p>Clearing conditions:</p> <ul style="list-style-type: none"> <li>Stop condition detected</li> <li>SLADDR1FCLR writes "1"</li> <li>Communication reset</li> <li>Internal state reset</li> </ul>                                                                                         | RO     |
| 1   | SLADDR0F | Slave address register 0 matching consistent mark bit | <ul style="list-style-type: none"> <li>0b0: No consistent address was detected in the slave address register 0.</li> <li>0b1: A consistent address is detected in the slave address register 0.</li> </ul> <p>Setting conditions:</p> <ul style="list-style-type: none"> <li>When I2C_SLR0.ADDRM0D0=0, the received slave address is compared with the I2C_SLR0.SLADDR0[7:1].</li> <li>When I2C_SLR0.ADDRM0D0=1, the first byte address of the received 10-bit slave address matches 0b11110 + I2C_SLR0.SLADDR0 [9: 8] and the second byte address matches I2C_SL0.SLADDR0 [7: 0]</li> </ul> <p>Clearing conditions:</p> <ul style="list-style-type: none"> <li>Stop condition detected</li> <li>SLADDR0FCLR writes "1"</li> <li>Communication reset</li> <li>Internal state reset</li> </ul> | RO     |
| 0   | STARTF   | Start condition/restart condition flag bit            | <ul style="list-style-type: none"> <li>0b0: Bus not detected starting condition</li> <li>0b0: Bus detected starting condition</li> </ul> <p>Setting conditions:</p> <ul style="list-style-type: none"> <li>Start condition detected</li> </ul> <p>Clearing conditions:</p> <ul style="list-style-type: none"> <li>Stop condition detected</li> <li>STARTFCLR writes "1"</li> <li>Communication reset</li> <li>Internal state reset</li> </ul>                                                                                                                                                                                                                                                                                                                                                 | RO     |

## 30.5.10 I2C Status Clear Register (I2C\_CLR)

| Offset | Bit Position |    |    |    |    |    |    |    |             |    |    |    |             |    |    |    |                |    |    |    |             |    |    |    |          |   |   |   |   |   |   |   |           |    |    |    |          |    |    |    |          |    |    |    |          |    |    |    |          |    |    |    |          |    |    |    |          |  |  |  |  |  |  |  |           |    |    |    |          |    |    |    |          |    |    |    |          |    |    |    |             |    |    |    |             |    |    |    |           |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|--------|--------------|----|----|----|----|----|----|----|-------------|----|----|----|-------------|----|----|----|----------------|----|----|----|-------------|----|----|----|----------|---|---|---|---|---|---|---|-----------|----|----|----|----------|----|----|----|----------|----|----|----|----------|----|----|----|----------|----|----|----|----------|----|----|----|----------|--|--|--|--|--|--|--|-----------|----|----|----|----------|----|----|----|----------|----|----|----|----------|----|----|----|-------------|----|----|----|-------------|----|----|----|-----------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| 0x20   | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23          | 22 | 21 | 20 | 19          | 18 | 17 | 16 | 15             | 14 | 13 | 12 | 11          | 10 | 9  | 8  | 7        | 6 | 5 | 4 | 3 | 2 | 1 | 0 |           |    |    |    |          |    |    |    |          |    |    |    |          |    |    |    |          |    |    |    |          |    |    |    |          |  |  |  |  |  |  |  |           |    |    |    |          |    |    |    |          |    |    |    |          |    |    |    |             |    |    |    |             |    |    |    |           |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |             |    |    |    |             |    |    |    |                |    |    |    |             |    |    |    |          |   |   |   |   |   |   |   |           |    |    |    |          |    |    |    |          |    |    |    |          |    |    |    |          |    |    |    |          |    |    |    |          |  |  |  |  |  |  |  |           |    |    |    |          |    |    |    |          |    |    |    |          |    |    |    |             |    |    |    |             |    |    |    |           |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Name   | Reserved     |    |    |    |    |    |    |    | SMBALRTFCLR |    |    |    | SMBHOSTFCLR |    |    |    | SMBDEFAULTFCLR |    |    |    | GENCALLFCLR |    |    |    | Reserved |   |   |   |   |   |   |   | TMOUTFCLR |    |    |    | Reserved |    |    |    | NACKFCLR |    |    |    | Reserved |    |    |    | RFREQCLR |    |    |    | ARLOFCLR |    |    |    | Reserved |  |  |  |  |  |  |  | RFULLFCLR |    |    |    | Reserved |    |    |    | STOPFCLR |    |    |    | TENDFCLR |    |    |    | SLADDR1FCLR |    |    |    | SLADDR0FCLR |    |    |    | STARTFCLR |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Access |              |    |    |    |    |    |    |    | WO          | WO | WO | WO | WO          | WO | WO | WO | WO             | WO | WO | WO | WO          | WO | WO | WO |          |   |   |   |   |   |   |   | WO        | WO | WO | WO | WO       | WO | WO | WO | WO       | WO | WO | WO | WO       | WO | WO | WO | WO       | WO | WO | WO | WO       | WO | WO | WO |          |  |  |  |  |  |  |  | WO        | WO | WO | WO | WO       | WO | WO | WO | WO       | WO | WO | WO | WO       | WO | WO | WO | WO          | WO | WO | WO | WO          | WO | WO | WO | WO        | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO |

| Bit   | Name           | Function                                                    | Description                                                       | Access |
|-------|----------------|-------------------------------------------------------------|-------------------------------------------------------------------|--------|
| 31:24 | Reserved       | Reserved                                                    | -                                                                 | -      |
| 23    | SMBALRTFCLR    | SMBUS alarm response address matching consistent flag clear | Write 1 to clear the SMBALRTF flag bit, write 0 has no effect.    | WO     |
| 22    | SMBHOSTFCLR    | SMBUS master address matching consistent flag clear         | Write 1 to clear the SMBHOSTF flag bit, write 0 has no effect.    | WO     |
| 21    | SMBDEFAULTFCLR | SMBUS default address matching consistent flag clear        | Write 1 to clear the SMBDEFAULTF flag bit, write 0 has no effect. | WO     |
| 20    | GENCALLFCLR    | Broadcast call address matching flag clear                  | Write 1 to clear the GENCALLF flag bit, write 0 has no effect.    | WO     |
| 19:15 | Reserved       | Reserved                                                    | -                                                                 | -      |
| 14    | TMOUTFCLR      | Timeout flag clear                                          | Write 1 to clear the TMOUTF flag bit, write 0 has no effect.      | WO     |
| 13    | Reserved       | Reserved                                                    | -                                                                 | -      |
| 12    | NACKFCLR       | NACK flag clear                                             | Write 1 to clear the NACKF flag bit, write 0 has no effect.       | WO     |
| 11    | Reserved       | Reserved                                                    | -                                                                 | -      |
| 10    | RFREQCLR       | Receive FIFO data request flag clear                        | Write 1 to clear the RFREQF flag bit, write 0 has no effect.      | WO     |
| 9     | ARLOFCLR       | Arbitration failure flag clear                              | Write 1 to clear the ARLOF flag bit, write 0 has no effect.       | WO     |
| 8:7   | Reserved       | Reserved                                                    | -                                                                 | -      |
| 6     | RFULLFCLR      | Receive data full flag clear                                | Write 1 to clear the RFULLF flag bit, write 0 has no effect.      | WO     |
| 5     | Reserved       | Reserved                                                    | -                                                                 | -      |

| Bit | Name        | Function                                                | Description                                                   | Access |
|-----|-------------|---------------------------------------------------------|---------------------------------------------------------------|--------|
| 4   | STOPFCLR    | Stop condition flag clear                               | Write 1 to clear the STOPF flag bit,write 0 has no effect.    | WO     |
| 3   | TENDFCLR    | Transmit data end flag clear                            | Write 1 to clear the TENDF flag bit,write 0 has no effect.    | WO     |
| 2   | SLADDR1FCLR | Slave address register 1 matching consistent flag clear | Write 1 to clear the SLADDR1F flag bit,write 0 has no effect. | WO     |
| 1   | SLADDR0FCLR | Slave address register 0 matching consistent flag clear | Write 1 to clear the SLADDR0F flag bit,write 0 has no effect. | WO     |
| 0   | STARTFCLR   | Start condition/re-start condition flag clear           | Write 1 to clear the STARTF flag bit,write 0 has no effect.   | WO     |

### 30.5.11 I2C Data Transmitting Register (I2C\_DTR)

| Offset | Bit Position |   |   |   |   |   |   |   |
|--------|--------------|---|---|---|---|---|---|---|
| 0x24   | 7            | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0xFF         |   |   |   |   |   |   |   |
| Name   | DT           |   |   |   |   |   |   |   |
| Access | RW           |   |   |   |   |   |   |   |

| Bit | Name | Function      | Description                                                                                                                                                                                           | Access |
|-----|------|---------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 7:0 | DT   | Transmit data | If the internal shift register is empty, the data written to the I2C_DTR register will be transferred to the internal shift register. In transmit mode, the data transmission to the SDA line begins. | RW     |

The internal shift register and the I2C\_DTR register form a double-buffered structure. During the data transmission of the internal shift register, if data is written in advance to the I2C\_DTR register, continuous data transmission can be achieved.

The I2C\_DTR register is readable and writable. Write I2C\_DTR register only once when the transmit data is empty interrupt request occurs.

When the I2C operates in FIFO mode, accessing the I2C\_DTR register address allows direct access to the transmit FIFO. If the current EMPTYF flag is enabled, a write operation to the I2C\_DTR register address writes the current bus data into the transmit FIFO. When the transmit FIFO is enable and the I2C is enabled to transmit, data from the transmit FIFO is transferred to the internal shift register, and data transmission to the SDA line begins.

The selection of the transmit data source is determined by the FEN bit in the FIFO Status Register (I2C\_FSTR) in I2C :

- When FEN is 1, the path from the external data bus to the FIFO and from the FIFO to the internal shift register is selected.
- When FEN is 0, the path from the external data bus to the I2C\_DTR register and from the I2C\_DTR register to the internal shift register is selected.

By default, the non-FIFO mode is selected.

### 30.5.12 I2C Data Receiving Register (I2C\_DRR)

| Offset | Bit Position |   |   |   |   |   |   |   |
|--------|--------------|---|---|---|---|---|---|---|
| 0x28   | 7            | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00         |   |   |   |   |   |   |   |
| Name   | DR           |   |   |   |   |   |   |   |
| Access | RO           |   |   |   |   |   |   |   |

| Bit | Name | Function     | Description                                                                                                                                                                | Access |
|-----|------|--------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 7:0 | DR   | Receive Data | If a frame of data is received, the received data can be transferred from the internal shift register to the I2C_DRR register, and then into the next data receiving state | RO     |

The internal shift register and the I2C\_DRR register form a double-buffered structure. During data reception in the internal shift register, if the data in the I2C\_DRR register is read, continuous data reception can be achieved.

Do not write to I2C\_DRR register. Read I2C\_DRR register only once when the receive data full interrupt request occurs.

When the I2C\_SR.RFULLF flag is set to 1 and the data in the I2C\_DRR register is not read before the next data is received, the SCL clock will be automatically held low after the 7th or 8th falling edge during data reception (refer to the [slave receive timing diagram](#) and [the fast ACK/NACK timing diagram](#)), until the data in the I2C\_DRR register is read.

When the I2C operates in FIFO mode, accessing the I2C\_DRR register address directly accesses the receive FIFO:

- If one frame of data is received, the received data will be transferred from the internal shift register to the receive FIFO. Once the receive FIFO has received active data, an RFREQ request status signal will be asserted. At this point, a read operation to the I2C\_DRR register address directly accesses the data in the receive FIFO.
- When the I2C\_SR.RFULLF flag is set to 1 (i.e., the receive FIFO has stored 2 bytes of data and FIFO is full), and if the data in the receive FIFO is not read before the next data is received, the SCL clock will be automatically held low after the 7th or 8th falling edge of the received data, until the data in the receive FIFO is read.

The receive data path of the I2C is selected through the FEN bit in the I2C FIFO Status Register (I2C\_FSTR) :

- When FEN is 1, the data path from the internal shift register to the receive FIFO and from the receive FIFO to the data bus is selected.
- When FEN is 0, the data path from the internal shift register to the I2C\_DRR register and from the I2C\_DRR register to the data bus is selected.

By default, the non-FIFO mode is selected.

## 30.5.13 I2C Clock Control Register (I2C\_CCR)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |       |    |        |    |    |    |    |    |    |       |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|-------|----|--------|----|----|----|----|----|----|-------|---|---|---|---|---|---|---|---|---|
| 0x2C   | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18    | 17 | 16     | 15 | 14 | 13 | 12 | 11 | 10 | 9     | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00001F1F   |    |    |    |    |    |    |    |    |    |    |    |    |       |    |        |    |    |    |    |    |    |       |   |   |   |   |   |   |   |   |   |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    | CKDIV |    | SHIGHW |    |    |    |    |    |    | SLOWW |   |   |   |   |   |   |   |   |   |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    | RW    |    | RW     |    |    |    |    |    |    | RW    |   |   |   |   |   |   |   |   |   |

| Bit   | Name     | Function                                     | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | Access |
|-------|----------|----------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:19 | Reserved | Reserved                                     | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | -      |
| 18:16 | CKDIV    | Setting bit of I2C reference clock frequency | <ul style="list-style-type: none"> <li>0b000: I2C Reference Clock Frequency=PCLK3</li> <li>0b001: I2C Reference Clock Frequency=PCLK3/2</li> <li>0b010: I2C Reference Clock Frequency=PCLK3/4</li> <li>0b011: I2C Reference Clock Frequency=PCLK3/8</li> <li>0b100: I2C Reference Clock Frequency=PCLK3/16</li> <li>0b101: I2C Reference Clock Frequency=PCLK3/32</li> <li>0b110: I2C Reference Clock Frequency=PCLK3/64</li> <li>0b111: I2C Reference Clock Frequency=PCLK3/128</li> </ul> | RW     |
| 15:8  | SHIGHW   | Set SCL high level width bit                 | Setting the high level width of SCL clock                                                                                                                                                                                                                                                                                                                                                                                                                                                   | RW     |
| 7:0   | SLOWW    | Set SCL low level width bit                  | Setting the low level width of SCL clock                                                                                                                                                                                                                                                                                                                                                                                                                                                    | RW     |

The baud rate is calculated as follows:

| I2C_FLTR.DNFE<br>N | I2C_CCR.CKDIV | Baud rate                                                                             |
|--------------------|---------------|---------------------------------------------------------------------------------------|
| 0b0                | 0b000         | $B = \frac{1}{\frac{(SHIGHW + 3) + (SLOWW + 3)}{\phi_{I2C}} + t_R + t_F}$             |
| 0b1                | 0b000         | $B = \frac{1}{\frac{(SHIGHW + 3 + DNF) + (SLOWW + 3 + DNF)}{\phi_{I2C}} + t_R + t_F}$ |
| 0b0                | Non-0         | $B = \frac{1}{\frac{(SHIGHW + 2) + (SLOWW + 2)}{\phi_{I2C}} + t_R + t_F}$             |
| 0b1                | Non-0         | $B = \frac{1}{\frac{(SHIGHW + 2 + DNF) + (SLOWW + 2 + DNF)}{\phi_{I2C}} + t_R + t_F}$ |



Where:

- B: Baud rate, unit: Mbps
- $\Phi I2C$ : I2C reference clock frequency, unit: MHz
- DNF: Digital filter capability, set by the I2C\_FLTR.DNF register
- $t_R$ : Rise time of the SCL line, unit: ns
- $t_F$ : Fall time of the SCL line, unit: ns



#### Note

- In master mode, SHIGHW is used to set the high level width of the SCL clock. The setting has no effect in slave mode.
- In master mode, SLOWW is used to set the low level width of the SCL clock. In slave mode, the set value must be greater than the data setup time ( $t_{SU,DAT}$ ), which is 250ns for standard mode and 100ns for fast mode.
- The rise time ( $t_R$ ) and fall time ( $t_F$ ) of the SCL line depend on the total bus capacitance ( $C_b$ ) and pull-up resistance ( $R_p$ ). For detailed information, refer to the I2C bus specification.

### 30.5.14 I2C Filter Control Register (I2C\_FLTR)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |      |   |      |   |          |   |     |  |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|------|---|------|---|----------|---|-----|--|
| 0x30   | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5    | 4 | 3    | 2 | 1        | 0 |     |  |
| Reset  | 0x00000010   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |      |   |      |   |          |   |     |  |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   | ANFN |   | DNFN |   | Reserved |   | DNF |  |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   | RW   |   | RW   |   |          |   | RW  |  |

| Bit  | Name     | Function                                                | Description                                                                                                                                                                                                                                                                                                          | Access |
|------|----------|---------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:6 | Reserved | Reserved                                                | -                                                                                                                                                                                                                                                                                                                    | -      |
| 5    | ANFEN    | Analog filtering function allowed bit                   | <ul style="list-style-type: none"> <li>● 0b0: Analog filter function disabled</li> <li>● 0b1: Analog filter function enabled</li> </ul>                                                                                                                                                                              | RW     |
| 4    | DNFEN    | Digital filtering function allowed bit                  | <ul style="list-style-type: none"> <li>● 0b0: Digital filter function disabled</li> <li>● 0b1: Digital filter function enabled</li> </ul>                                                                                                                                                                            | RW     |
| 3:2  | Reserved | Reserved                                                | -                                                                                                                                                                                                                                                                                                                    | -      |
| 1:0  | DNF      | Selection of filtering capabilities for digital filters | <ul style="list-style-type: none"> <li>● 0b00: Filtering capability 1 I2C reference clock period</li> <li>● 0b01: Filtering capability 2 I2C reference clock period</li> <li>● 0b10: Filtering capability 3 I2C reference clock period</li> <li>● 0b11: Filtering capability 4 I2C reference clock period</li> </ul> | RW     |

## 30.5.15 I2C FIFO Status Register (I2C\_FSTR)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |    |    |      |      |    |              |        |        |     |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|----|----|------|------|----|--------------|--------|--------|-----|
| 0x34   | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7  | 6  | 5    | 4    | 3  | 2            | 1      | 0      |     |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |    |    |      |      |    |              |        |        |     |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |    |    | RFST | TFST |    | NACK_TFFLUSH | RFLUSH | TFLUSH | FEN |
| RW     |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |    |    |      |      |    | RW           | RW     | RW     |     |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   | RO | RO | RW   | RW   | RW | RW           |        |        |     |

| Bit  | Name         | Function                  | Description                                                                                                                                                                               | Access |
|------|--------------|---------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:8 | Reserved     | Reserved                  | -                                                                                                                                                                                         | -      |
| 7:6  | RFST         | Receive FIFO status bit   | <ul style="list-style-type: none"> <li>0b00: FIFO empty</li> <li>0b01: FIFO half full</li> <li>0b10: FIFO full</li> <li>0b11: Reserved</li> </ul>                                         | RO     |
| 5:4  | TFST         | Transmit FIFO status bit  | <ul style="list-style-type: none"> <li>0b00: FIFO empty</li> <li>0b01: FIFO half empty</li> <li>0b10: FIFO full</li> <li>0b11: Reserved</li> </ul>                                        | RO     |
| 3    | NACK_TFFLUSH | NACK resets transmit FIFO | <ul style="list-style-type: none"> <li>0b0: Do not execute TFIFO FLUSH after receiving NACK in slave mode</li> <li>0b1: Execute TFIFO FLUSH after receiving NACK in slave mode</li> </ul> | RW     |
| 2    | RFLUSH       | RX FIFO pointer reset     | <ul style="list-style-type: none"> <li>0b0: Write disabled</li> <li>0b1: Reset RX FIFO pointer</li> </ul>                                                                                 | RW     |
| 1    | TFLUSH       | TX FIFO pointer reset     | <ul style="list-style-type: none"> <li>0b0: Write disabled</li> <li>0b1: Reset TX FIFO pointer</li> </ul>                                                                                 | RW     |
| 0    | FEN          | FIFO function enable bit  | <ul style="list-style-type: none"> <li>0b0: FIFO function disabled</li> <li>0b1: FIFO function enabled</li> </ul>                                                                         | RW     |

## 30.5.16 I2C Slave Address Receive Register (I2C\_SLVADRR)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |        |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|--------|---|---|---|---|---|---|---|---|---|
| 0x38   | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9      | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |        |   |   |   |   |   |   |   |   |   |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | SLVADR |   |   |   |   |   |   |   |   |   |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RO     |   |   |   |   |   |   |   |   |   |

| Bit   | Name     | Function                       | Description                                                                           | Access |
|-------|----------|--------------------------------|---------------------------------------------------------------------------------------|--------|
| 31:10 | Reserved | Reserved                       | -                                                                                     | -      |
| 9:0   | SLVADR   | Slave address receive register | SLVADR[6:0]: 7-bit slave receive address<br>SLVADR[9:0]: 10-bit slave receive address | RO     |

## 31 Serial Peripheral Interface (SPI)

### 31.1 Introduction

This product is equipped with 6-channel serial peripheral interface SPI, supports high-speed full-duplex serial synchronous transmission, and facilitates data exchange with peripheral devices. Users can set the range of 3/4-wire, master/slave and baud rate as as required.

### 31.2 Features

Table 31-1 SPI Main Features

| Essentials                          | Description                                                                                                                                                                                                                                                                                                                                                                         |
|-------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Serial Communication Function       | <ul style="list-style-type: none"> <li>● Supporting 4-wire SPI mode and 3-wire clock synchronization mode</li> <li>● Supports full-duplex synchronous communication mode and transmit-only communication mode</li> <li>● Polarity and phase of adjustable communication clock SCK</li> </ul>                                                                                        |
| Data Format                         | <ul style="list-style-type: none"> <li>● Selectable data shift order: MSB start/LSB start</li> <li>● Selectable data width: 4/5/6/7/8/9/10/11/12/13/14/15/16/20/24/32 bits</li> <li>● A maximum of 4 frames width of 32-bit data can be transmitted or received at a single time</li> </ul>                                                                                         |
| Baud Rate                           | <ul style="list-style-type: none"> <li>● The baud rate can be adjusted by the built-in special baud rate generator in master mode. The baud rate ranges from 2 to 256 frequency division of PCLK1.</li> <li>● Maximum baud rate allowed in slave mode is 6 frequency division of PCLK1</li> </ul>                                                                                   |
| Data Buffer                         | <ul style="list-style-type: none"> <li>● With 16-byte data buffer</li> <li>● Supports double buffering</li> </ul>                                                                                                                                                                                                                                                                   |
| Error Monitoring                    | <ul style="list-style-type: none"> <li>● Mode fault error monitoring</li> <li>● Data overload error monitoring</li> <li>● Data underload error monitoring</li> <li>● Parity error monitoring</li> </ul>                                                                                                                                                                             |
| Chip Selection Signal Control       | <ul style="list-style-type: none"> <li>● Each channel is configured with 4 chip-selected signal line</li> <li>● The relative timing relationship between the chip select signal and the communication clock can be adjusted</li> <li>● The inactive time of the chip select signal between two consecutive communications can be adjusted</li> <li>● Polarity adjustable</li> </ul> |
| Transmission Control in Master Mode | <ul style="list-style-type: none"> <li>● Start transmission by writing data to the data register</li> <li>● Communication auto suspend function</li> </ul>                                                                                                                                                                                                                          |

| Essentials        | Description                                                                                                                                                                                                                           |
|-------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Interrupt         | <ul style="list-style-type: none"><li>● Receive buffer is full</li><li>● Transmit buffer is empty</li><li>● SPI error (mode/overload/underload/parity)</li><li>● SPI vacant</li><li>● Transmit complete (event source only)</li></ul> |
| Low Power Control | <ul style="list-style-type: none"><li>● Configurable module stop</li></ul>                                                                                                                                                            |
| Other Functions   | <ul style="list-style-type: none"><li>● SPI initialization function</li></ul>                                                                                                                                                         |

**Note**

When the communication auto-suspend feature is used in master receive mode, overload errors will not occur because the communication clock stops. Refer to "[Overload Error](#)" for more information.

## 31.3 Functional Description

### 31.3.1 System Block Diagram

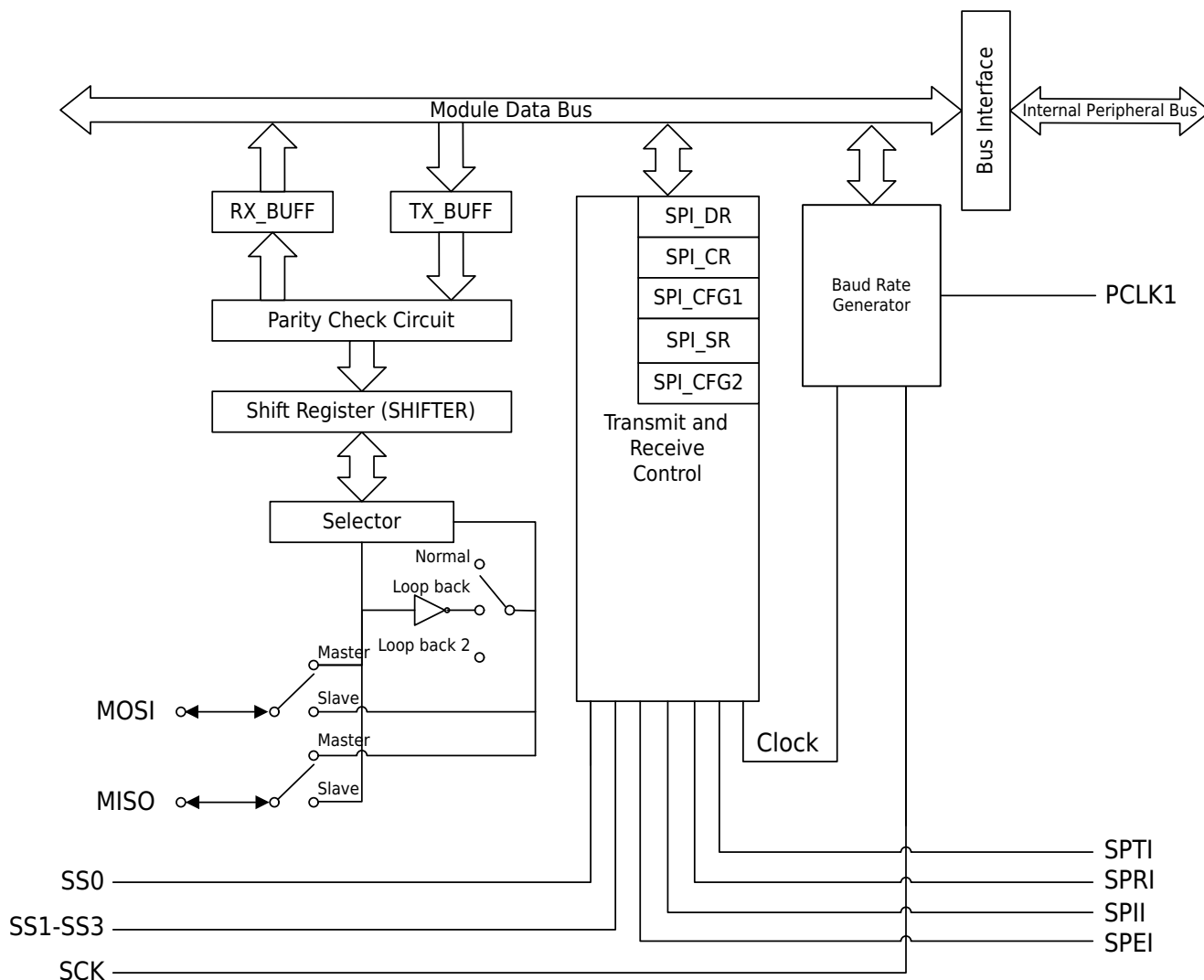


Figure 31-1 SPI System Block Diagram

Table 31-2 Pin Description

| Pin Name | Port direction | Function                      |
|----------|----------------|-------------------------------|
| SCK      | Input/Output   | Communication clock pin       |
| MOSI     | Input/Output   | Master data transmission pin  |
| MISO     | Input/Output   | Slave data transmission pin   |
| SS0      | Input/Output   | Slave select input/output pin |
| SS1      | Output         | Slave select output pin       |
| SS2      | Output         | Slave select output pin       |

| Pin Name | Port direction | Function                |
|----------|----------------|-------------------------|
| SS3      | Output         | Slave select output pin |

### 31.3.2 SPI Action System Description

#### 31.3.2.1 Pin Status in Master Mode

When SPI works in master mode, the status of each pin is shown below.

Table 31-3 SPI Pin Status Description in Master Mode

| Mode                                       |                               | Pin Name           | Pin Status                      |
|--------------------------------------------|-------------------------------|--------------------|---------------------------------|
| SPI Action (SPIMDS=0)                      | Master Mode (MSTR=1, MODFE=0) | SCK                | Output                          |
|                                            |                               | SS0-SS3            | Output                          |
|                                            |                               | MOSI               | Output                          |
|                                            |                               | MISO               | Input                           |
| Clock Synchronization Operation (SPIMDS=1) | Master Mode (MSTR=1)          | SCK                | Output                          |
|                                            |                               | SS0-SS3 (not used) | Hi-Z (available as general I/O) |
|                                            |                               | MOSI               | Output                          |
|                                            |                               | MISO               | Input                           |



#### Note

Please set the pin input type to CMOS input and the output type to high drive mode. For related settings, please refer to "General-Purpose IO (GPIO) " "Register-General Control Register (PCRxy) ".

#### 31.3.2.2 Pin Status in Slave Mode

When SPI works in slave mode, the status of each pin is shown below.

Table 31-4 SPI Pin Status Description in Slave Mode

| Mode                                       |                              | Pin Name            | Pin Status                      |
|--------------------------------------------|------------------------------|---------------------|---------------------------------|
| SPI Action (SPIMDS=0)                      | Slave Mode (MSTR=0, MODFE=0) | SCK                 | Input                           |
|                                            |                              | SS0                 | Input                           |
|                                            |                              | SS1- SS3 (not used) | Hi-Z (available as general I/O) |
|                                            |                              | MOSI                | Input                           |
|                                            |                              | MISO                | Output/Hi-Z                     |
| Clock Synchronization Operation (SPIMDS=1) | Slave Mode (MSTR=0)          | SCK                 | Input                           |
|                                            |                              | SS0-SS3 (not used)  | Hi-Z (available as general I/O) |
|                                            |                              | MOSI                | Input                           |
|                                            |                              | MISO                | Output                          |

**Note**

Please set the pin input type to CMOS input and the output type to high drive mode. For related settings, please refer to "General-Purpose IO (GPIO) " "Register-General Control Register (PCRxy) ".

### 31.3.2.3 SPI System Connection Examples

#### Master Mode

In the master-multi-slave mode SPI system architecture, the master drives SCK, MOSI, and SS0-SS3. In SPI slave equipment 0-3, the slave equipment drives the MISO when the SS input of a slave computer is a active level.

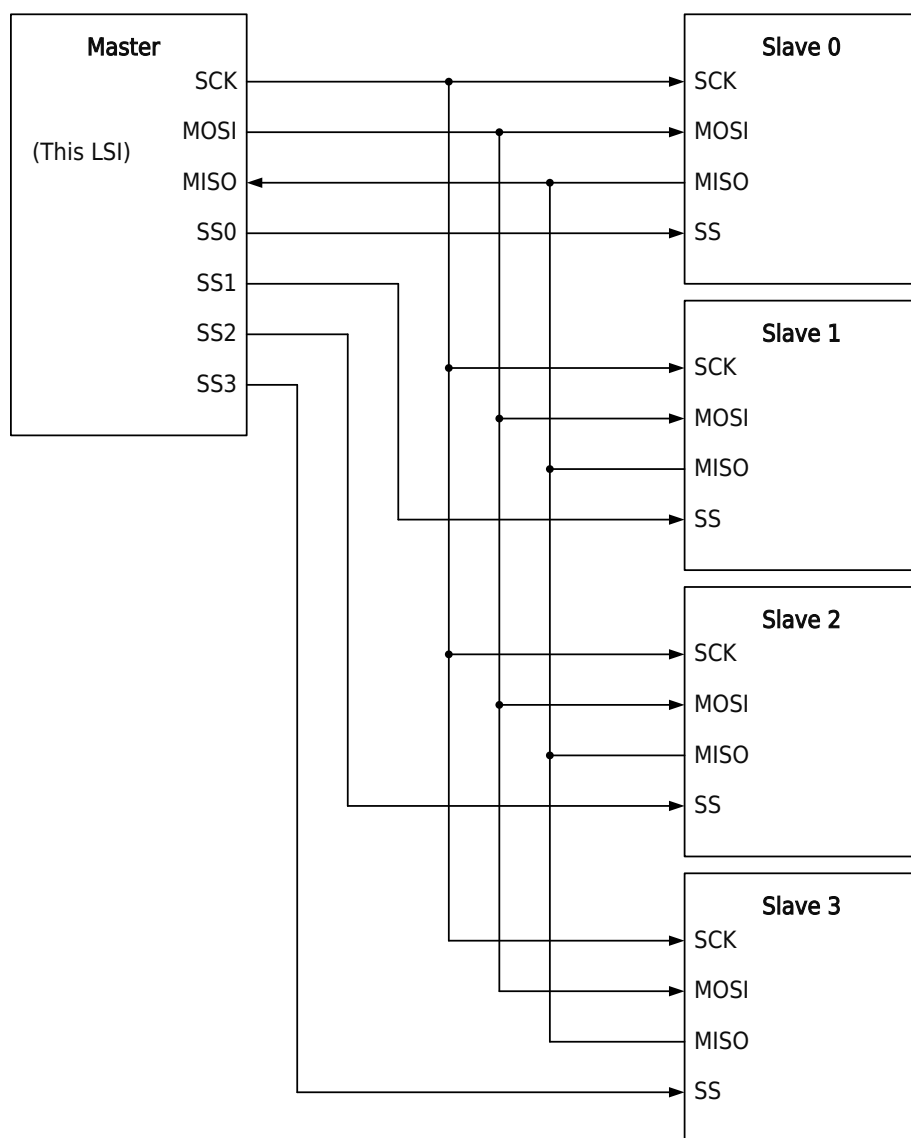


Figure 31-2 Master Mode Structure

#### Clock Synchronization Operation

In the SPI system structure used for clock synchronous operation, the master device drives SCK and MOSI, and the slave device drives MISO. SS pins are not used.



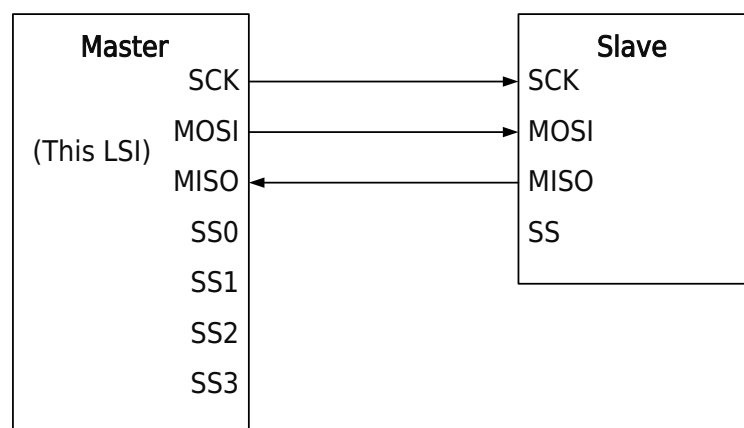


Figure 31-3 Three-wire Clock Synchronization Operation

### 31.3.3 Data Communication Description

#### 31.3.3.1 Baud Rate

In master mode, SPI clock is provided by internal baud rate generator. In slave mode, the clock is input to the SCK pin.

Baud rate depends on SPI\_CFG2.MBR bit setting. The calculation method is shown in the following formula:

$$\text{Baud Rate} = \frac{f_{\text{PCLK1}}}{2^{N+1} \times (M+1)}$$

Where:

- N: SPI\_CFG2.MBR bit setting value, the range is from 0-3;
- M: SPI\_CFG1.CLKDIV bit setting value, the range is from 0-15.

Table 31-5 Bit Speed of Some Set Values

| MBR Bit Setting Value (N) | CLKDIV Bit Setting Value (M) | Frequency Dividing Ratio | Baud Rate  |             |             |             |
|---------------------------|------------------------------|--------------------------|------------|-------------|-------------|-------------|
|                           |                              |                          | PCLK1=5MHz | PCLK1=10MHz | PCLK1=20MHz | PCLK1=40MHz |
| 0                         | 0                            | 2                        | 2.50Mbps   | 5.00Mbps    | 10.0Mbps    | 20.0Mbps    |
| 0                         | 1                            | 4                        | 1.25Mbps   | 2.50Mbps    | 5.00Mbps    | 10.0Mbps    |
| 0                         | 3                            | 8                        | 625kbps    | 1.25Mbps    | 2.50Mbps    | 5.00Mbps    |
| 0                         | 7                            | 16                       | 313kbps    | 625kbps     | 1.25Mbps    | 2.50Mbps    |
| 0                         | 15                           | 32                       | 156kbps    | 313kbps     | 625kbps     | 1.25Mbps    |
| 1                         | 15                           | 64                       | 78kbps     | 156kbps     | 313kbps     | 625kbps     |
| 2                         | 15                           | 128                      | 39kbps     | 78kbps      | 156kbps     | 313kbps     |
| 3                         | 15                           | 256                      | 20kbps     | 39kbps      | 78kbps      | 156kbps     |

#### 31.3.3.2 Data Format

The data format of SPI depends on the setting value of SPI\_CFG2 register and SPI\_CR.PAE bit setting value. SPI treats the data of a certain length starting from the LSB bit in the data register SPI\_DR (the data length is set by the SPI\_CFG2.DSIZE) as the transfer object, regardless of the MSB/LSB shift order.

SPI\_CFG2.DSIZE determines the bit width of the data. The bit width range is 4-32 bits. SPI\_CR.PAE determines the last bit of the data. When PAE is 1, the last bit is used as the parity bit. When it is 0, it is the lowest bit of the data itself.

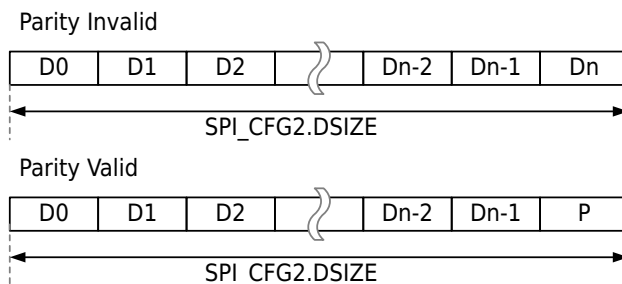


Figure 31-4 Data Format

When sending SPI data, the transmitted data first enters the transmit buffer (TX\_BUFF), and then the data in the TX\_BUFF is copied to the SHIFTER register. The SHIFTER sequentially outputs the data. During SPI data reception, data is sequentially shifted in from the SHIFTER. Once the shifting is complete, the data in the SHIFTER is copied to the receive buffer (RX\_BUFF).

During data transmission, there are four situations according to the settings of the shift order control bit SPI\_CFG2.LSBF and the parity control bit SPI\_CR.PAE:

- **MSB Transmitted First, Parity Invalid**

During transmitting, data D31-D0 is copied sequentially from TX\_BUFF to the SHIFTER and shifted out from the highest bit of the SHIFTER in the sequence of D31-D0.

During receiving, data D31-D0 is shifted in from the least significant bit of the SHIFTER and copied to RX\_BUFF after all bits have been received.

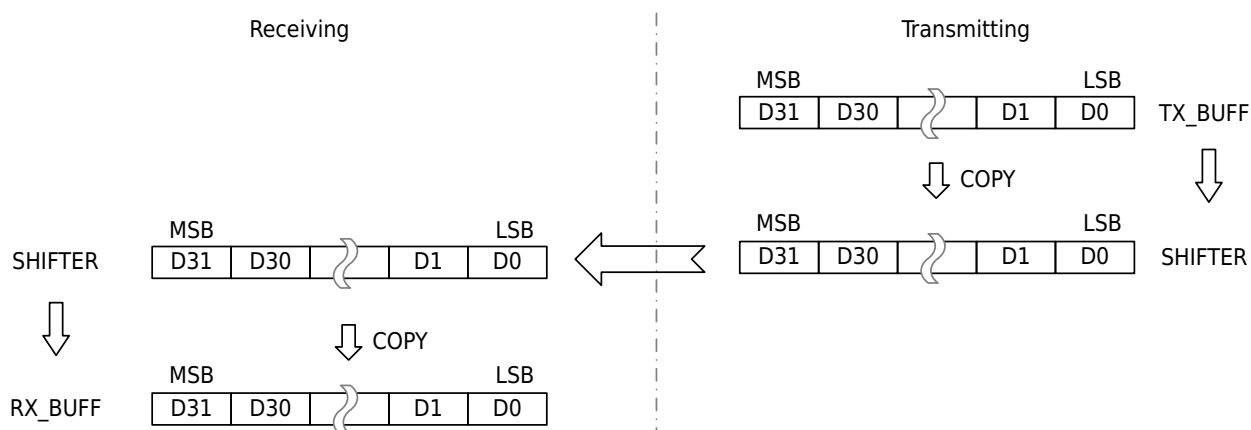


Figure 31-5 MSB Transmitted First, Parity Invalid

- **LSB Transmitted First, Parity Invalid**

During transmitting, D31-D0 is copied from TX\_BUFF to SHIFTER in the sequence of D0-D31, and shifted out from the highest bit of the SHIFTER in the sequence of D0-D31.

During receiving, data D0-D31 is moved in from the lowest bit of SHIFTER, and then copied from SHIFTER to RX\_BUFF in the order D31-D0 after all data is moved in.

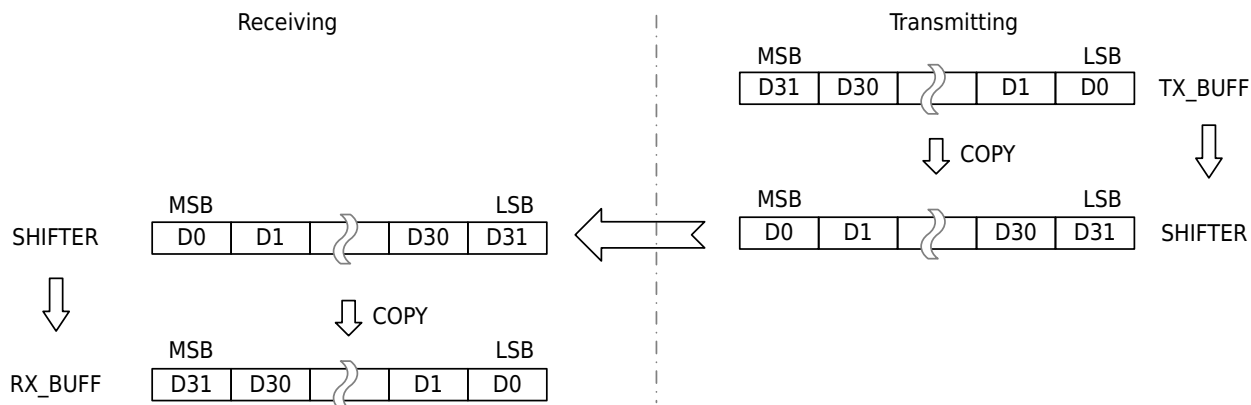


Figure 31-6 LSB Transmitted First, Parity Invalid

### ● MSB Transmitted First, Parity Valid

During transmitting, the value of parity bit P is calculated according to the value of D31-D1, then P is replaced by D0, copied to SHIFTER according to the order of D31-D1, P, and shifted out from the highest bit of the SHIFTER in the sequence of D31-P.

During receiving, the data D31-P moves in from the lowest bit of SHIFTER, and parity is performed when the data is copied to SHIFTER. Finally, copy the data to RX\_BUFF.

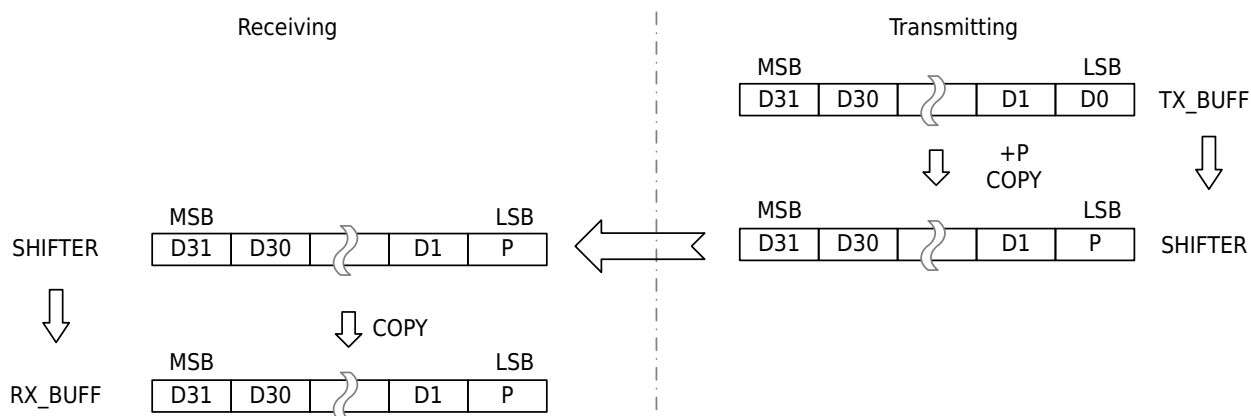


Figure 31-7 MSB Transmitted First, Parity Valid

### ● LSB Transmitted First, Parity Valid

During transmitting, the value of parity bit P is calculated according to the value of D30-D0, and then P is used instead of D31, which is copied from TX\_BUFF to SHIFTER in the sequence of D0-P and shifted out from the highest bit of the SHIFTER in the sequence of D0-P.

During receiving, the data D0-P moves in from the lowest bit of SHIFTER, and parity is performed when the data is copied to SHIFTER. The data D0-P is rearranged at the time of replication and copied to RX\_BUFF in the sequence of P-D0.

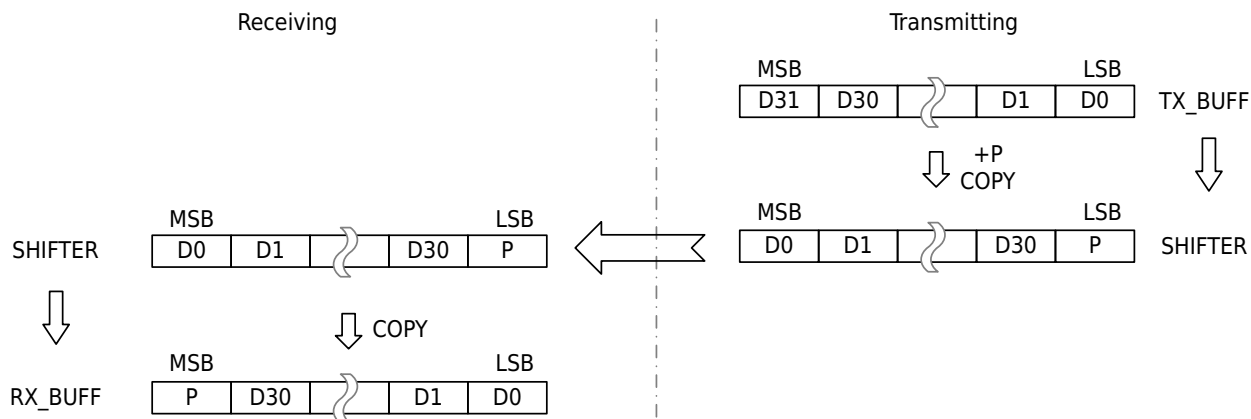


Figure 31-8 LSB Transmitted First, Parity Valid

### 31.3.3.3 Transfer Format

#### CPHA=0

When the SPI\_CFG2.CPHA bit is 0, SPI samples data on the odd edges of SCK and updates data on the even edges. The figure below is the SPI transmission timing diagram when CPHA=0. When the input level of the SSi signal becomes a active level, MOSI/MISO starts to update the transmission data. After the SSi signal becomes active for the first time on the edge of the first SCK signal, the data is sampled once per SCK period. MOSI/MISO signals are updated at 1/2 SCK periods after each sampling. The set value of CPOL bit does not affect the running time of SCK signal, but only affects the polarity of signal.

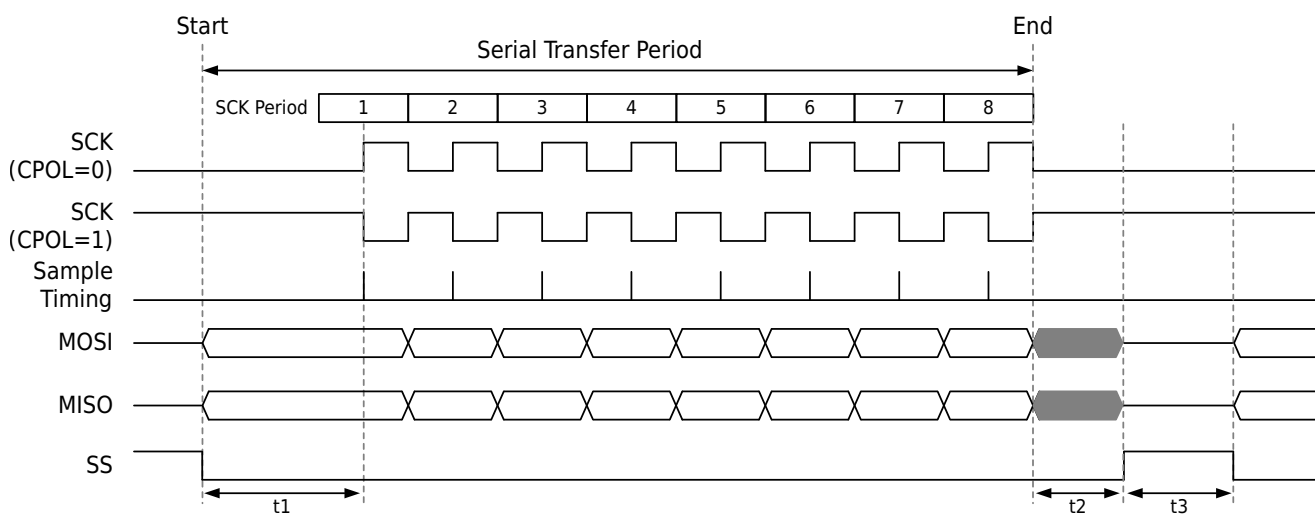


Figure 31-9 Data Transmission Format Diagram (CPHA=0)

In the above figure, t1 represents the interval from the SS signal being active to the SCK oscillation (SCK delay time, which can be controlled by setting SPI\_CFG1.MSSI and SPI\_CFG2.MSSIE), t2 represents the interval from the SCK oscillation stopping to the SS signal becoming inactive (SS inactive delay time, which can be controlled by setting SPI\_CFG1.MSSDL and SPI\_CFG2.MSSDLE), and t3 represents the minimum waiting time from the end of serial transmission to the start of the next transmission (the next access delay, which can be controlled by setting SPI\_CFG1.MIDI and SPI\_CFG2.MIDIE). t1, t2 and t3 are controlled by the master device on the SPI system. For details, see "[Master Action in SPI Operation Mode](#)".

#### CPHA=1

When SPI\_CFG2.CPHA bit is "1", SPI updates the data at the odd edge of SCK and samples the data at the even edge. The figure below is the SPI transmission timing diagram when CPHA=1. MOSI/MISO starts

to transmit data updates at the edge of the first SCK signal after the SSI signal becomes active. After this, data is updated every SCK period. Data is sampled at 1/2 SCK periods after each update. The set value of the SPI\_CFG2.CPOL bit does not affect the timing of the operation of the SCK signal but only affects the polarity of the signal.

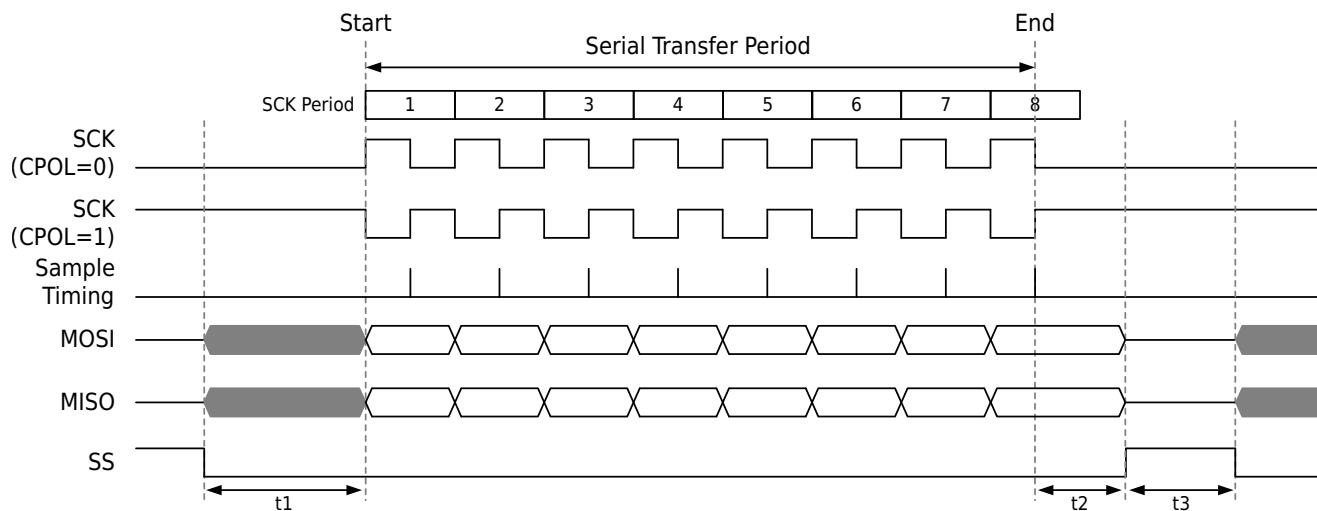


Figure 31-10 Data Transmission Format Diagram (CPHA=1)

t1, t2, t3 are the same as when the CPHA bit is 0.

#### 31.3.3.4 Communication Method

This SPI has two communication modes: Full-duplex synchronous serial communication and transmit-only communication, which can be selected through the SPI\_CR.TXMDS bit.

#### Full-duplex Synchronous serial Communication Mode

When the SPI\_CR.TXMDS bit is set to 0, the SPI operates in full-duplex synchronous serial communication mode. As shown in the figure below, the SPI\_CFG1.FTHLV bit is 0b00, the SPI\_CFG2.CPHA bit is 1 and the SPI\_CFG2.CPOL bit is 0, and the SPI performs 8-bit serial transmission.

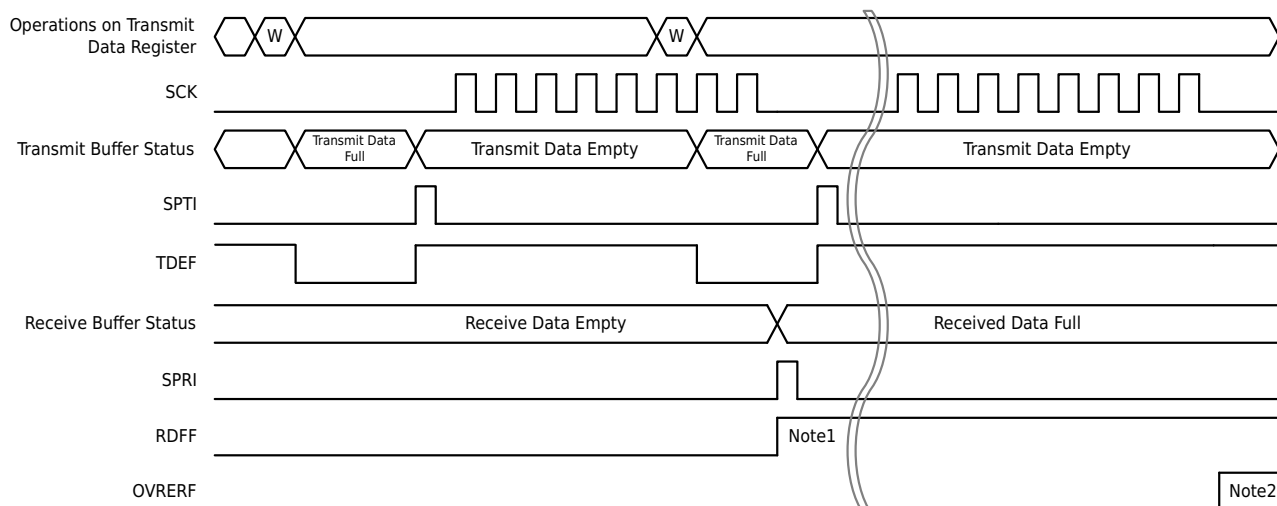


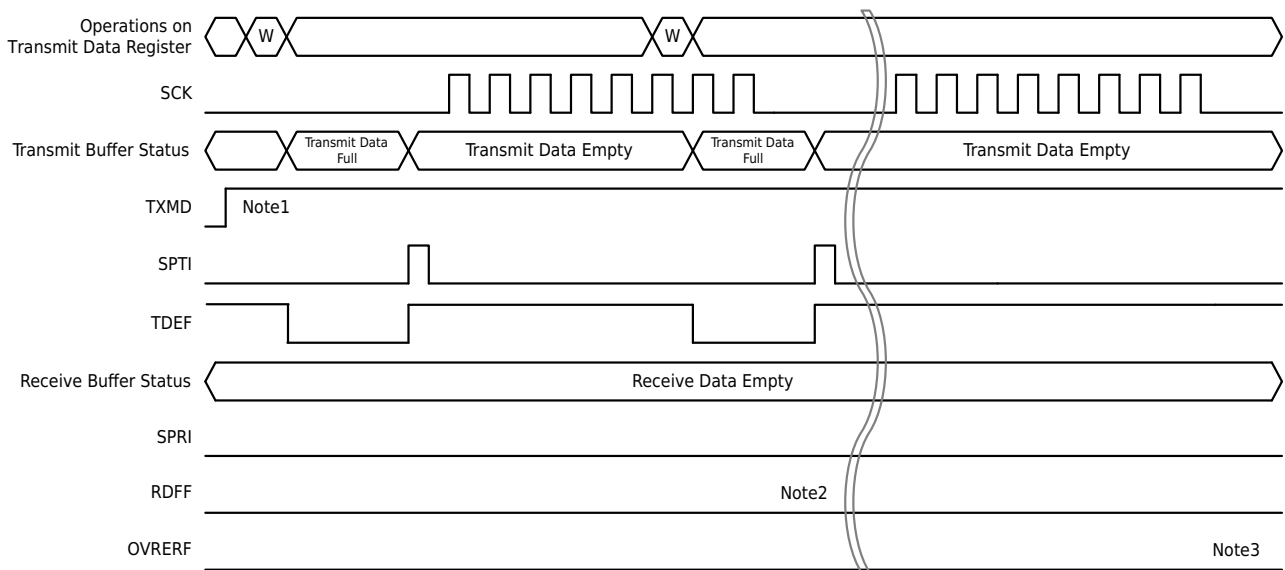
Figure 31-11 Full-duplex Synchronous Serial Communication

**Note**

1. When the serial transmission ends, if the receive buffer is empty, SPI will copy the received data from the SHIFTER register to the receive buffer, the receive buffer full flag RDFF is set to 1, and a receive data full interrupt request SPRI is generated.
2. When the serial transmission ends, if the receive buffer still holds the data received last time and has not been read by the system, the SPI will set the data overload flag OVRERF to 1, the data reception will be inactive, and the data in the receive SHIFTER register will be discarded.

**Transmit-Only Communication Mode**

When the SPI\_CR.TXMDS bit is set to 1, the SPI operates in transmit-only communication mode. As shown in the figure below, the SPI\_CFG1.FTHLV bit is 0b00, the SPI\_CFG2.CPHA bit is 1 and the SPI\_CFG2.CPOL bit is 0, and the SPI performs 8-bit serial transmission.

**Figure 31-12 Transmit-only Communication****Note**

1. Before setting up the transmit-only communication mode, make sure that there is no unread data in the receive buffer (RDFF=0) and no data overload error (OVRERF=0).
2. In the transmit-only communication mode, when the serial transmission ends, no data will be received even if the receive buffer is empty, and RDFF will always remain 0.
3. In the transmit-only communication mode, since the receive buffer is always empty, no data overload error will occur and the OVRERF flag will always remain 0.

**31.3.3.5 Communication Interval Time****Communication Interval Time Under Normal Communication Setting**

When the SPI\_CFG1.CTMDS bit is set to 0, the SPI uses normal communication setting. The interval time between two consecutive communications can be set by SPI\_CFG1.MIDI and SPI\_CFG2.MIDIE. The minimum interval time is  $1 \times \text{SCK} + 2 \times \text{PCLK1}$ . As shown in the figure below, the SPI\_CFG2.CPHA bit is 1 and the SPI\_CFG2.CPOL bit is 0, and the SPI performs serial transmission.

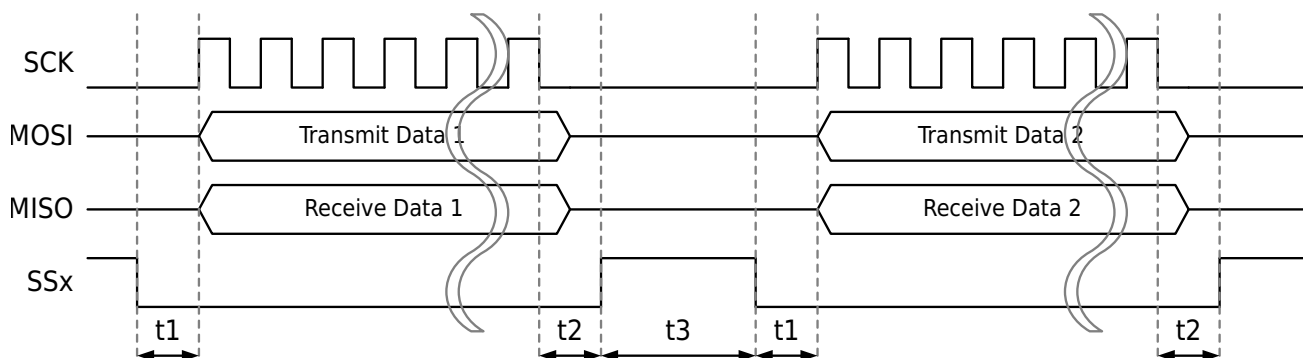


Figure 31-13 Normal Communication

In the above figure,  $t_1$  represents the time interval from the active SS signal to the first edge of SCK (SCK delay time, which can be controlled by setting SPI\_CFG1.MSSI and SPI\_CFG2.MSSIE),  $t_2$  represents the time interval from the last edge of SCK to the inactive SS signal (SS inactive delay time, which can be controlled by setting SPI\_CFG1.MSSDL and SPI\_CFG2.MSSDLE), and  $t_3$  represents the minimum waiting time from the end of this transmission to the beginning of the next transmission (the next access data delay, which can be controlled by setting SPI\_CFG1.MIDI and SPI\_CFG2.MIDIE).

**Note**

When the communication rate is 2 divided by PCLK1 (SPI\_CFG2.MBR=0b00 and SPI\_CFG2.CLKDIV=0b0000), please configure  $t_3$  to  $2 \times \text{SCK} + 2 \times \text{PCLK1}$  or above, that is, SPI\_CFG2.MIDIE=1 and SPI\_CFG1.MIDI=0x1-0x7.

### Communication Interval Time Under Continuous Communication Setting

When the SPI\_CFG1.CTMDS bit is set to 1, the SPI uses the continuous communication setting. Completely continuous and uninterrupted data communication can be achieved. As shown in the figure below, the SPI\_CFG2.CPHA bit is 1 and the SPI\_CFG2.CPOL bit is 0, and the SPI performs serial transmission.

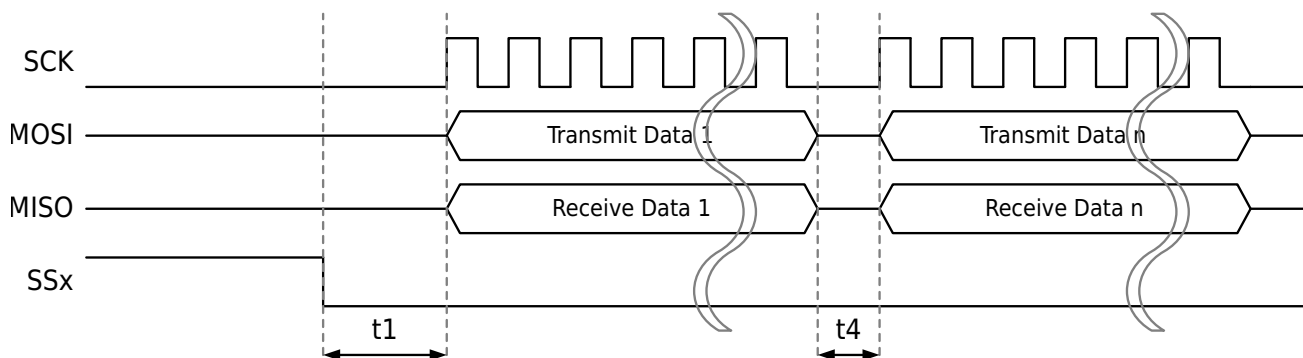


Figure 31-14 Continuous Communication

In the above figure,  $t_1$  represents the time interval from the activity of the SS signal to the first edge of SCK (SCK delay time, which can be controlled by setting SPI\_CFG1.MSSI and SPI\_CFG2.MSSIE), and  $t_4$  represents the time interval between each transmission, which can be controlled by setting SPI\_CFG1.MSSI and SPI\_CFG2.MSSIE. The relationship between  $t_1$  and  $t_4$  is defined as  $t_4 = t_1 - 1 \times \text{SCK}$ .

**Note**

- In continuous communication mode, if the master has no new data written into the transmit buffer, it will enter a pause state, SCK stops outputting, and the SS signal remains active, waiting for new transmit data to be written into the transmit buffer before continuing to transmit the next frame of data.
- In continuous communication mode, if the slave has no new data written into the transmit buffer, an underload error will occur and SPI\_SR.MODFERF and SPI\_SR.UDRERF will be set to 1. For detailed description, see "[Underload Error](#)".
- For continuous communication in clock synchronization mode, if the transmission format with CPHA=0 is used, ensure that the data is written before CPHA is set to 0.
- When using the transmission format of CPHA=0, the slave needs to write new data to the transmit buffer before the previous frame of data is sent. Otherwise, no matter whether the master starts the communication of the next frame of data, the slave will have an underload error.
- In continuous communication mode, the time interval between the last SCK of the previous frame of data and the first SCK of the next frame of data can also be adjusted through MSS1.

### 31.3.4 Operating Instructions

#### 31.3.4.1 Operational Mode Overview

This SPI supports 4-wire SPI mode and 3-wire clock synchronization mode. Serial communication can be performed as a master or slave in each operating mode. Set the SPI\_CR.MSTR and SPI\_CR.SPIMDS bits to set the SPI mode. The relationship between the SPI mode and the SPI\_CR register setting and the overview of each mode are shown in the following table.

Table 31-6 SPI Mode and Register Setting Relationship

| Mode                        | Master (SPI Operation)       | Slave (SPI Operation) | Master (Clock Synchronous Operation) | Slave (Clock Synchronous Operation) |
|-----------------------------|------------------------------|-----------------------|--------------------------------------|-------------------------------------|
| MSTR bit setting            | 1                            | 0                     | 1                                    | 0                                   |
| SPIMDS bit setting          | 0                            | 0                     | 1                                    | 1                                   |
| SCK signal                  | Output                       | Input                 | Output                               | Input                               |
| MOSI signal                 | Output                       | Input                 | Output                               | Input                               |
| MISO signal                 | Input                        | Output/Hi-Z           | Input                                | Output                              |
| SS0 signal                  | Output                       | Input                 | Hi-Z (not used)                      | Hi-Z (not used)                     |
| SS1-SS3 signal              | Output                       | Hi-Z                  | Hi-Z (not used)                      | Hi-Z (not used)                     |
| SS polarity change function | Yes                          | Yes                   | -                                    | -                                   |
| Maximum transmission rate   | PCLK1/2                      | PCLK1/6               | PCLK1/2                              | PCLK1/6                             |
| Clock source                | Internal baud rate generator | SCK input             | Internal baud rate generator         | SCK input                           |
| Clock polarity              | 2 species                    | 2 species             | 2 species                            | 2 species                           |
| Clock phase                 | 2 species                    | 2 species             | 2 species                            | 2 species <sup>*3</sup>             |
| Start transfer bit          | MSB/LSB                      | MSB/LSB               | MSB/LSB                              | MSB/LSB                             |



| Mode                            | Master (SPI Operation)                                           | Slave (SPI Operation)             | Master (Clock Synchronous Operation)                             | Slave (Clock Synchronous Operation) |
|---------------------------------|------------------------------------------------------------------|-----------------------------------|------------------------------------------------------------------|-------------------------------------|
| Transmission data length        | 4-32 bits                                                        | 4-32 bits                         | 4-32 bits                                                        | 4-32 bits                           |
| SCK delay control               | Yes                                                              | No                                | Yes                                                              | No                                  |
| SS inactive delay control       | Yes                                                              | No                                | Yes                                                              | No                                  |
| Next access delay control       | Yes                                                              | No                                | Yes                                                              | No                                  |
| Transport start method          | Write transmit buffer by transmit buffer empty interrupt request | SS input active or SCK clock edge | Write transmit buffer by transmit buffer empty interrupt request | SCK oscillation                     |
| Mode failure error              | No                                                               | Yes                               | No                                                               | Yes                                 |
| Transmit buffer empty detection | Yes                                                              | Yes                               | Yes                                                              | Yes                                 |
| Receive buffer full detection   | Yes <sup>*1</sup>                                                | Yes <sup>*1</sup>                 | Yes <sup>*1</sup>                                                | Yes <sup>*1</sup>                   |
| Overload error detection        | Yes <sup>*1</sup>                                                | Yes <sup>*1</sup>                 | Yes <sup>*1</sup>                                                | Yes <sup>*1</sup>                   |
| Parity error detection          | Yes <sup>*1 *2</sup>                                             | Yes <sup>*1 *2</sup>              | Yes <sup>*1 *2</sup>                                             | Yes <sup>*1 *2</sup>                |
| Underload error detection       | No                                                               | Yes                               | No                                                               | Yes                                 |

**Note**

1. When the SPI\_CR.TXMDS bit is 1, the receive buffer full detection, overload error detection, and parity error detection are not performed.
2. When the SPI\_CR.PAE bit is 0, parity error detection is not performed.
3. When the SPI\_CFG2.CPHA bit is 0, it is only active in continuous transmission mode. For specific setting steps, please refer to "[Slave Action in Clock Synchronous Operation Mode](#)".

### 31.3.4.2 Master Action in SPI Operation Mode

#### Action Description as Master

When the SPI data transmit buffer (TX\_BUFF) is empty (SPI\_SR.TDEF is 1), write the data of the frame length set by the SPI\_CFG1.FTHLV to the SPI\_DR register, SPI will update the SPI\_DR data to TX\_BUFF. If the SHIFTER register is empty, SPI copies the TX\_BUFF data to the SHIFTER register to start serial transmission.

When transmit data is copied to SHIFTER, SPI will change SHIFTER state to full state; When serial delivery is over, change to empty. The status of SHIFTER cannot be read.

After the SPI has generated the final SCK edge required for the last sampling timing, the current serial transmission concludes. The completion time is independent of the SPI\_CFG2.CPHA bit. When the

receive buffer (RX\_BUFF) is empty, SPI copies the data from SHIFTER to RX\_BUFF after serial delivery, which can be read by the SPI\_DR register.

The final sampling timing depends on the bit length of the transmit data, and the SPI data length of the master mode depends on SPI\_CFG2.DSIZE. The polarity of the SS output pin depends on the set value of SPI\_CFG1 register. For details about the SPI transfer format, see "[Transfer Format](#)".

### Initialization of SPI Master Mode

- Step1.** Set the SPI\_CFG1 register, including the setting of the baud rate, the setting of the number of frames used, the setting of various delay times, etc.;
- Step2.** Set the SPI\_CFG2 register, including SS level setting, data shift sequence setting, various delay allow bit setting, data format and clock polarity phase setting, etc.;
- Step3.** If you need to use interrupt, set the interrupt register of the system;
- Step4.** If you need to use DMA, set the relevant register for DMA;
- Step5.** Set the input/output pin;
- Step6.** Set the SPI\_CR register, including the mode and operation mode settings, self-diagnosis function settings, parity check settings, etc.;
- Step7.** Confirm the setting of SPI\_CR register;
- Step8.** Clear various flag bits;
- Step9.** Set the interrupt permission bit;
- Step10.** Set the SPI\_CR.SPE bit to 1 and the action starts.

### 31.3.4.3 Slave Action in SPI Operation Mode

#### Action Description as Slave

When the SPI\_CFG2.CPHA bit is 0, if SPI detects that the SS0 input signal becomes active, it needs to start to output the signal to MISO to drive the active data. Therefore, when the CPHA bit is 0, the SS0 input signal level is changed from inactive to active as a trigger signal that starts serial transmission.

When the CPHA bit is 1, if SPI detects the original SCK edge when the SS0 input signal is at an active level, it needs to start to drive active data to the MISO output signal. Therefore, when the CPHA bit is 1, the first SCK edge of the SS0 signal in the active state is considered as the trigger signal that starts serial transmission.

If SPI detects the start of serial delivery in the SHIFTER empty state, it changes SHIFTER to full state and cannot transfer data from TX\_BUFF to SHIFTER during serial delivery. If SHIFTER is full before starting serial transfer, SPI keeps SHIFTER full.

If SPI detects the SCK edge of the last sampling sequence, the current serial transmission concludes. The completion time is independent of the SPI\_CFG2.CPHA bit. When RX\_BUFF is empty, SPI copies the data from SHIFTER to RX\_BUFF after serial delivery. The data can be read by accessing SPI\_DR. SPI changes SHIFTER to an empty state after serial delivery, independent of the state of RX\_BUFF.

During serial delivery, a mode fault error occurs if SPI detects that the SS0 input signal is disabled.

The final sampling timing depends on the bit length of the transmit data, and the SPI data length depends on SPI\_CFG2.DSIZE. The polarity of SS0 input signal depends on SPI\_CFG1.SS0PV bit setting. For details about the SPI transfer format, see "[Transfer Format](#)".

**Note**

When the SPI\_CFG2.CPHA bit is 0, the SS0 input signal level is changed from inactive to active as a trigger signal to start serial transmission. Since the SS0 input signal is fixed in a slave mode, SPI will not start serial transmission normally. Therefore, in the structure where the SS0 input signal is fixed to the active state, the CPHA bit must be set to 1 in order to send and receive the SPI in the slave mode normally. If you need to set the CPHA bit to 0, you cannot fix the SS0 input signal.

**Initialization of SPI Slave Mode**

- Step1.** Set the SPI\_CFG1 register, mainly including the setting of the number of frames used;
- Step2.** Set the SPI\_CFG2 register, including transmission rate, data format and clock polarity phase setting;
- Step3.** If you need to use interrupt, set the interrupt register of the system;
- Step4.** If you need to use DMA, set the relevant register for DMA;
- Step5.** Set the input/output pin;
- Step6.** Set the SPI\_CR register, including the mode and operation mode settings, self-diagnosis function settings, parity check settings, etc.;
- Step7.** Confirm the setting of SPI\_CR register;
- Step8.** Clear various flag bits;
- Step9.** Set the interrupt permission bit;
- Step10.** Set the SPI\_CR.SPE bit to 1 and the action starts.

**31.3.4.4 Master Action in Clock Synchronous Operation Mode**

When the SPI\_CR.SPIMDS bit is 1, the SPI is in clock synchronous operation mode. When operating in this mode, SPI only uses the 3 pins SCK, MOSI and MISO for communication, and the SSi pin is released for normal I/O functions.

Although the SSi pins are not used in the clock synchronous operation mode, the internal operation of the module is the same as the SPI mode of operation.

**Action Description as Master**

When the SPI data transmit buffer (TX\_BUFF) is empty (SPI\_SR.TDEF is 1), write the data with the frame length set by the SPI\_CFG1.FTHLV to the SPI\_DR register, SPI will update the SPI\_DR data to TX\_BUFF. If the SHIFTER register is empty, SPI copies the TX\_BUFF data to the SHIFTER register to start serial transmission.

When transmit data is copied to SHIFTER, SPI will change SHIFTER state to full state; When serial delivery is over, change to empty. The status of SHIFTER cannot be read.

After the SPI has generated the final SCK edge required for the last sampling timing, the current serial transmission concludes. The completion time is independent of the SPI\_CFG2.CPHA bit. When the receive buffer (RX\_BUFF) is empty, SPI copies the data from SHIFTER to RX\_BUFF after serial delivery, which can be read by the SPI\_DR register.

The final sampling timing depends on the bit length of the transmit data, and the SPI data length of the master mode depends on SPI\_CFG2.DSIZE. For details about the SPI transfer format, see "[Transfer Format](#)".

**Master Initialization in Clock Synchronous Operation Mode**

- Step1.** Set the SPI\_CFG1 register, including the setting of the baud rate, the setting of the number of frames used, the setting of various delay times, etc.;

- Step2.** Set the SPI\_CFG2 register, including the data shift sequence setting, the setting of the allowable bit of various delays, the setting of the data format and the clock polarity phase, etc.;
- Step3.** If you need to use interrupt, set the interrupt register of the system;
- Step4.** If you need to use DMA, set the relevant register for DMA;
- Step5.** Set the input/output pin;
- Step6.** Set the SPI\_CR register, including the mode and operation mode settings, self-diagnosis function settings, parity check settings, etc.;
- Step7.** Confirm the setting of SPI\_CR register;
- Step8.** Clear various flag bits;
- Step9.** Set the interrupt permission bit;
- Step10.** Set the SPI\_CR.SPE bit to 1 and the action starts.

### 31.3.4.5 Slave Action in Clock Synchronous Operation Mode

#### Action Description as Slave

When the SPI\_CFG2.CPHA bit is 0, the SPI needs to detect that the SS0 input signal becomes a active level as a trigger signal to start serial communication. Since the SS0 pin is not used in the clock synchronous operation mode, the following sequence must be followed for normal communication:

- Step1.** Set up communication and set the SPI\_CR.SPE to 1;
- Step2.** Set SPI\_CFG2.CPHA to 1;
- Step3.** Set SPI\_CFG1.CTMDS to 1;
- Step4.** Write transmit data to SPI\_DR;
- Step5.** Set SPI\_CFG2.CPHA to 0;
- Step6.** Waiting for the master clock to start communicating.

When the CPHA bit is 1, if SPI detects the original SCK edge when the SS0 input signal is at an active level, it needs to start to drive active data to the MISO output signal. Since the SS0 pin is not used in the clock synchronous mode of operation, when the CPHA bit is 1, the first SCK edge is considered as the trigger signal to start the serial transmission.

If SPI detects the start of serial delivery in the SHIFTER empty state, it changes SHIFTER to full state and cannot transfer data from TX\_BUFF to SHIFTER during serial delivery. If SHIFTER is full before starting serial transfer, SPI keeps SHIFTER full.

If SPI detects the SCK edge of the last sampling sequence, the current serial transmission concludes. When RX\_BUFF is empty, SPI copies the data from SHIFTER to RX\_BUFF after serial delivery. The data can be read by accessing SPI\_DR. SPI changes SHIFTER to an empty state after serial delivery, independent of the state of RX\_BUFF.

The final sampling timing depends on the bit length of the transmitted data, and the data length of the SPI in slave mode depends on SPI\_CFG2.DSIZE.

#### Slave Initialization in Clock Synchronous Operation Mode

- Step1.** Set the SPI\_CFG1 register, mainly including the setting of the number of frames used;
- Step2.** Set the SPI\_CFG2 register, including transmission rate, data format and clock polarity phase setting, etc.;
- Step3.** If you need to use interrupt, set the interrupt register of the system;
- Step4.** If you need to use DMA, set the relevant register for DMA;
- Step5.** Set the input/output pin;

- Step6.** Set the SPI\_CR register, including the mode and operation mode settings, self-diagnosis function settings, parity check settings, etc.;
- Step7.** Confirm the setting of SPI\_CR register;
- Step8.** Clear various flag bits;
- Step9.** Set the interrupt permission bit;
- Step10.** Set the SPI\_CR.SPE bit to 1 and the action starts.

#### 31.3.4.6 Processing Flow of Several SPI Actions

##### SPI Data Transmission Process as Master

- Step1.** Wait for the interrupt of the data transmit buffer being empty or confirm that the data transmit buffer is in an empty state by polling;
- Step2.** Writes the transmit data to SPI\_DR register;
- Step3.** Repeat steps 1 and 2 until the final data is transmit;
- Step4.** Reset the allowable bit TXIE of transmit data register empty interrupt, and set the allowable bit IDIE of SPI idle state interrupt to 1;
- Step5.** The hardware generates an SPI idle state interrupt;
- Step6.** Set SPE to 0, stop SPI, and reset IDIE.

##### Data Reception Process

- Step1.** Wait for the full interrupt of the data receive buffer register or confirm by polling that the data receive buffer register is full;
- Step2.** Read data from the receive buffer register by accessing SPI\_DR;
- Step3.** Repeat steps 1 and 2 until the last received data is read;
- Step4.** Clears the RXIE bit for the full interrupt of the data receive buffer register.

##### Communication Error Process

- Step1.** Wait for the communication error interrupt or confirm that the communication error flag bit (MODFERF/OVERRF/UDRERF/PERF) is set to 1 by polling;
- Step2.** Verify SS0 status and troubleshoot mode errors;
- Step3.** Clear SPE and stop SPI;
- Step4.** Identify the types of communication errors by error flag bits and handle them;
- Step5.** Clear the error flag bit;
- Step6.** Start SPI and restart the communication.

#### 31.3.5 Self-diagnosis of Parity

Parity check circuit is composed of parity check bits for transmitting data and error detection part for receiving data. You can use the self-diagnosis function to diagnose the fault of the parity circuit as shown in the following figure.

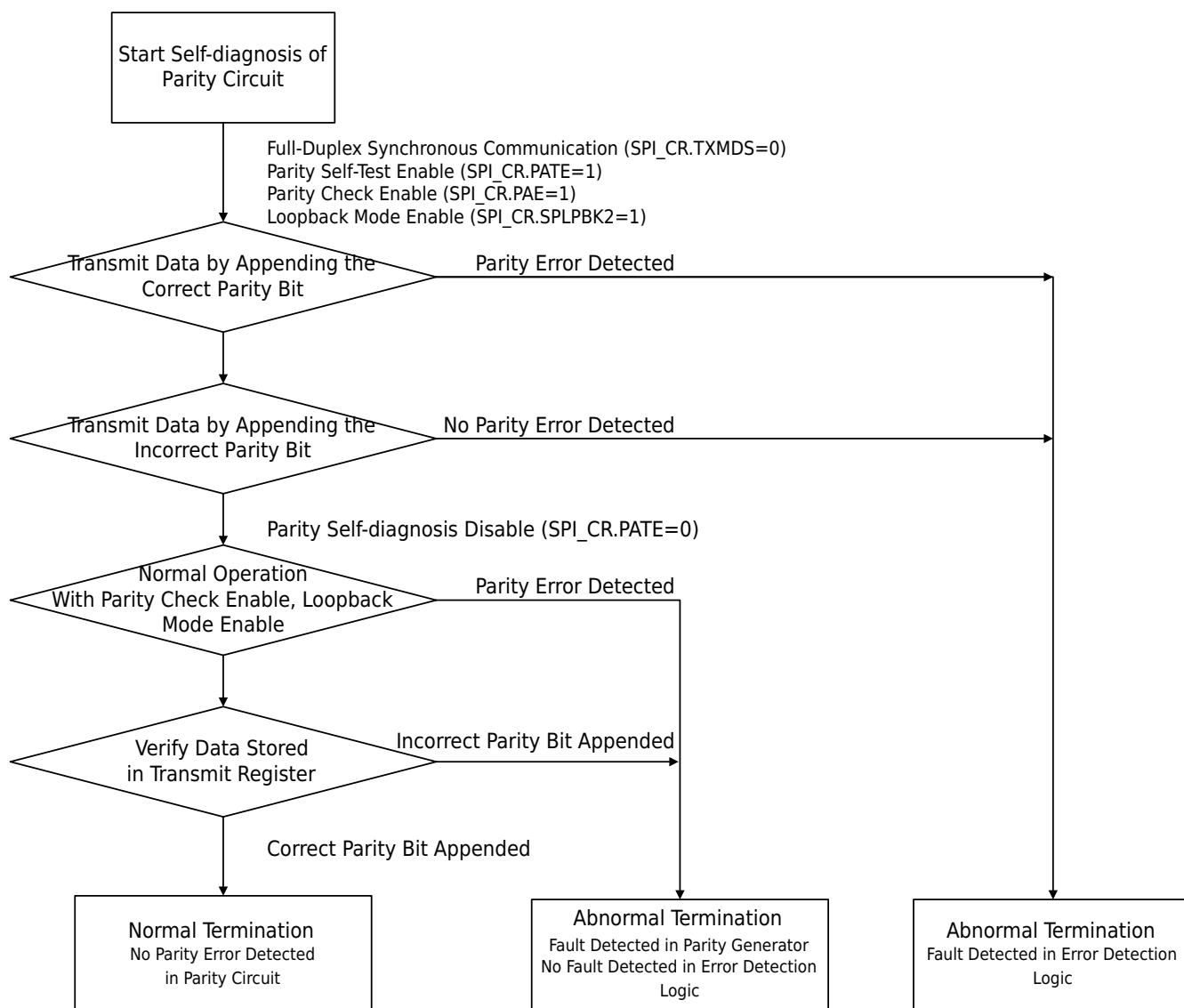


Figure 31-15 Parity Check Process

### 31.3.6 Error Detection

#### 31.3.6.1 Error Detection Overview

In the normal SPI serial transmission, the system writes the data serially to SPI\_DR register and obtains the data serially received by reading SPI\_DR register. However, due to the status of the transmit/receive buffer and the SPI at the start/end of serial delivery, abnormal delivery may occur in some cases. When an abnormal transfer occurs, SPI detects this transfer as an underload error, an overload error, a parity error, or a mode fault error. The correspondence between abnormal transmission and SPI error detection is shown in the following table.

Table 31-7 Error Detection Correspondence Table

| SN | Occurrence Condition                              | SPI Operation                    | Detect Errors |
|----|---------------------------------------------------|----------------------------------|---------------|
| ①  | Write SPI_DR register in full transmit buffer     | Hold transmit buffer content     | No            |
|    |                                                   | Write data loss                  |               |
| ②  | Read SPI_DR register when receive buffer is empty | Output last serial received data | No            |

| SN | Occurrence Condition                                                                                             | SPI Operation                      | Detect Errors   |
|----|------------------------------------------------------------------------------------------------------------------|------------------------------------|-----------------|
| ③  | In slave mode, serial delivery begins without data being transferred to the SHIFTER register                     | Aborting serial transmission       | Underload Error |
|    |                                                                                                                  | Loss of sending and receiving data |                 |
|    |                                                                                                                  | Stop driving MISO output signal    |                 |
|    |                                                                                                                  | Stop SPI function                  |                 |
| ④  | Slave mode: The active level width of the SS0 pin does not reach the time required for data transmission         | Abort transmission                 | Mode Error      |
|    |                                                                                                                  | Loss of sending and receiving data |                 |
|    |                                                                                                                  | Stop SPI function                  |                 |
| ⑤  | End serial transfer in full receive buffer                                                                       | Hold receive buffer content        | Overload Error  |
|    |                                                                                                                  | Loss of received data              |                 |
| ⑥  | Error parity bits are received in full-duplex synchronous serial communication and the parity function is active | Parity error flag is valid         | Parity Error    |

In case ① described, SPI does not detect errors. To prevent data omissions during data write to SPI\_DR register, data must be written to SPI\_DR register by an empty interrupt of the transmit buffer. Similarly in case ②, SPI does not detect errors. To prevent unrelated data from being read, SPI\_DR data read must be done by a full interrupt request of receive register.

### 31.3.6.2 Underload Error

When the MSTR bit is 0, the SPI operates in the slave state. If the SPE is set to 1, the transmission data is not ready before the SS0 pin receives a active level, and the SPI has an underload error, SPI\_SR.MODFERF and SPI\_SR.UDRERF will be set to 1.

When an underload error is detected, SPI will stop driving signal output, and SPI\_CR.SPE is set to 0.

Monitoring of underload errors can be done by accessing SPI\_SR register directly, or by using SPI error interrupt to read SPI\_SR. If you do not use the error interrupt, use the polling method to monitor for unloaded errors.

When SPI\_SR.MODFERF is 1, the system disables writing 1 to SPE bits. To set SPI\_CR.SPE to 1 to enable the SPI function, the MODFERF flag must be cleared to 0 first.

### 31.3.6.3 Mode Error

When SPI is in master mode, please do not set SPI\_CR.MODFE to 1. In slave mode, when the SSi active level width does not reach the time required for data transmission, a mode fault occurs, SPI\_SR.MODFERF is set to 1, and SPI\_CR.SPE is set to 0. When transmission is required, set SPI\_CR.SPE to 1 after clearing SPI\_SR.MODFERF.

### 31.3.6.4 Overload Error

SPI\_SR.OVRERF flag is set to 1 if serial delivery is ended in full receive buffer. Since SPI does not copy the data of the SHIFTER register to the receive buffer in the case of the OVRERF flag is 1, the receive buffer stores the received data before the error occurs. The OVRERF flag only can be written to 0 after the SPI\_SR register is read when the OVRERF flag is 1.



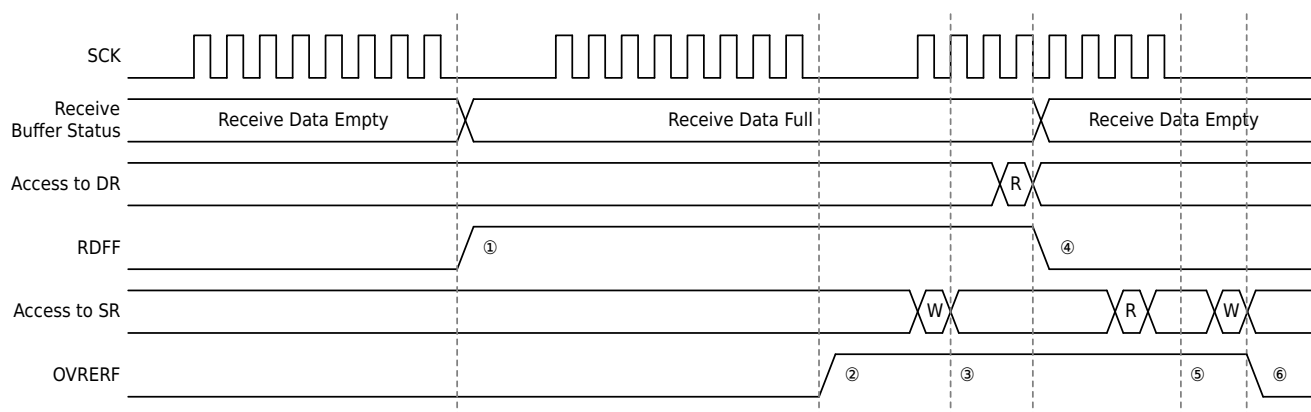


Figure 31-16 Overload Error Processing

The following describes the operation content of the flags in the sequence shown in ①-⑥ in the figure:

① When the receive buffer is empty, the serial transmission ends, the SPI operates normally, the data of the SHIFTER register is copied to the receive buffer, and the RDFF flag is set to 1.

② At the end of the serial transfer with the receive buffer full, the SPI detects an overload error and sets the OVRERF flag to 1. SPI will not copy the data of the SHIFTER register to the receive buffer. Even when the PAE bit is 1, a parity error is not detected.

③ Clear the OVRERF flag.

④ Read access to SPI\_DR register allows the SPI to read the data in the receive buffer. The RDFF flag becomes 0. Even if the receive buffer status is empty at this time, the OVRERF flag will not become 0.

⑤ When the serial transmission ends when the OVRERF flag is 1 (overload error), the SPI will not copy the data of the SHIFTER register to the receive buffer, nor will it generate a receive buffer full interrupt, and the RDFF flag will remain at 0. Even if the PAE bit is 1, no parity error is detected. In the state where an overload error occurs, if the serial transmission ends without copying the received data from the SHIFTER register to the receive buffer, the SPI judges that the SHIFTER register is in an empty state and allows data to be transferred from the transmit buffer to the SHIFTER register.

⑥ Normal reception operation can only be performed after the OVRERF flag changes to 0.

Monitoring overload errors can be done by accessing the SPI\_SR register directly or using the SPI error interrupt to access the SPI\_SR register. When performing serial transmission, the occurrence of overload errors must be detected as early as possible by methods such as reading the SPI\_SR register immediately after reading the SPI\_DR register.

In master mode, if the communication automatic suspend function is enabled (set the SPI\_CR.CSUSPE bit to 1), the SPI will suspend the communication clock in the last sampling period before the overload error occurs. At this time, since the SHIFTER register has not completed the reception of the last bit, the SPI remains in the normal communication state and the overload error will not occur. The receive buffer can be read during the communication clock pause. After reading, the receive buffer status becomes empty, and the SPI restarts the communication clock to complete the reception of the last bit of data.



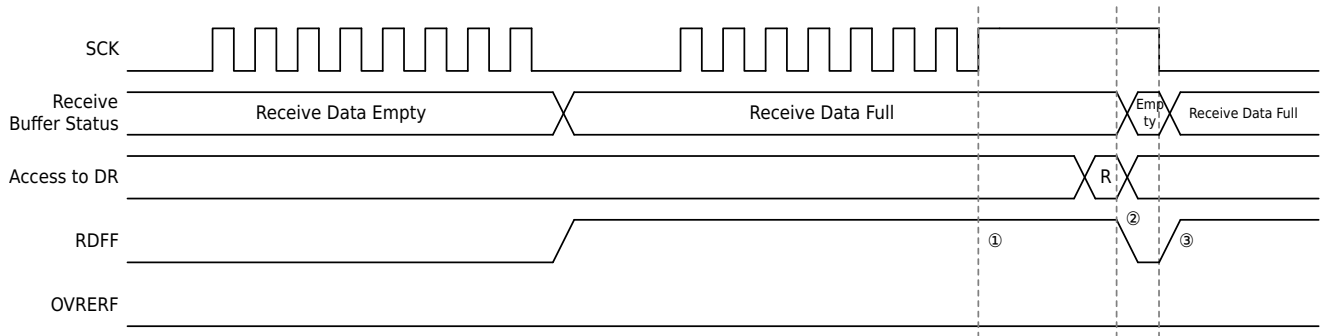


Figure 31-17 Schematic Diagram of Action when Clock Automatic Stop Function Enabled (CPHA=1)

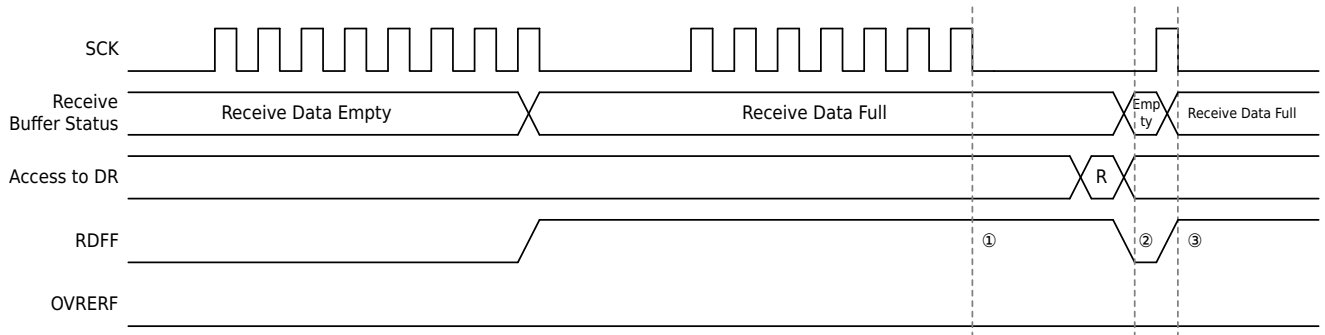


Figure 31-18 Schematic Diagram of Action when Clock Automatic Stop Function Enabled (CPHA=0)

The following describes the running contents of the flags in the time product family shown in figures ①-③:

- ① When the receive buffer is full, the SPI pauses the communication clock before the last bit of data is received. No overload error will occur at this time.
- ② After reading the data in the receive buffer by accessing SPI\_DR, the receive buffer becomes empty, the RDFF flag is cleared, and the SPI restarts the communication clock to complete the last bit of data communication.
- ③ The last bit of data communication is completed, the receive buffer register becomes full again, the RDFF flag is set to 1, and the received data can be read by accessing SPI\_DR.

### 31.3.6.5 Parity Error

When the SPI\_CR.TXMDS bit is 0 and the SPI\_CR.PAE bit is 1, the SPI will perform a parity check at the end of full-duplex synchronous serial communication. When SPI detects a parity error in the received data, set the SPI\_SR.PERF flag to 1. In the SPI\_SR.OVRERF bit 1 state, the received data is not checked for parity errors because SPI does not copy the data of the SHIFTER register to the receive buffer. The PERF flag only can be written to 0 after the SPI\_SR register is read when the PERF flag is 1.

The following figure shows an example of the OVRERF flag and the PERF flag in action. In the example in the figure, the SPI performs 8-bit serial transmission of full-duplex synchronous serial communication when the SPI\_CR.TXMDS bit is 0 and the SPI\_CR.PAE bit is 1.

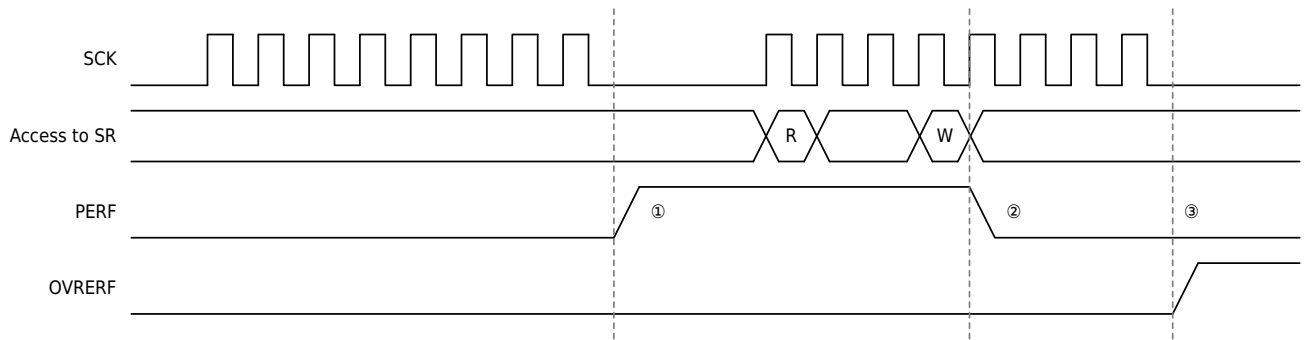


Figure 31-19 Parity Error

The following describes the running contents of the flags in the time product family shown in figures ① to ③:

① SPI did not detect an overload error and serial delivery ended normally. SPI copies the data of the SHIFTER register to the receive buffer. At this time, SPI performs parity check on the received data. If a parity error is detected, the PERF flag is set to 1.

② Clear the RERF flag.

③ SPI detects an overload error. At this time, SPI does not copy the data of the SHIFTER register to the receiving buffer, SPI does not perform parity check on the data, and no parity check error occurs.

Monitoring parity errors can be done by accessing the SPI\_SR register directly or using the SPI error interrupt to access the SPI\_SR register. When performing serial transmission, the occurrence of parity errors must be detected as early as possible by methods such as reading the SPI\_SR register.

### 31.3.7 SPI Initialization

Write operation or mode fault error detection is used to clear SPE bit, which can inactiveate SPI functions and initialize some SPI functions. If a system reset occurs or the function reset register RMU\_FRST1 is used, all SPI functions are initialized.

#### Initialization by Clear SPE Bit

When the SPI\_CR.SPE bit is set to 0, the SPI performs the following initialization operations:

- Abort in-progress serial delivery.
- If in slave state, stop driving the output signal (the state changes to Hi-Z).
- SPI internal state is initialized.
- Clear transmit buffer, set SPI\_SR.TDEF to 1.

SPI control bits are not initialized by clear SPE bit. As a result, SPI can be started in the same transmission mode as before initialization, as long as SPE is relocated to 1.

Clearing SPE bit does not initialize error flag bits and sequence states. Thus, even after the SPE is cleared, it is possible to confirm the occurrence of errors during SPI transfer by reading the data of the receive buffer.

Clearing SPE bit clears the transmit buffer and sets the SPI\_SR.TDEF to 1. Therefore, if the SPI\_CR.TXIE bit is set to 1 after initialization, an SPI transmit buffer empty interrupt will be generated. To avoid this interrupt, you must clear the SPE bit to 0 while setting the TXIE bit to 0.

## Initialization by Reset

Initialization by system reset or using the function reset register RMU\_FRST1 will initialize all SPI control bits, status bits and data registers. For the contents of the system reset and function reset registers, please refer to the relevant description in the "Reset Module Unit (RMU)" chapter.

### 31.3.8 Interrupt and Events

The SPI receive buffer full, transmit buffer empty, mode fault (overload, underload, parity error) and SPI idle can be used as interrupt sources or internal trigger sources. Transmission completion can only be used as an internal trigger source.

The interrupt of overload, underload and parity errors is integrated into the SPI error interrupt SPEI, so the actual interrupt source needs to be judged by the flag. The specific description of SPI interrupt/internal trigger source is shown in the following table. Once the condition is met, the corresponding interrupt/internal trigger request is generated. For receive buffer full and transmit buffer empty interrupt sources, data transmission changes buffer status to clear.

**Table 31-8 SPI Interrupt Source and Trigger Source Description**

| Interrupt/Internal Trigger Source               | Abbreviation | Interrupt Source | Internal Trigger Source | Condition                                                                                                                                                                                                                                                                |
|-------------------------------------------------|--------------|------------------|-------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Receive buffer full                             | SPI_SPRI     | √                | √                       | When the receive buffer becomes full while the SPI_CR.RXIE bit is 1                                                                                                                                                                                                      |
| Transmit buffer empty                           | SPI_SPTI     | √                | √                       | When the transmit buffer becomes empty while the SPI_CR.TXIE bit is 1                                                                                                                                                                                                    |
| SPI errors (overload, underload, parity errors) | SPI_SPEI     | √                | √                       | When the SPI_CR.EIE bit is 1, the SPI_SR.OVRERF, SPI_SR.PERF or SPI_SR.MODFERF and SPI_SR.UDRERF flags become 1.                                                                                                                                                         |
| SPI idle                                        | SPI_SPII     | √                | √                       | When the SPI_CR.IDIE bit is 1:<br><ul style="list-style-type: none"> <li>● Master mode: Data transmissions is completed or SPI_CR.SPE is written from 1 to 0.</li> <li>● Slave mode: SPI_CR.SPE is written from 1 to 0.</li> </ul>                                       |
| Data transmission completed                     | SPI_SPEND    | -                | √                       | <ul style="list-style-type: none"> <li>● Master mode: Data transmission completed.</li> <li>● Slave mode: Four-wire, when the SSL pin is at an inactive level; three-wire, when the SCK pin clock reaches the number of clocks required for data transmission</li> </ul> |

The user can write the vectors that can be used as trigger sources in the above table into different trigger object registers to implement various event trigger functions. For the vectors corresponding to the above event trigger sources, please refer to the "Interrupt Event List" in the "Interrupt Controller (INTC) " chapter.

## 31.4 Register Description

### 31.4.1 Register List

- SPI1 base address: 0x4001C000
- SPI2 base address: 0x4001C400
- SPI3 base address: 0x4001C800
- SPI4 base address: 0x40020000
- SPI5 base address: 0x40020400
- SPI6 base address: 0x40020800

Table 31-9 Serial Peripheral Interface (SPI) Register Offset

| Offset | Register                 | Description                                | Width |
|--------|--------------------------|--------------------------------------------|-------|
| 0x00   | <a href="#">SPI_DR</a>   | SPI Data Register                          | 32    |
| 0x04   | <a href="#">SPI_CR</a>   | SPI Control Register                       | 32    |
| 0x0C   | <a href="#">SPI_CFG1</a> | SPI Communication Configuration Register 1 | 32    |
| 0x14   | <a href="#">SPI_SR</a>   | SPI Status Register                        | 32    |
| 0x18   | <a href="#">SPI_CFG2</a> | SPI Communication Configuration Register 2 | 32    |

31.4.2 SPI Data Register (SPI\_DR)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0x00   | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | SPD          |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access | RW           |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit  | Name | Function    | Description      | Access |
|------|------|-------------|------------------|--------|
| 31:0 | SPD  | Serial data | SPI data storage | RW     |

## 31.4.3 SPI Control Register (SPI\_CR)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |     |      |      |       |      |      |      |     |        |     |         |        |      |          |       |        |    |    |  |    |    |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-----|------|------|-------|------|------|------|-----|--------|-----|---------|--------|------|----------|-------|--------|----|----|--|----|----|
| 0x04   | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15  | 14   | 13   | 12    | 11   | 10   | 9    | 8   | 7      | 6   | 5       | 4      | 3    | 2        | 1     | 0      |    |    |  |    |    |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |     |      |      |       |      |      |      |     |        |     |         |        |      |          |       |        |    |    |  |    |    |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | PAE | PAOE | PATE | MODFE | IDIE | RXIE | TXIE | EIE | CSUSPE | SPE | SPLPBK2 | SPLPBK | MSTR | Reserved | TXMDS | SPIMDS |    |    |  |    |    |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RW  | RW   | RW   | RW    | RW   | RW   | RW   | RW  | RW     | RW  | RW      | RW     | RW   | RW       | RW    | RW     | RW | RW |  | RW | RW |
|        |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |     |      |      |       |      |      |      |     |        |     |         |        |      |          |       |        |    |    |  |    |    |
|        |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |     |      |      |       |      |      |      |     |        |     |         |        |      |          |       |        |    |    |  |    |    |

| Bit | Name            | Function                                    | Description                                                                                                                                                      | Access |
|-----|-----------------|---------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 7   | CSUSPE          | Communication auto suspend function enabled | <ul style="list-style-type: none"> <li>● 0b0: Communication auto suspend function inactive</li> <li>● 0b1: Communication auto suspend function active</li> </ul> | RW     |
| 6   | SPE             | SPI function enabled                        | <ul style="list-style-type: none"> <li>● 0b0: SPI function disabled</li> <li>● 0b1: SPI function enabled</li> </ul>                                              | RW     |
| 5   | SPLPBK2         | SPI Loop 2-bit                              | <ul style="list-style-type: none"> <li>● 0b0: Normal mode</li> <li>● 0b1: Loopback mode (transmit data=receive data)</li> </ul>                                  | RW     |
| 4   | SPLPBK          | SPI loopback position                       | <ul style="list-style-type: none"> <li>● 0b0: Normal mode</li> <li>● 0b1: Loopback mode (inverted phase of transmit data=received data)</li> </ul>               | RW     |
| 3   | MSTR            | SPI Master-slave mode selection             | <ul style="list-style-type: none"> <li>● 0b0: Slave mode</li> <li>● 0b1: Master mode</li> </ul>                                                                  | RW     |
| 2   | <i>Reserved</i> | <i>Reserved</i>                             | -                                                                                                                                                                | -      |
| 1   | TXMDS           | Communication mode selection                | <ul style="list-style-type: none"> <li>● 0b0: Full-duplex synchronous serial communication</li> <li>● 0b1: Send serial communication only</li> </ul>             | RW     |
| 0   | SPIMDS          | SPI mode selection                          | <ul style="list-style-type: none"> <li>● 0b0: SPI operation (4-line)</li> <li>● 0b1: Clock synchronous operation (3-wire)</li> </ul>                             | RW     |

### 31.4.4 SPI Communication Configuration Register 1 (SPI\_CFG1)

| Offset | Bit Position |      |    |    |          |       |    |    |          |      |    |    |          |    |    |        |    |    |       |    |    |       |       |       |          |        |   |    |          |    |       |       |       |
|--------|--------------|------|----|----|----------|-------|----|----|----------|------|----|----|----------|----|----|--------|----|----|-------|----|----|-------|-------|-------|----------|--------|---|----|----------|----|-------|-------|-------|
| 0x0C   | 31           | 30   | 29 | 28 | 27       | 26    | 25 | 24 | 23       | 22   | 21 | 20 | 19       | 18 | 17 | 16     | 15 | 14 | 13    | 12 | 11 | 10    | 9     | 8     | 7        | 6      | 5 | 4  | 3        | 2  | 1     | 0     |       |
| Reset  | 0x00000010   |      |    |    |          |       |    |    |          |      |    |    |          |    |    |        |    |    |       |    |    |       |       |       |          |        |   |    |          |    |       |       |       |
| Name   | Reserved     | MIDI |    |    | Reserved | MSSDL |    |    | Reserved | MSSI |    |    | Reserved |    |    | CLKDIV |    |    | SS3PV |    |    | SS2PV | SS1PV | SS0PV | Reserved | SPRDTD |   |    | Reserved |    |       | CTMDS | FTHLV |
| Access |              | RW   |    |    |          | RW    |    |    |          | RW   |    |    |          |    |    | RW     |    |    | RW    | RW | RW | RW    |       | RW    |          |        |   | RW | CTMDS    | RW | FTHLV |       |       |

| Bit   | Name     | Function                                               | Description                                                                                                                                                                                                                                                                                                                             | Access |
|-------|----------|--------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31    | Reserved | Reserved                                               | -                                                                                                                                                                                                                                                                                                                                       | -      |
| 30:28 | MIDI     | Master next access data interval idle time setting bit | <ul style="list-style-type: none"> <li>● 0b000: 1 SCK + 2 PCLK1s</li> <li>● 0b001: 2 SCKs + 2 PCLK1s</li> <li>● 0b010: 3 SCKs + 2 PCLK1s</li> <li>● 0b011: 4 SCKs + 2 PCLK1s</li> <li>● 0b100: 5 SCKs + 2 PCLK1s</li> <li>● 0b101: 6 SCKs + 2 PCLK1s</li> <li>● 0b110: 7 SCKs + 2 PCLK1s</li> <li>● 0b111: 8 SCKs + 2 PCLK1s</li> </ul> | RW     |
| 27    | Reserved | Reserved                                               | -                                                                                                                                                                                                                                                                                                                                       | -      |
| 26:24 | MSSDL    | Master SS inactive delay setting bit                   | <ul style="list-style-type: none"> <li>● 0b000: 1 SCKs</li> <li>● 0b001: 2 SCKs</li> <li>● .....</li> <li>● 0b110: 7 SCKs</li> <li>● 0b111: 8 SCKs</li> </ul>                                                                                                                                                                           | RW     |
| 23    | Reserved | Reserved                                               | -                                                                                                                                                                                                                                                                                                                                       | -      |
| 22:20 | MSSI     | Master SS idle time setting bit                        | <ul style="list-style-type: none"> <li>● 0b000: 1 SCKs</li> <li>● 0b001: 2 SCKs</li> <li>● .....</li> <li>● 0b110: 7 SCKs</li> <li>● 0b111: 8 SCKs</li> </ul>                                                                                                                                                                           | RW     |
| 19:16 | Reserved | Reserved                                               | -                                                                                                                                                                                                                                                                                                                                       | -      |
| 15:12 | CLKDIV   | Communication reference clock selection bit            | <ul style="list-style-type: none"> <li>● 0000: PCLK1/2</li> <li>● 0001: PCLK1/4</li> <li>● 0010: PCLK1/6</li> <li>● 0011: PCLK1/8</li> <li>● 0100: PCLK1/10</li> <li>● 0101: PCLK1/12</li> <li>● 0110: PCLK1/14</li> <li>● 0111: PCLK1/16</li> <li>● 1000: PCLK1/18</li> <li>● 1001: PCLK1/20</li> <li>● 1010: PCLK1/22</li> </ul>      | RW     |



| Bit   | Name            | Function                                    | Description                                                                                                                                                                | Access |
|-------|-----------------|---------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 15:12 | CLKDIV          | Communication reference clock selection bit | <ul style="list-style-type: none"> <li>● 1011: PCLK1/24</li> <li>● 1100: PCLK1/26</li> <li>● 1101: PCLK1/28</li> <li>● 1110: PCLK1/30</li> <li>● 1111: PCLK1/32</li> </ul> | RW     |
| 11    | SS3PV           | Polarity setting of SS3 signal              | <ul style="list-style-type: none"> <li>● 0b0: Low level active</li> <li>● 0b1: High level active</li> </ul>                                                                | RW     |
| 10    | SS2PV           | Polarity setting of SS2 signal              | <ul style="list-style-type: none"> <li>● 0b0: Low level active</li> <li>● 0b1: High level active</li> </ul>                                                                | RW     |
| 9     | SS1PV           | Polarity setting of SS1 signal              | <ul style="list-style-type: none"> <li>● 0b0: Low level active</li> <li>● 0b1: High level active</li> </ul>                                                                | RW     |
| 8     | SS0PV           | Polarity setting of SS0 signal              | <ul style="list-style-type: none"> <li>● 0b0: Low level active</li> <li>● 0b1: High level active</li> </ul>                                                                | RW     |
| 7     | <i>Reserved</i> | <i>Reserved</i>                             | -                                                                                                                                                                          | -      |
| 6     | SPRDTD          | Data register read target selection         | <ul style="list-style-type: none"> <li>● 0b0: SPI_DR read receive buffer</li> <li>● 0b1: SPI_DR read transmit buffer (Can only be read when TDEF=1)</li> </ul>             | RW     |
| 5     | <i>Reserved</i> | <i>Reserved</i>                             | -                                                                                                                                                                          | -      |
| 4     | <i>Reserved</i> | <i>Reserved</i>                             | <i>Reads as 1, Write as 1</i>                                                                                                                                              | -      |
| 3     | <i>Reserved</i> | <i>Reserved</i>                             | -                                                                                                                                                                          | -      |
| 2     | CTMDS           | Continuous communication mode selection     | <ul style="list-style-type: none"> <li>● 0b0: Normal communication mode</li> <li>● 0b1: Continuous communication mode</li> </ul>                                           | RW     |
| 1:0   | FTHLV           | Frames setting bit                          | <ul style="list-style-type: none"> <li>● 0b00: 1 frame</li> <li>● 0b01: 2 frames</li> <li>● 0b10: 3 frames</li> <li>● 0b11: 4 frames</li> </ul>                            | RW     |

## 31.4.5 SPI Status Register (SPI\_SR)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |    |     |    |        |    |      |    |        |    |      |    |         |    |       |    |        |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|----|-----|----|--------|----|------|----|--------|----|------|----|---------|----|-------|----|--------|
| 0x14   | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7  | 6   | 5  | 4      | 3  | 2    | 1  | 0      |    |      |    |         |    |       |    |        |
| Reset  | 0x00000020   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |    |     |    |        |    |      |    |        |    |      |    |         |    |       |    |        |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |    | RDF |    | FMIDLF |    | TDEF |    | UDRERF |    | PERF |    | MODFERF |    | IDLNF |    | OVRERF |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   | RO | RDF | RO | FMIDLF | RO | TDEF | RW | UDRERF | RW | PERF | RW | MODFERF | RO | IDLNF | RW | OVRERF |

| Bit  | Name     | Function                                      | Description                                                                                                                                                                                                                                                                                                                                                                     | Access |
|------|----------|-----------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:8 | Reserved | Reserved                                      | -                                                                                                                                                                                                                                                                                                                                                                               | -      |
| 7    | RDF      | Receive buffer full flag                      | Hardware set and clear <ul style="list-style-type: none"> <li>0b0: The receive buffer is not full</li> <li>0b1: The receive buffer is full</li> </ul>                                                                                                                                                                                                                           | RO     |
| 6    | FMIDLF   | IDLE status flag of master in continuous mode | Hardware set and clear <ul style="list-style-type: none"> <li>0b0: Continuous mode master is in normal operation state</li> <li>0b1: Continuous mode master is in idle state</li> </ul>                                                                                                                                                                                         | RO     |
| 5    | TDEF     | Transmit buffer empty flag                    | Hardware set and clear <ul style="list-style-type: none"> <li>0b0: Data present in transmit buffer</li> <li>0b1: No data present in the transmit buffer</li> </ul>                                                                                                                                                                                                              | RO     |
| 4    | UDRERF   | Underload error flag                          | Hardware set; read this bit as 1 and then write 0 to clear the status bit <ul style="list-style-type: none"> <li>0b0: No underload error occurred</li> <li>0b1: Underload error occurred (MODFERF=1)</li> </ul> <div>  <b>Note</b><br/> When MODFERF=0, this bit will be initialized. </div> | RW     |
| 3    | PERF     | Parity error flag                             | Hardware set; read this bit as 1 and then write 0 to clear the status bit <ul style="list-style-type: none"> <li>0b0: No parity error occurred</li> <li>0b1: Parity error occurred</li> </ul>                                                                                                                                                                                   | RW     |
| 2    | MODFERF  | Mode fault error flag                         | Hardware set; read this bit as 1 and then write 0 to clear the status bit <ul style="list-style-type: none"> <li>0b0: No mode fault error occurred</li> <li>0b1: Mode fault error occurred</li> </ul>                                                                                                                                                                           | RW     |
| 1    | IDLNF    | SPI idle flag                                 | Hardware set and clear <ul style="list-style-type: none"> <li>0b0: SPI is in idle state</li> <li>0b1: SPI is in transmission state</li> </ul>                                                                                                                                                                                                                                   | RO     |

| Bit | Name   | Function            | Description                                                                                                                                                                                        | Access |
|-----|--------|---------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 0   | OVRERF | Overload error flag | Hardware set; read this bit as 1 and then write 0 to clear the status bit <ul style="list-style-type: none"><li>● 0b0: No overload error occurred</li><li>● 0b1: Overload error occurred</li></ul> | RW     |

## 31.4.6 SPI Communication Configuration Register 2 (SPI\_CFG2)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |       |        |       |      |       |    |   |   |     |   |          |     |   |      |      |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-------|--------|-------|------|-------|----|---|---|-----|---|----------|-----|---|------|------|---|
| 0x18   | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15    | 14     | 13    | 12   | 11    | 10 | 9 | 8 | 7   | 6 | 5        | 4   | 3 | 2    | 1    | 0 |
| Reset  | 0x00000F1D   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |       |        |       |      |       |    |   |   |     |   |          |     |   |      |      |   |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | MSSIE | MSSDLE | MIDIE | LSBF | DSIZE |    |   |   | SSA |   | Reserved | MBR |   | CPOL | CPHA |   |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RW    | RW     | RW    | RW   | RW    |    |   |   | RW  |   |          | RW  |   | RW   | RW   |   |

| Bit   | Name     | Function                      | Description                                                                                                                                                                                                                                                                                                                                                                                                                                        | Access |
|-------|----------|-------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:16 | Reserved | Reserved                      | -                                                                                                                                                                                                                                                                                                                                                                                                                                                  | -      |
| 15    | MSSIE    | SCK delay enabled             | <ul style="list-style-type: none"> <li>0b0: SCK delay is 1 SCK period</li> <li>0b1: SCK delay is the value set by MSSIE</li> </ul>                                                                                                                                                                                                                                                                                                                 | RW     |
| 14    | MSSDLE   | SS invalid delay enabled      | <ul style="list-style-type: none"> <li>0b0: SS invalid delay is 1 SCK</li> <li>0b1: SS invalid delay is the value set by MSSDLE</li> </ul>                                                                                                                                                                                                                                                                                                         | RW     |
| 13    | MIDIE    | SPI next access delay enabled | <ul style="list-style-type: none"> <li>0b0: The next access delay is 1 SCK+2 PCLK1</li> <li>0b1: The next access delay is the value set by MIDI</li> </ul>                                                                                                                                                                                                                                                                                         | RW     |
| 12    | LSBF     | SPI LSB first bit             | <ul style="list-style-type: none"> <li>0b0: MSB first</li> <li>0b1: LSB first</li> </ul>                                                                                                                                                                                                                                                                                                                                                           | RW     |
| 11:8  | DSIZE    | SPI data length setting bit   | <ul style="list-style-type: none"> <li>0b0000: 4 bits</li> <li>0b0001: 5 bits</li> <li>0b0010: 6 bits</li> <li>0b0011: 7 bits</li> <li>0b0100: 8 bits</li> <li>0b0101: 9 bits</li> <li>0b0110: 10 bits</li> <li>0b0111: 11 bits</li> <li>0b1000: 12 bits</li> <li>0b1001: 13 bits</li> <li>0b1010: 14 bits</li> <li>0b1011: 15 bits</li> <li>0b1100: 16 bits</li> <li>0b1101: 20 bits</li> <li>0b1110: 24 bits</li> <li>0b1111: 32 bits</li> </ul> | RW     |
| 7:5   | SSA      | SS signal enabled setting bit | <ul style="list-style-type: none"> <li>0b000: SS0</li> <li>0b001: SS1</li> <li>0b010: SS2</li> <li>0b011: SS3</li> <li>Others: Setting prohibited</li> </ul>                                                                                                                                                                                                                                                                                       | RW     |
| 4     | Reserved | Reserved                      | Reads as 1, Write as 1                                                                                                                                                                                                                                                                                                                                                                                                                             | -      |

| Bit | Name | Function                                | Description                                                                                                                                                                                                                                                                                                                  | Access |
|-----|------|-----------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 3:2 | MBR  | Bit rate frequency division setting bit | <ul style="list-style-type: none"> <li>● 0b00: Select the reference communication clock</li> <li>● 0b01: Select the 2-division of the reference communication clock</li> <li>● 0b10: Select the 4-division of the base communication clock</li> <li>● 0b11: Select the 8-division of the base communication clock</li> </ul> | RW     |
| 1   | CPOL | SCK polarity setting bit                | <ul style="list-style-type: none"> <li>● 0b0: SCK is low level when idle</li> <li>● 0b1: SCK is high level when idle</li> </ul>                                                                                                                                                                                              | RW     |
| 0   | CPHA | SCK phase setting bit                   | <ul style="list-style-type: none"> <li>● 0b0: Data sampling at odd edges, changing at even edges</li> <li>● 0b1: Data changing at odd edges, sampling at even edges</li> </ul>                                                                                                                                               | RW     |

## 32 Quad Serial Peripheral Interface (QSPI)

### 32.1 Introduction

The Quad Serial Peripheral Interface (QSPI) is a memory control module designed to communicate with serial ROMs with an SPI-compatible interface. Its objects mainly include serial flash memory, serial EEPROM and serial FeRAM.

### 32.2 Features

Table 32-1 QSPI Main Specifications

| Parameters                    | Specifications                                                                                                                                                                                            |
|-------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Number of Channels            | 1 channel                                                                                                                                                                                                 |
| SPI                           | Support multiple protocols such as extended SPI, two-wire SPI and four-wire SPI                                                                                                                           |
|                               | SPI mode 0 and SPI mode 3 are supported                                                                                                                                                                   |
|                               | Address line width can be selected from 8 bits/16 bits/24 bits/32 bits                                                                                                                                    |
| Timing Adjustment             | Timing adjustable to support various serial flash                                                                                                                                                         |
| Flash Read                    | Support multiple reading methods:<br>1. Standard Read/Fast Read<br>2. Two-wire output fast read/two-wire input and output fast read<br>3. Four-wire output fast read/four-wire input and output fast read |
|                               | Free setting instruction                                                                                                                                                                                  |
|                               | Adjustable number of dummy periods                                                                                                                                                                        |
|                               | 16-byte read prefetch function                                                                                                                                                                            |
|                               | Status query function                                                                                                                                                                                     |
|                               | SPI bus period extension function                                                                                                                                                                         |
|                               | XIP control Function                                                                                                                                                                                      |
| Direct Communication Function | Flexible and extensive support for a large number of serial flash software control instructions, including erasing, writing, ID reading and power down control, etc.                                      |
| Interrupt Source              | Hardware error interrupt                                                                                                                                                                                  |
| Module Stop Function          | Power consumption can be reduced by module stopping                                                                                                                                                       |

## 32.3 Memory Map

### 32.3.1 Internal Bus Space

The location of the serial flash memory and related control registers in the AHB bus space is determined by the overall address range configuration.

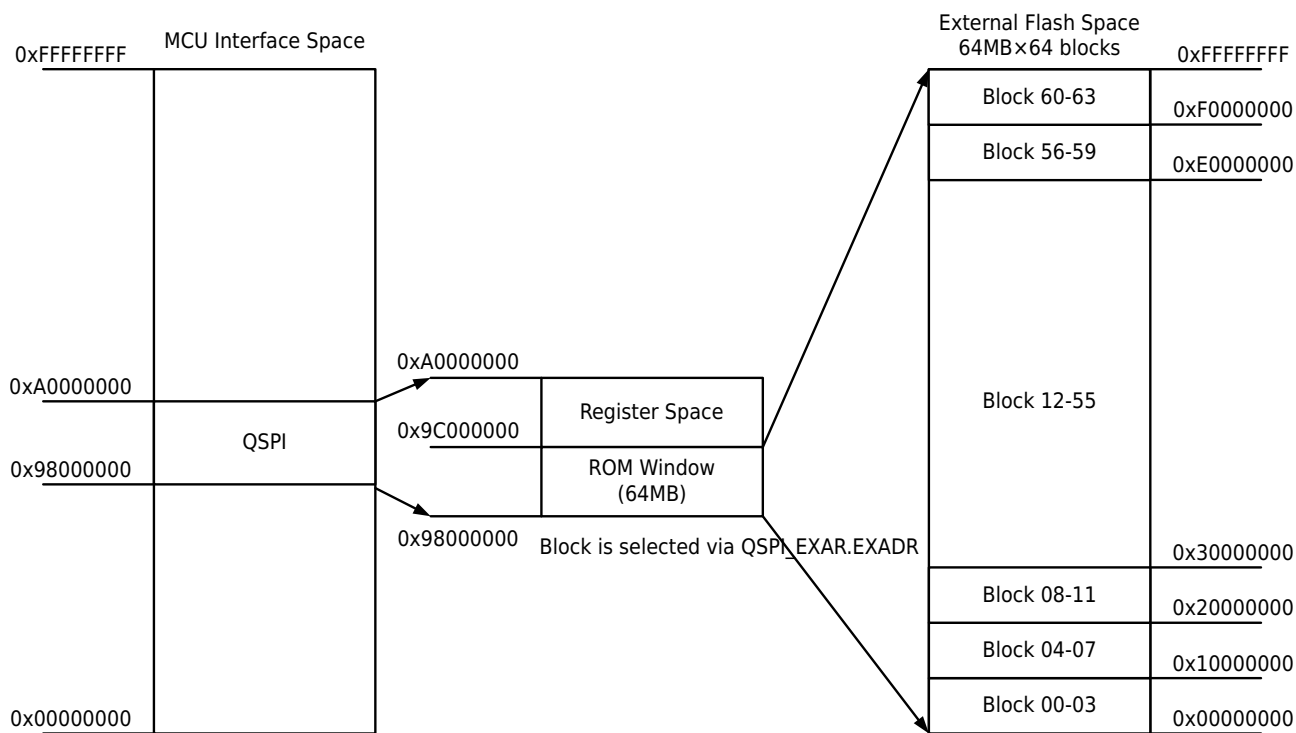


Figure 32-1 Default Area Setting and AHB Bus Space Memory Mapping Relationship Diagram

### 32.3.2 ROM Space and Bus Address Width

QSPI has a 32-bit address bus width to match serial flash memory. Whenever the ROM space of QSPI is read and accessed, the QSPI bus starts to work automatically and transmits the data read from the serial flash memory.

QSPI can not only use 32-bit address bus width, but also select 8-bit/16-bit/24-bit address bus width by setting QSPI\_FCR.AWSL bit.

If the address bus width of 8-bit/16-bit/24-bit is selected, only the low-order space whose address matches it can be accessed normally, that is, accessing the high-order serial flash image space in QSPI will repeatedly appear in the low-order space content.

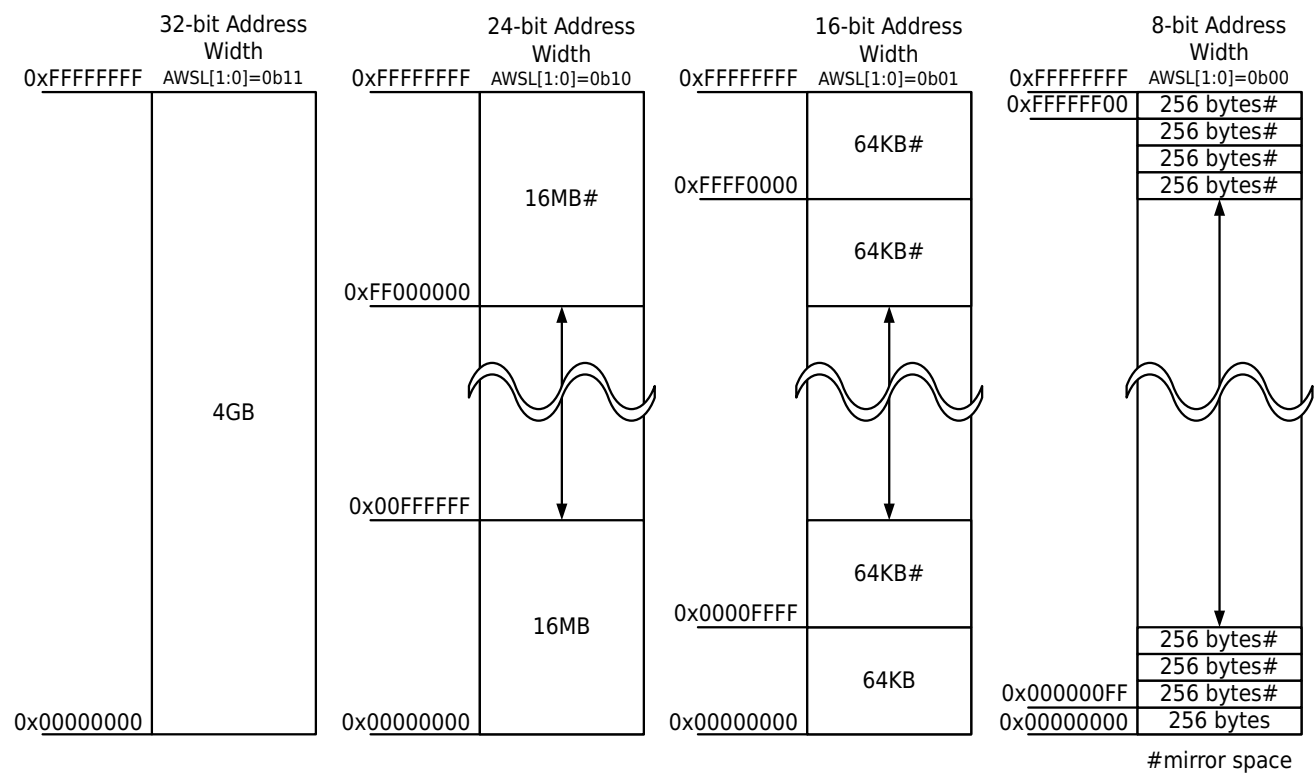


Figure 32-2 QSPI-ROM Space Memory Image



**Note**

- The address bus width can be selected to use 8 bits/16 bits/24 bits/32 bits by setting QSPI\_FCR.AWSL bit.
- When an 8-bit address width is selected, standard read instructions only support access to the lower 256-byte space, and do not support access to the mirror space.



## 32.4 Functional Description

### 32.4.1 Basic Block Diagram

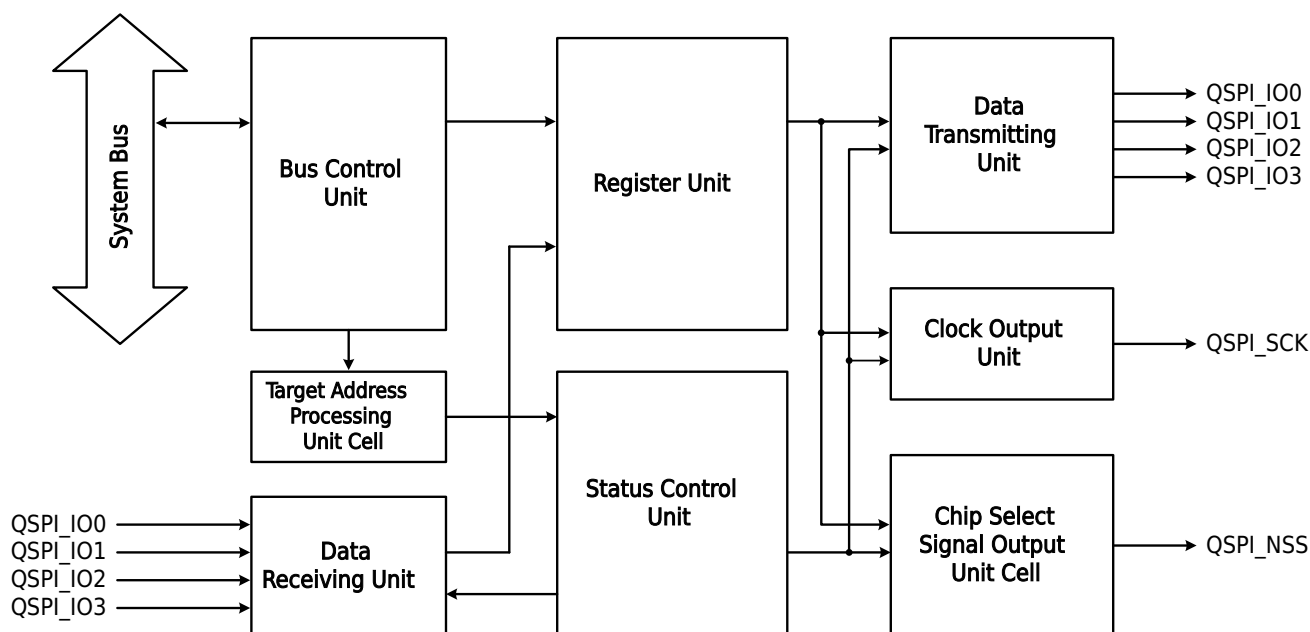


Figure 32-3 Module Composition Diagram of QSPI

Table 32-2 QSPI pin

| Pin Name | Input/Output | Functional description |
|----------|--------------|------------------------|
| QSPI_SCK | O            | QSPI clock output pin  |
| QSPI_NSS | O            | QSPI slave select pin  |
| QSPI_IO0 | I/O          | Data line 0            |
| QSPI_IO1 | I/O          | Data line 1            |
| QSPI_IO2 | I/O          | Data line 2            |
| QSPI_IO3 | I/O          | Data line 3            |



#### Note

QSPI\_SCK, QSPI\_NSS, QSPI\_IO0, QSPI\_IO1, QSPI\_IO2, and QSPI\_IO3 are hereinafter referred to as QSCK, QSSN, QSI00, QSI01, QSI02, and QSI03 respectively.

### 32.4.2 QSPI Bus

#### 32.4.2.1 SPI Protocol

This QSPI supports three protocols: extended SPI, two-wire SPI and four-wire SPI. The initial default protocol is the extended SPI protocol. The extended SPI protocol only uses the QSI00 pin single-wire for instruction output, and the subsequent addresses and data are output in single-wire/two-wire/four-wire according to the specific read mode instructions.

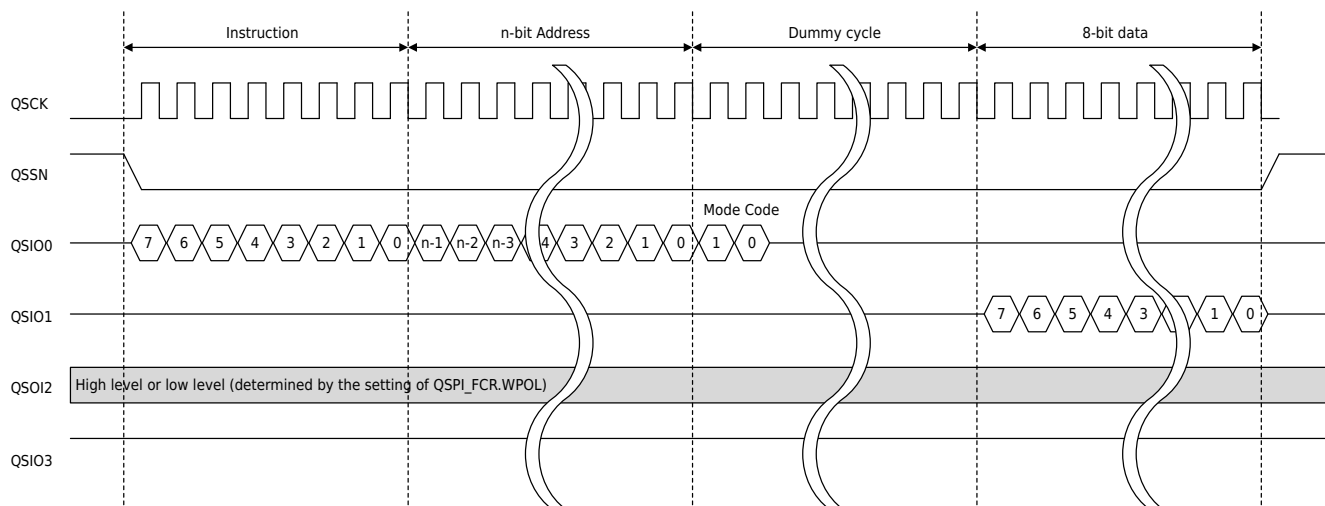


Figure 32-4 Diagram of Extended SPI Protocol Action 1 (Fast Read Mode)

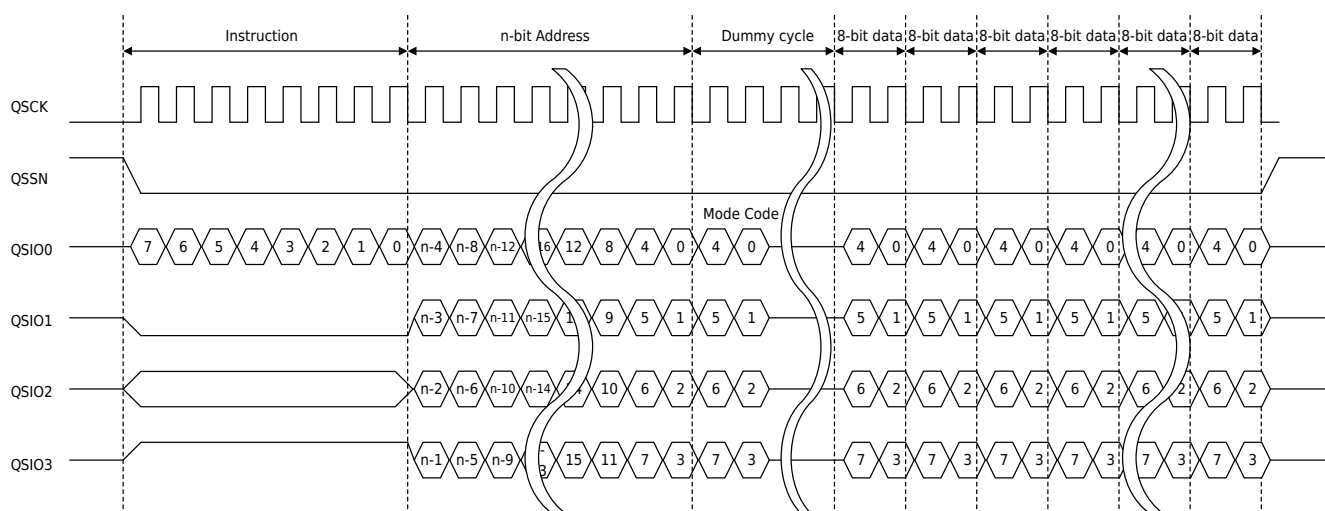


Figure 32-5 Diagram of Extended SPI Protocol Action 2 (Four-Wire Input/Output Fast Read Mode)

When the QSPI read mode is selected as custom standard read or custom fast read, the protocol of each stage can be configured separately by setting the QSPI\_CR.DPRSL/APRSL/IPRSL bits.

The two-wire SPI protocol uses the two pins QSPI0 and QSPI1 to perform corresponding operations, including sending instructions, addresses, receiving data, and so on.

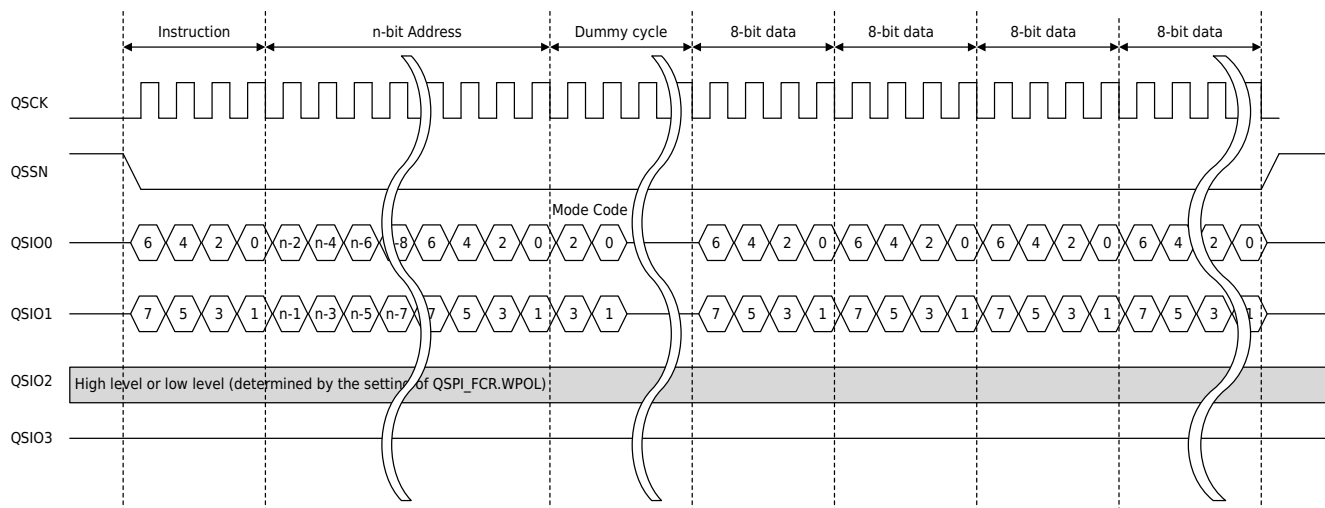


Figure 32-6 Diagram of Two-wire SPI Protocol Action (Fast Read Mode)

The four-wire SPI protocol uses four pins QSI00, QSI01, QSI02, and QSI03 to perform all related operations such as sending instructions, addressing, and receiving data.

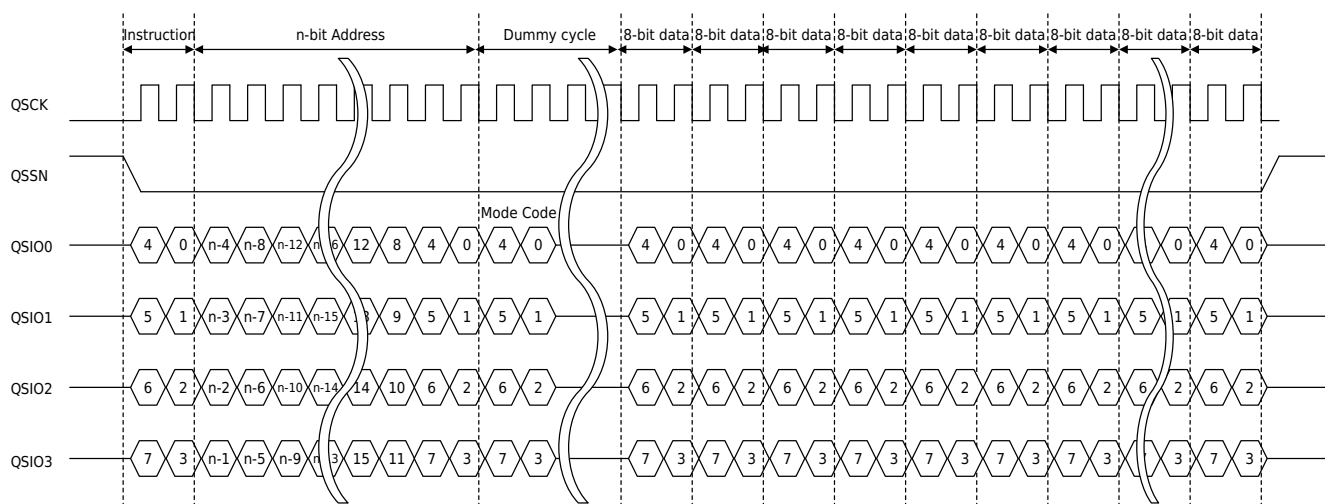


Figure 32-7 Diagram Four-Wire SPI Protocol Action (Fast Read Mode)

### 32.4.2.2 SPI Mode

There are two SPI modes, Mode 0 and Mode 3, and the mode switching can be realized by setting the QSPI\_CR.SPIMD3 bit. The difference between SPI mode 0 and mode 3 is the different levels of QCK in the idle state. In SPI mode 0, the idle state level of QCK is low level, while in mode 3, the idle state level is high level.

Serial data is output from the QSPI on the falling edge of the serial clock and read into the external flash memory on the rising edge. The external flash memory outputs serial data at the falling edge of the serial clock and is read in by QSPI at the falling edge of the next clock.

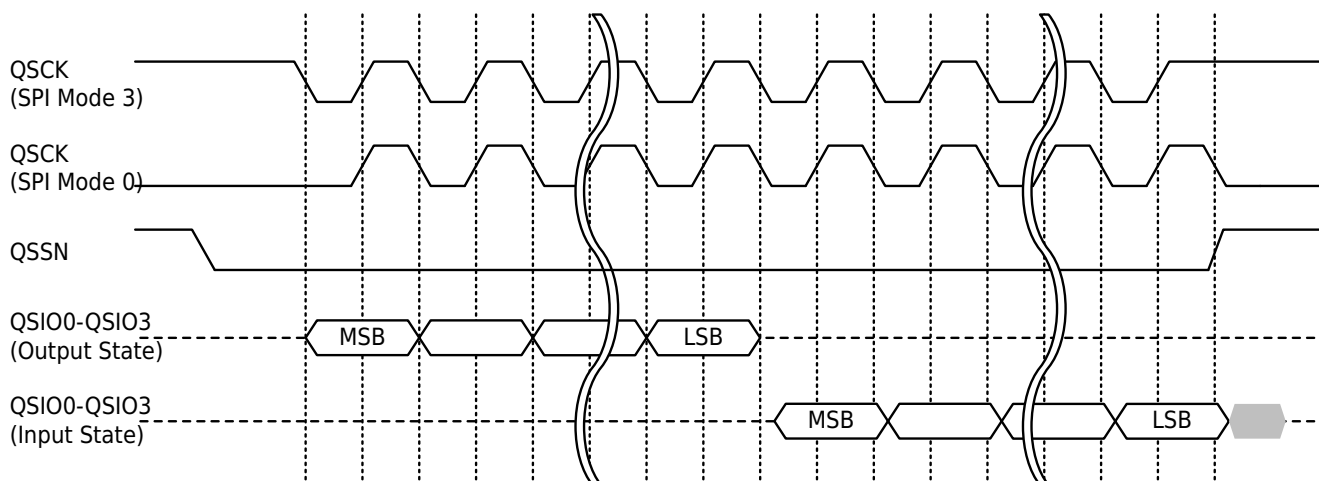


Figure 32-8 Basic Timing Diagram of Serial Interface

### 32.4.3 Timing Adjustment of QSPI Bus

The timing of QSPI bus signals can be fine-tuned through registers, and the timing after fine-tuning can be better applied to various QSPI bus accesses, whether it is ROM access mode or direct communication mode.

### 32.4.3.1 QSPI Bus Reference Clock

The reference clock of the QSPI bus is obtained by HCLK after frequency division. By setting the QSPI\_CR.DIV bit, various clock sources such as HCLK divided by 2 to 64 can be selected as the reference clock of the QSPI bus.

### 32.4.3.2 QSPI Communication Clock Duty Cycle Correction

When the reference clock selects an even multiple frequency division of HCLK, the high and low level times of the QSCK signal are consistent. If an odd multiple frequency division is selected, the high level time of the QSCK signal will be one HCLK period longer than the low level.

If you want QSCK to output a clock signal with a duty cycle of about 50% when selecting an odd frequency division, you can set the QSPI\_FCR.DUTY bit to 1. Through this setting, the output time of the rising edge of the QSCK signal will be half a HCLK period later than before the adjustment, and the output time of the falling edge remains unchanged. Thus, a QSCK signal with a duty cycle of 50% can be obtained. When the base clock is divided by an even multiple of HCLK, please set the DUTY bit to 0. In the initial state, the DUTY bit defaults to 1.

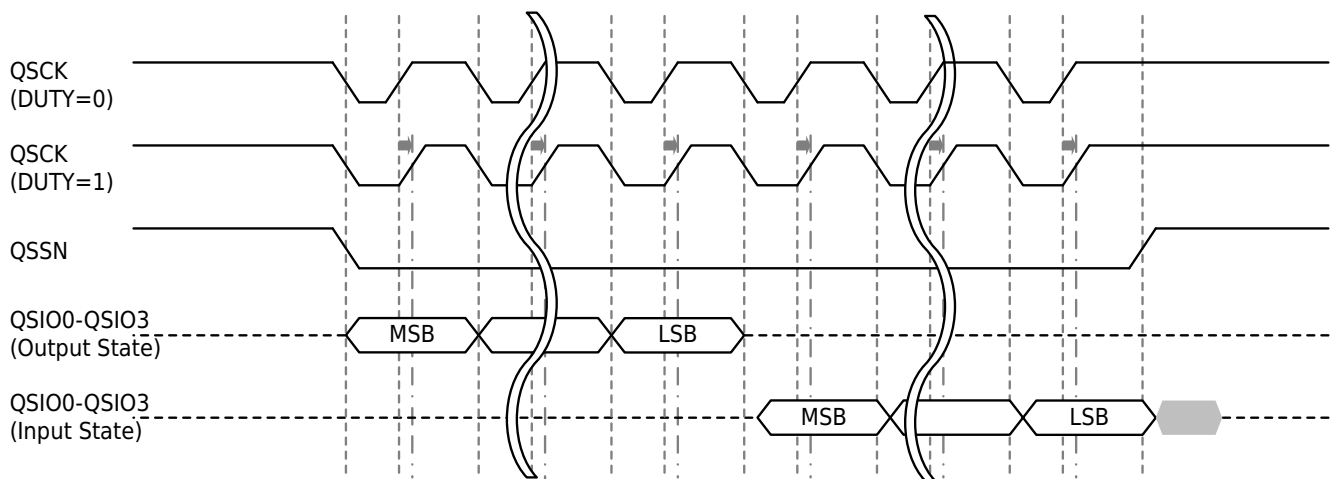


Figure 32-9 Diagram of Output Clock Duty Cycle Correction when HCLK is Divided by 3 as the Reference Clock

### 32.4.3.3 QSSN Signal Minimum High-level width

In order to meet the cancellation time requirement required by the serial flash memory, the QSSN signal must maintain a high level state (that is, an idle state) for a long enough time between two adjacent QSPI bus periods. The minimum high width of QSSN can be selected by setting the QSPI\_CSCR.SSHW bits, ranging from 1 to 16 QSPI reference clock periods.

### 32.4.3.4 QSSN Establishment Time

The time from when the QSSN signal starts to output a low level (that is, becomes a active state) to when the QSCK signal outputs the first edge is called the QSSN setup time, which can be configured through register settings to meet the requirements of the external serial flash memory. Set the QSPI\_FCR.SSNLD bit to select whether the QSSN setup time is 0.5 or 1.5 QSPI reference clock periods. This setting can also be used to configure the setup time of the data pin from data output permission to the first edge of the QSCK signal output, which can be used reasonably as needed.

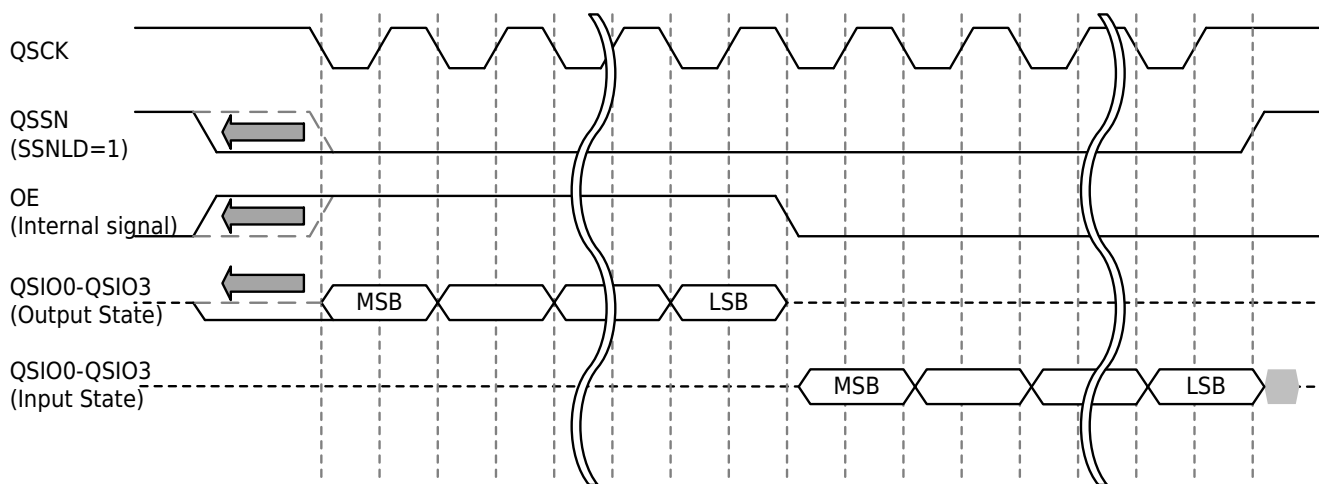


Figure 32-10 QSSN Setup Time Configuration Diagram

### 32.4.3.5 QSSN Hold Time

The time between the last rising edge of QSCCK signal output and the start of QSSN signal output high level (that is, becomes idle state) is called the hold time of QSSN, which can be configured through register settings to meet the requirements of external devices. Set the QSPI\_FCR.SSNHD bit to select whether the QSSN hold time is 0.5 or 1.5 QSPI reference clock periods.

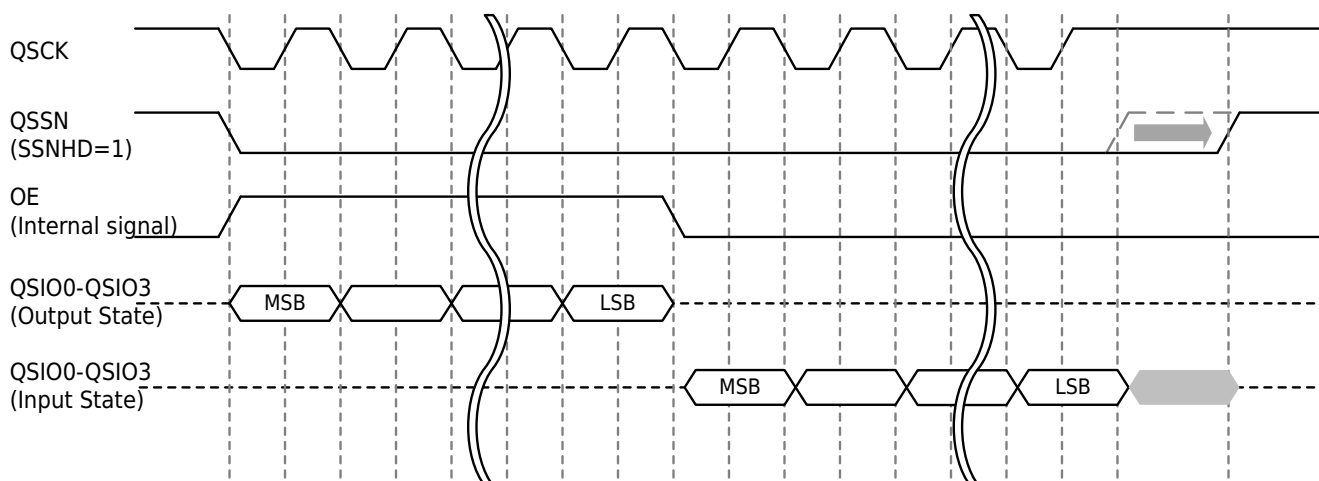


Figure 32-11 QSSN Hold Time Configuration Diagram

### 32.4.3.6 Serial Data Receive Delay

The data of the serial flash memory is output synchronously with the falling edge of QSCCK, and the QSPI receives the data on the next falling edge of QSCCK. The period from when the serial flash starts outputting data to when the data is received by QSPI is called receive latency. QSPI adds a delay adjustment period before the first data receive period. From the perspective of serial flash memory, this period can be regarded as an increase in the counterparty data receiving period. This delay adjustment period will only be generated when data is received.

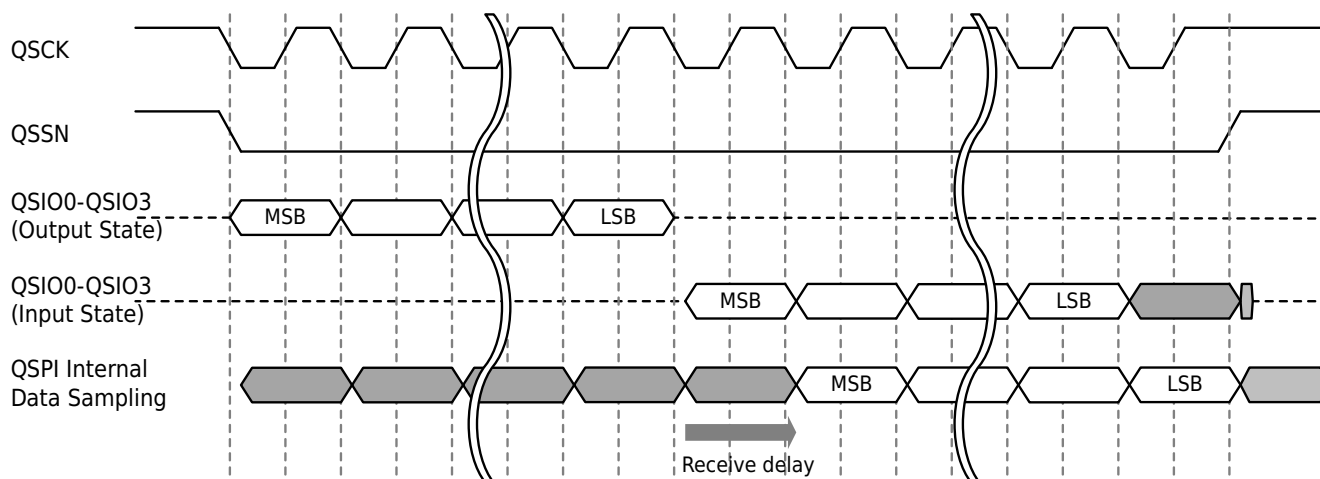


Figure 32-12 Diagram of Data Receiving Delay

### 32.4.4 Introduction to SPI Instructions for ROM Access

#### 32.4.4.1 Existing QSPI-ROM Instruction Reference

Table 32-3 Reference Instruction List

| Mode Name            |                                     | Instruction Code | Description |
|----------------------|-------------------------------------|------------------|-------------|
| 4-Bytes Address Mode | Standard read                       | 0x13             | -           |
|                      | Fast read                           | 0x0C             | -           |
|                      | Two-wire output fast read           | 0x3C             | -           |
|                      | Two-wire input and output fast read | 0xBC             | -           |
|                      | Four-wire output fast read          | 0x6C             | -           |
|                      | Four-wire input output fast read    | 0xEC             | -           |
|                      | Exit 4-Bytes mode                   | 0xE9             | -           |
| 3-Bytes Address Mode | Standard read                       | 0x03             | -           |
|                      | Fast read                           | 0x0B             | -           |
|                      | Two-wire output fast read           | 0x3B             | -           |
|                      | Two-wire input and output fast read | 0xBB             | -           |
|                      | Four-wire output fast read          | 0x6B             | -           |
|                      | Four-wire input output fast read    | 0xEB             | -           |
|                      | Enter 4-Bytes mode                  | 0xB7             | -           |
| -                    | Write mode                          | 0x06             | -           |

When the serial flash memory performs "custom standard read" and "custom fast read" access, the instruction needs to be set through the instruction register QSPI\_CCMD.

#### 32.4.4.2 Standard Read Instruction

The standard read instruction is a common read instruction supported by most serial flash memory. When a serial bus period starts, the serial flash select signal is set to the active state, and then QSPI outputs the instruction code (0x03/0x13) of the instruction, followed by the target address. The width of the address can be set by the QSPI\_FCR.AWSL bit. Then the data can be received. The instruction selected by the initial state of QSPI is the standard read instruction.

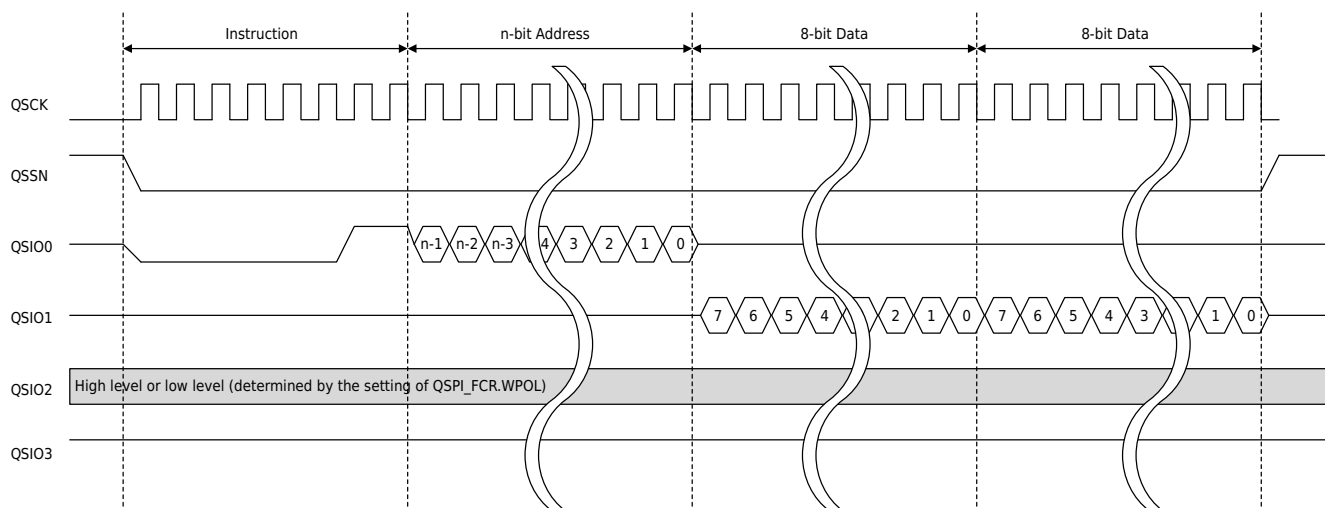


Figure 32-13 Diagram of a Standard Read Bus Cycle

#### 32.4.4.3 Fast Read Instruction

The fast read instruction is a read instruction that supports a faster communication clock. When a serial bus period starts, the serial flash select signal is set to the active state, and then QSPI outputs the instruction code (0x0B/0x0C) of the instruction, followed by the target address. The address width can be set by the QSPI\_FCR.AWSL bit. The address output is followed by a certain number of dummy periods, the specific number of which is determined by QSPI\_FCR.DMCYCN bit. Next is the reception of data.

The first two periods of the dummy period are used to decide whether to select XIP mode. When XIP mode is selected, the instruction used in this transmission will be applied to the next SPI bus period, and the instruction transmission part will be omitted in the next SPI bus period. For details, see "XIP Control".

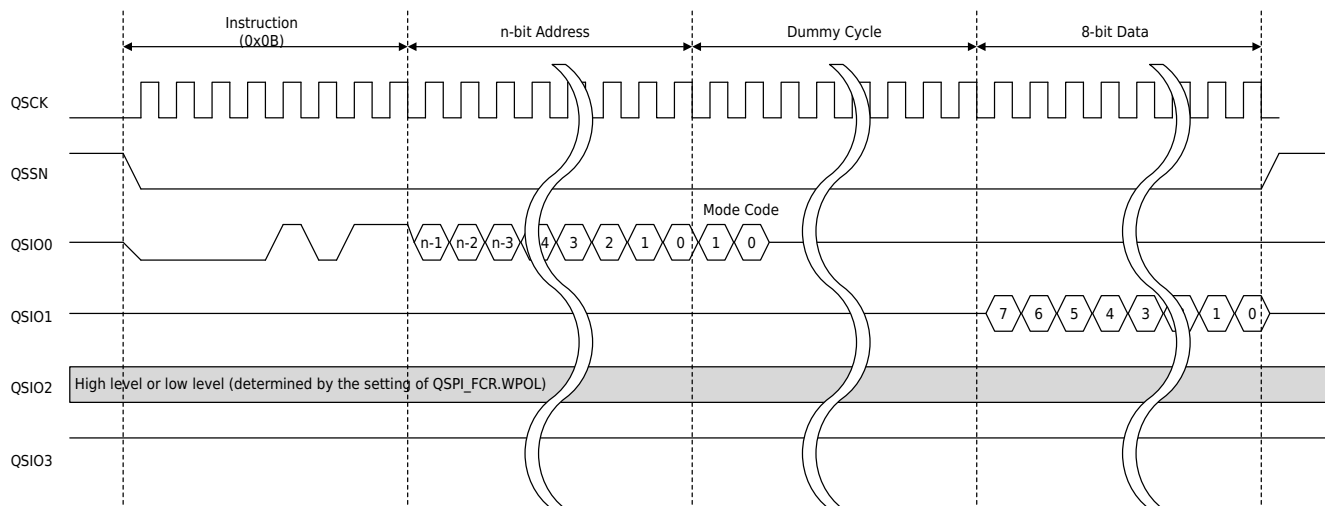


Figure 32-14 Fast Read Bus Cycle Diagram

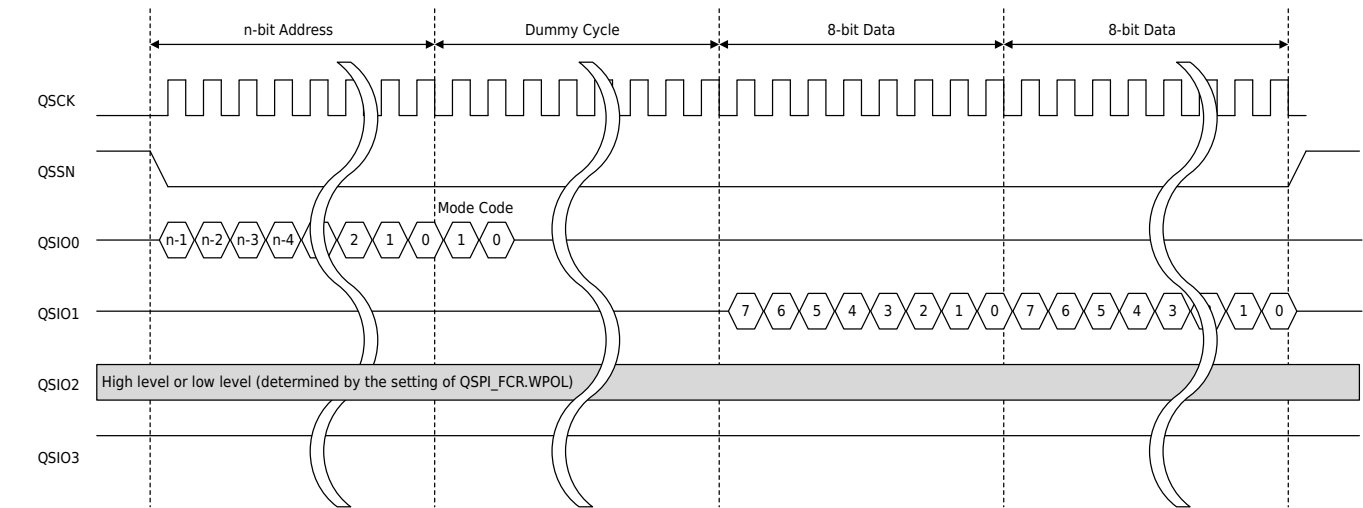


Figure 32-15 Fast Read Bus Cycle for Selecting XIP Mode Diagram



**Note**

To use the fast read instruction, make sure to use a serial flash that supports the fast read function.



### 32.4.4.4 Two-wire Output Fast Read Instruction

Two-wire output fast read is a read instruction that uses two signal lines for data reception. When a serial bus period starts, the serial flash select signal is set to the active state, and QSPI starts to output the instruction code (0x3B/0x3C) and target address of the instruction from the QSI00 pin. The address width can be set by the QSPI\_FCR.AWSL bit. This is followed by a certain number of dummy periods, the specific number of which is determined by QSPI\_FCR.DMTCYCN bit. Then start to receive data through the two pins QSI00 and QSI01. Even-numbered bit data is received at QSI00 and odd-numbered bit at QSI01.

The first two periods of the dummy period are used to decide whether to select XIP mode. When XIP mode is selected, the instruction used in this transmission will be applied to the next QSPI bus period, and the instruction transmission part will be omitted in the next QSPI bus period. For details, see "[XIP Control](#)".

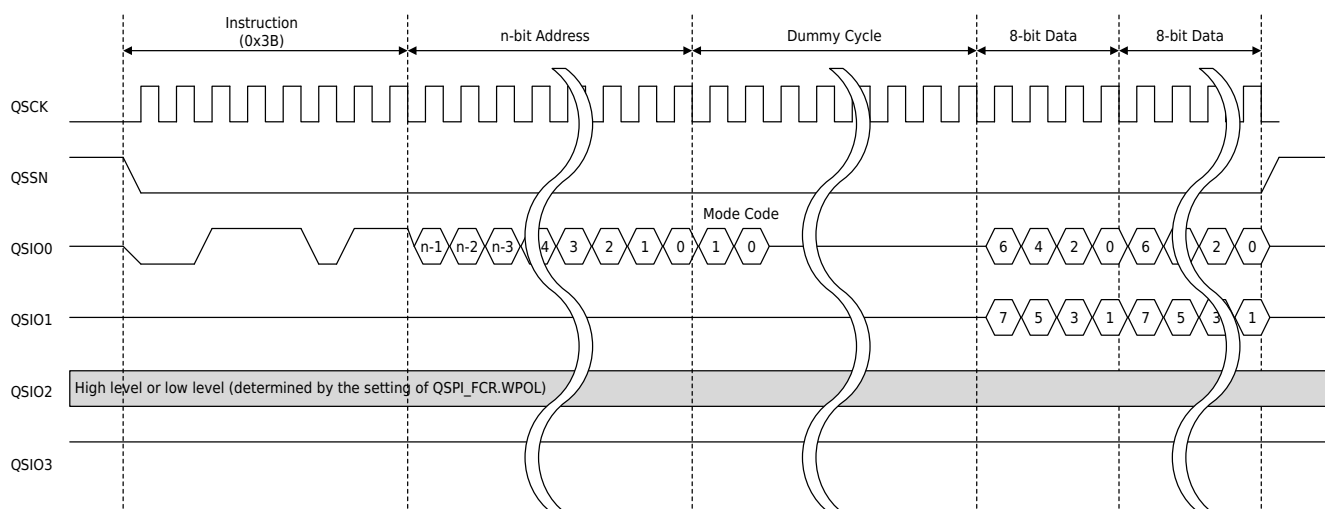


Figure 32-16 Two-Wire Output Fast Read Bus Cycle Diagram

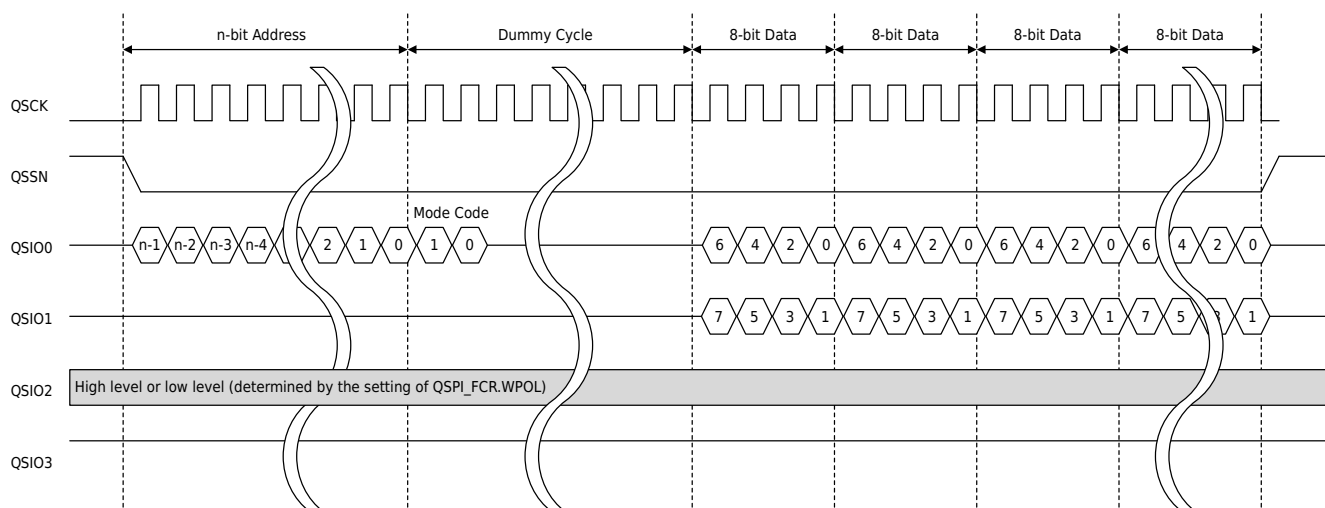


Figure 32-17 Two-Wire Output Fast Read Bus Cycle for Selecting XIP Mode Diagram



#### Note

To use the two-wire output fast read instruction, please make sure to use a serial flash memory that supports this function.

### 32.4.4.5 Two-wire Input and Output Fast Read Instruction

Two-wire input and output fast read is a read instruction that uses two signal lines for sending and data reception. When a serial bus period starts, the serial flash select signal is set to the active state, and the QSPI starts outputting the instruction code (0xBB/0xBC) of the instruction from the QSIO0 pin. After that, QSPI outputs the target address from the QSIO0 and QSIO1 pins. The address width can be set by the QSPI\_FCR.AWSL bit. This is followed by a certain number of dummy periods, the specific number of which is determined by QSPI\_FCR.DMCYCN bit. Then start to receive data through the two pins QSIO0 and QSIO1. The transmission and data reception of even-numbered addresses and dummy periods (including XIP mode selection information) use QSIO0 pins, and odd-numbered bits use QSIO1 pins.

The first two periods of the dummy period are used to decide whether to select XIP mode. When XIP mode is selected, the instruction used in this transmission will be applied to the next QSPI bus period, and the instruction transmission part will be omitted in the next QSPI bus period. For details, see "[XIP Control](#)".

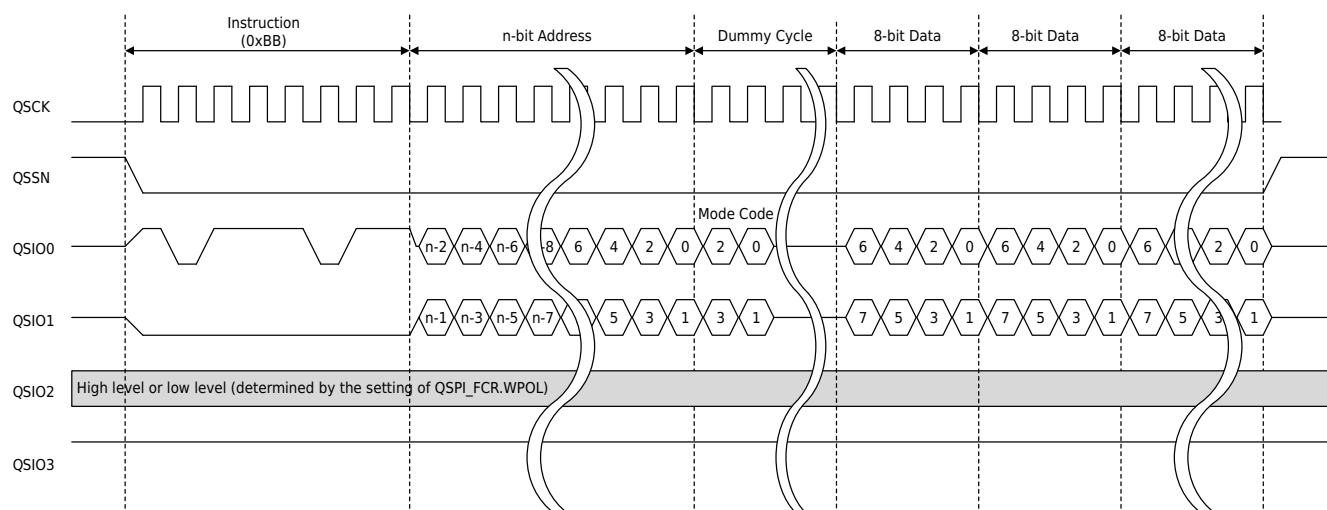


Figure 32-18 Two-Wire Input and Output Fast Read Bus Cycle Diagram

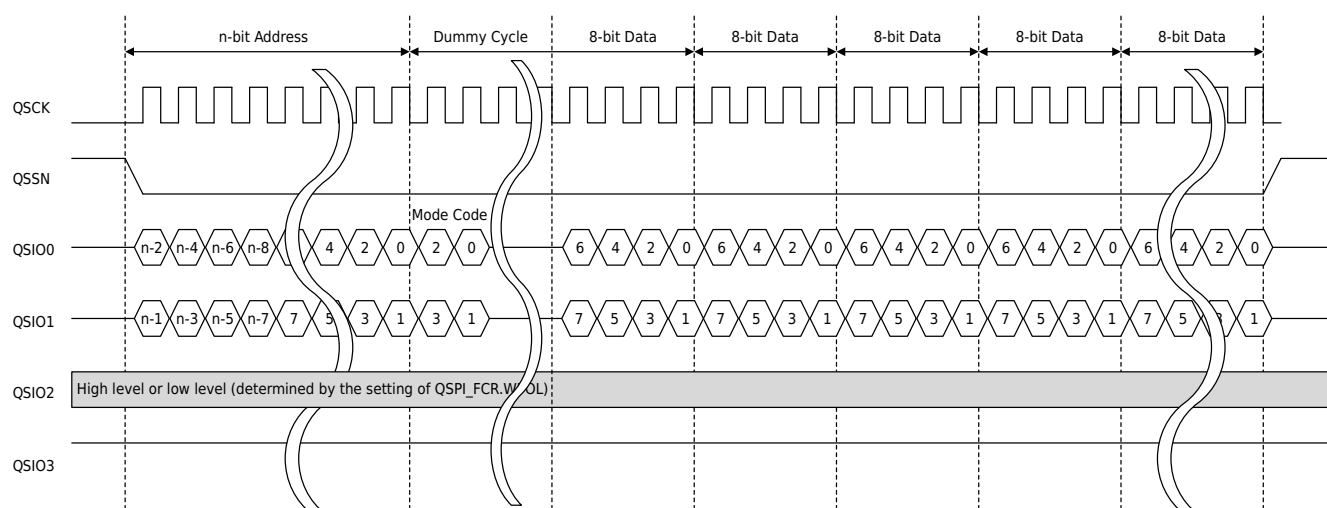


Figure 32-19 Two-Wire Input and Output Fast Read Bus Cycle for Selecting XIP Mode Diagram



#### Note

To use the two-wire input and output fast read instruction, please make sure to use a serial flash memory that supports this function.

### 32.4.4.6 Four-wire Output Fast Read Instruction

Four-wire output fast read is a read instruction that uses four signal lines for data reception. When a serial bus period starts, the serial flash select signal is set to the active state, and QSPI starts to output the instruction code (0x6B/0x6C) and target address of the instruction from the QSI00 pin. The address width can be set by the QSPI\_FCR.AWSL bit. This is followed by a certain number of dummy periods, the specific number of which is determined by QSPI\_FCR.DMCYCN bit. Then start to receive data through the four pins QSI00, QSI01, QSI02 and QSI03.

The first two periods of the dummy period are used to decide whether to select XIP mode. When XIP mode is selected, the instruction used in this transmission will be applied to the next SPI bus period, and the instruction transmission part will be omitted in the next SPI bus period. For details, see "XIP Control".

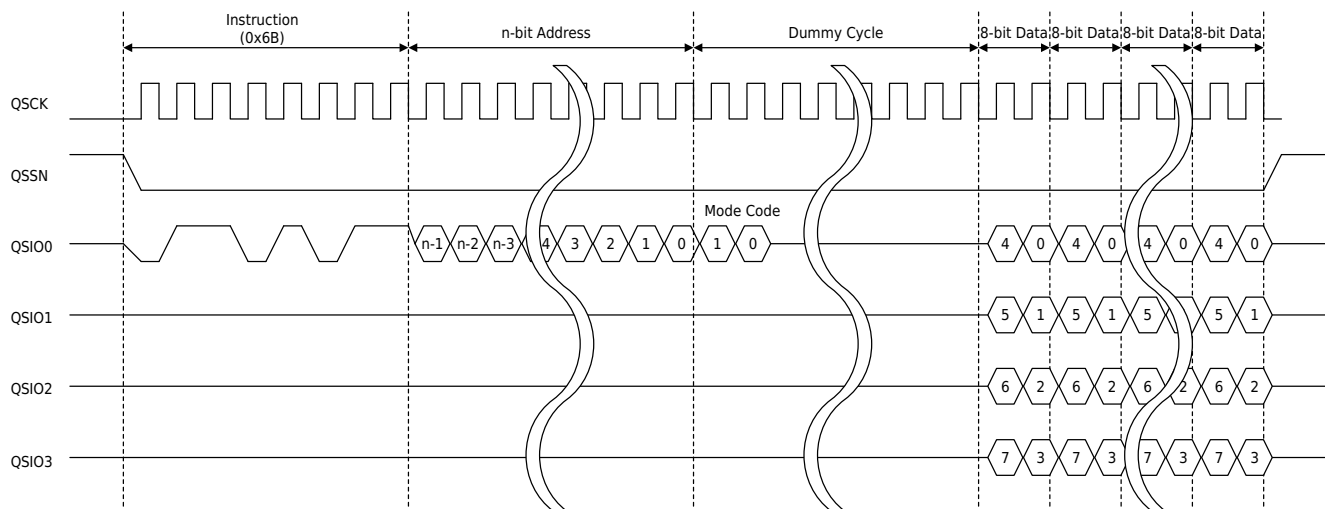


Figure 32-20 Four-Wire Output Fast Read Bus Cycle Diagram

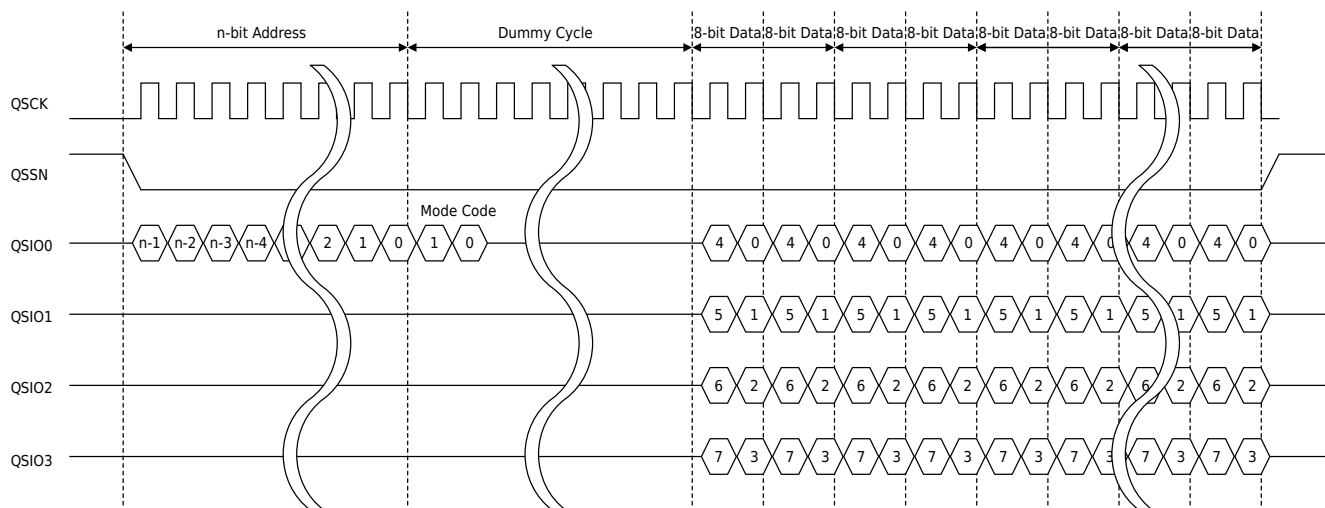


Figure 32-21 Four-Wire Output Fast Read Bus Cycle for Selecting XIP Mode Diagram



#### Note

To use the 4-wire output fast read instruction, make sure to use a serial flash that supports this feature.

### 32.4.4.7 Four-wire Input and Output Fast Read Instruction

Four-wire input and output fast read is a read instruction that uses four signal lines for address transmission and data reception. When a serial bus period starts, the serial flash select signal is set to the active state, and the QSPI starts outputting the instruction code (0xEB/0xEC) of the instruction from the QSI00 pin. After that, QSPI outputs the target address from the four pins QSI00, QSI01, QSI02 and QSI03. The address width can be set by the QSPI\_FCR.AWSL bit. This is followed by a certain number of dummy periods, the specific number of which is determined by QSPI\_FCR.DMCYCN bit. Then start to receive data through the four pins QSI00, QSI01, QSI02 and QSI03.

The first two periods of the dummy period are used to decide whether to select XIP mode. When XIP mode is selected, the instruction used in this transmission will be applied to the next QSPI bus period, and the instruction transmission part will be omitted in the next QSPI bus period. For details, see "[XIP Control](#)".

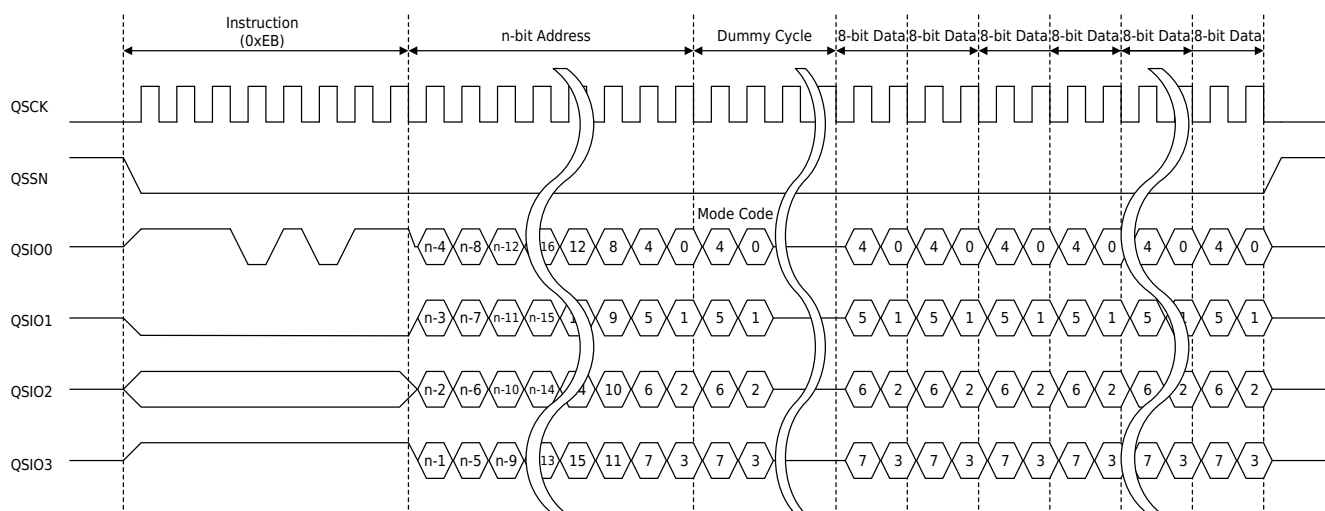


Figure 32-22 Four-Wire Input and Output Fast Read Bus Cycle Diagram

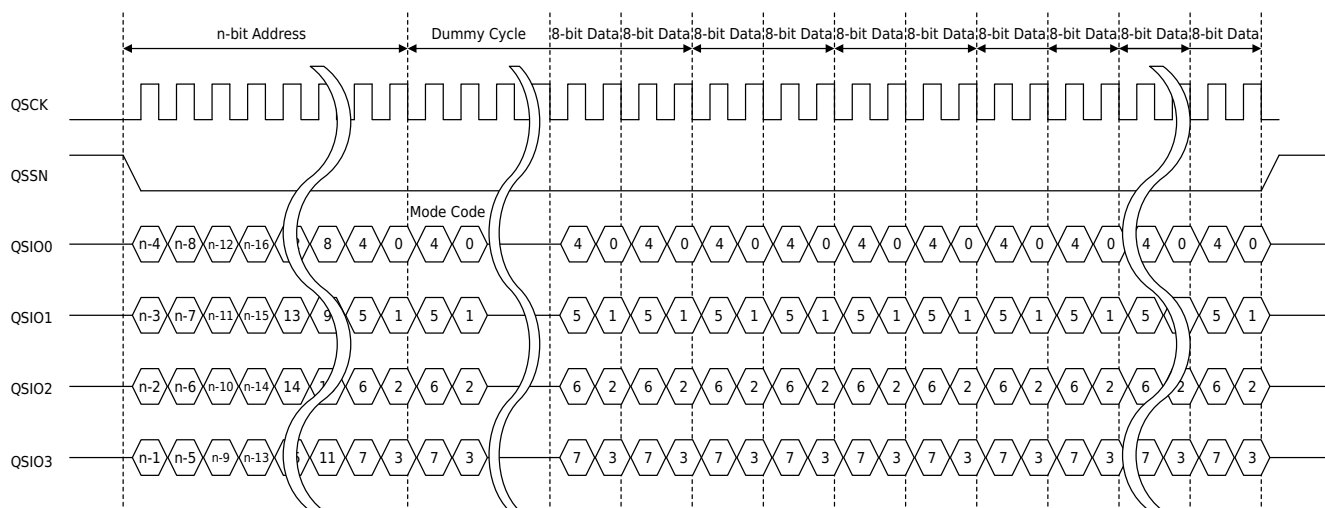


Figure 32-23 Four-Wire Input and Output Fast Read Bus Cycle for Selecting XIP Mode Diagram



#### Note

To use the four-wire input and output fast read instruction, make sure you use a serial flash memory that supports this feature.

### 32.4.4.8 Enter 4-Bytes Mode Instruction

The instruction to enter 4-Bytes mode can set the address width of the serial flash memory to 4 bytes. When a serial bus period starts, the serial flash select signal is asserted, and the QSPI starts to output the instruction code (0xB7) of the instruction from the QSI00 pin.

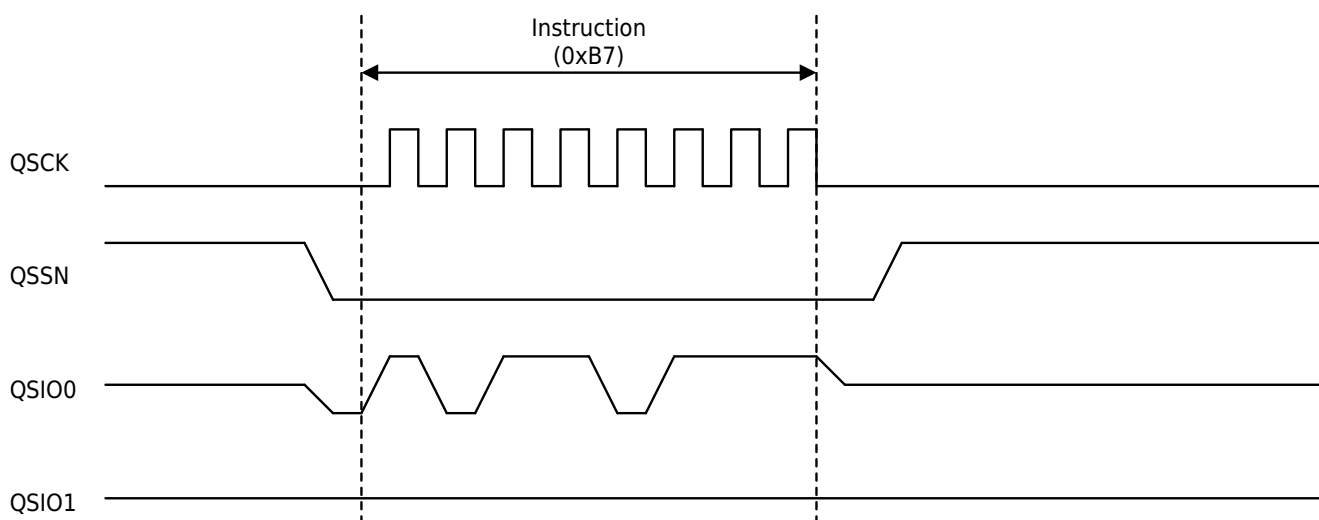


Figure 32-24 Diagram of Instruction Bus Cycle when Entering 4-Bytes Mode



#### Note

This instruction can be issued regardless of whether the serial flash memory is in 4-Bytes or 3-Bytes Mode.

### 32.4.4.9 Exit 4-Bytes Mode Instruction

The instruction to exit 4-Bytes mode can set the address width of the serial flash memory to 3 bytes. When a serial bus period starts, the serial flash select signal is set to the active state, and the QSPI starts to output the instruction code (0xE9) of the instruction from the QSI00 pin.

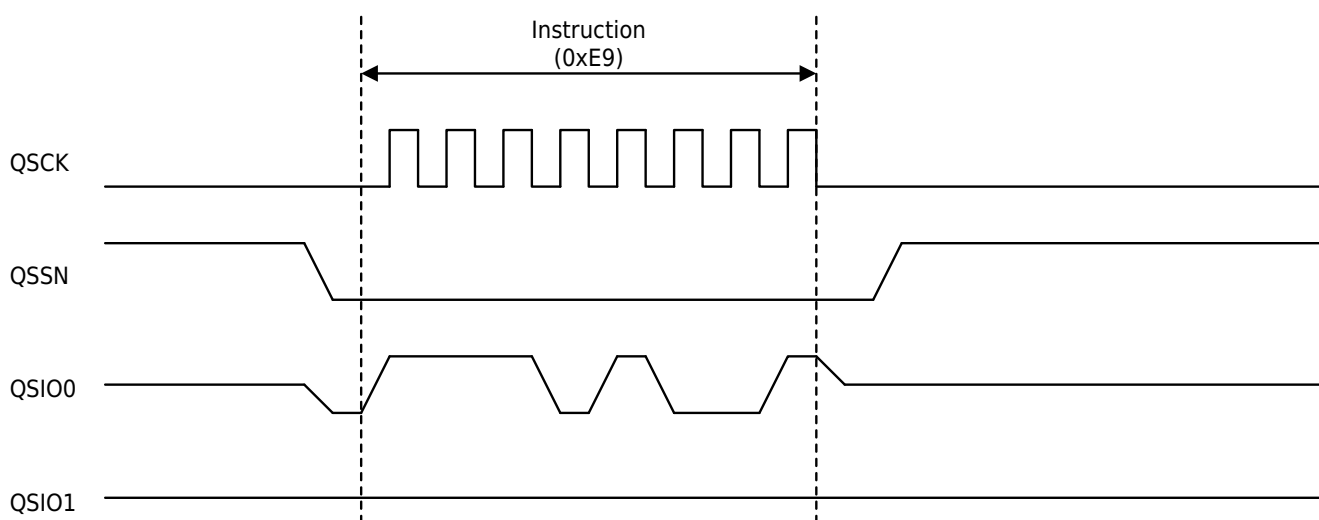


Figure 32-25 Exit 4-Byte Mode Instruction Bus Period Diagram



#### Note

This instruction can be issued regardless of whether the serial flash memory is in 4-Bytes or 3-Bytes Mode.

### 32.4.4.10 Write Mode Instruction

The write mode instruction allows the serial Flash memory to be written. When a serial bus period starts, the serial flash select signal is set to the active state, and the QSPI starts to output the instruction code (0x06) of the instruction from the QSI00 pin.

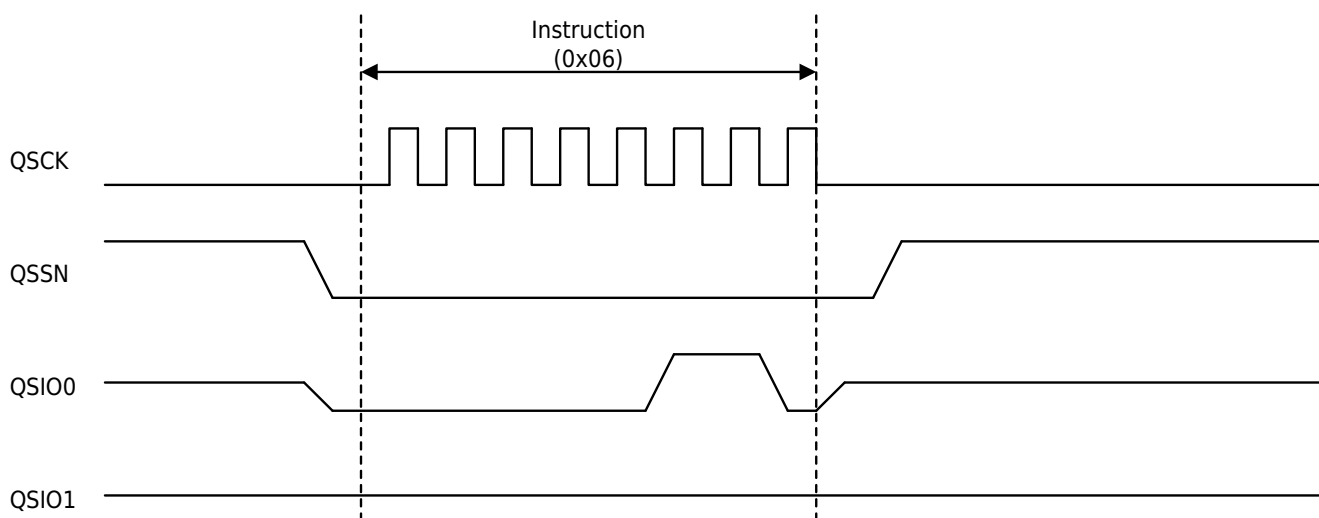


Figure 32-26 Diagram of Write Mode Instruction Bus Period

## 32.4.5 Scheduling of QSPI Bus Period

### 32.4.5.1 Single Flash Read with Independent Conversion

A single read instruction for ROM will be independently converted one-to-one from the internal bus period of the chip to the QSPI bus period. When a ROM read bus period is detected, the QSSN signal is set to the active state to initiate a QSPI bus period. After receiving the data from the serial flash memory, the QSSN signal becomes inactive, and the QSPI bus period is declared complete.

When another ROM read bus period is detected, the QSSN signal will be set to an active state again after ensuring that the inactive hold time has exceeded the minimum inactive hold width, and a new QSPI bus period begins.

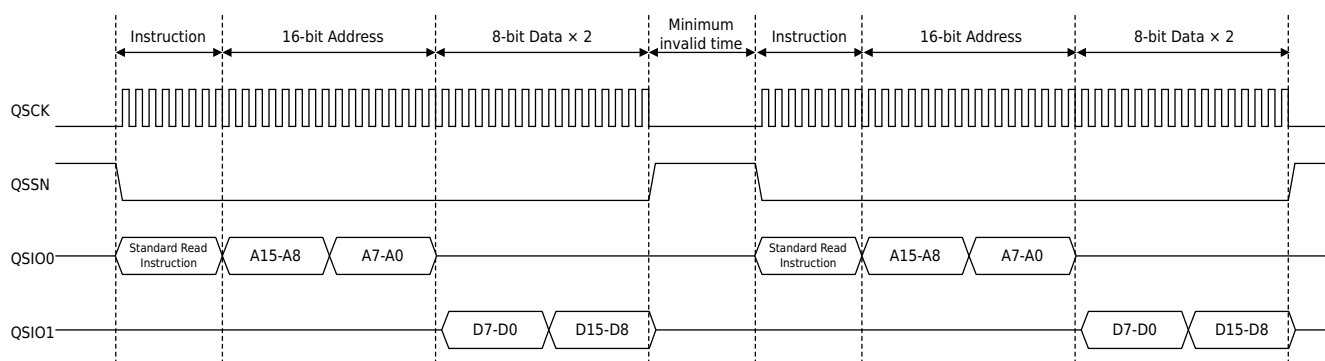


Figure 32-27 Diagram of Single Flash Memory Data Read Operation with Independent Conversion

### 32.4.5.2 Flash Read Using Pre-fetch

For the transfer of CPU instructions or data blocks, the system usually reads data in a sequentially increasing flash memory address sequence. Serial flash memory has continuous data transfer capability without sending instruction codes and addresses again. However, if the internal bus period released by the MCU is converted independently, the QSPI bus period is also divided into independent individuals,

resulting in the inability to actively utilize the advantages of continuous data transmission of the serial flash memory. In this regard, QSPI provides a pre-fetch function for continuous data reception.

The pre-fetch function is activated by setting the QSPI\_CR.PFE bit to 1. When this function is enabled, data will be continuously received and stored in the buffer without waiting for another flash read request. When the MCU issues a flash read operation, the QSPI will match the access address. If the match is successful, the data in the buffer at the corresponding position will be transmitted to the MCU. If the match fails, the data in the buffer will be discarded and a new QSPI bus period will be reissued.

The pre-fetch buffer can store a maximum of 16 bytes of data. In addition, there is a 2-byte data receiving buffer that can also store pre-fetch data. When all the buffer data is full, The QSPI bus period ends. When the buffer data is read and a new buffer space is generated, QSPI will automatically start a new QSPI bus period to resume the pre-fetch action.

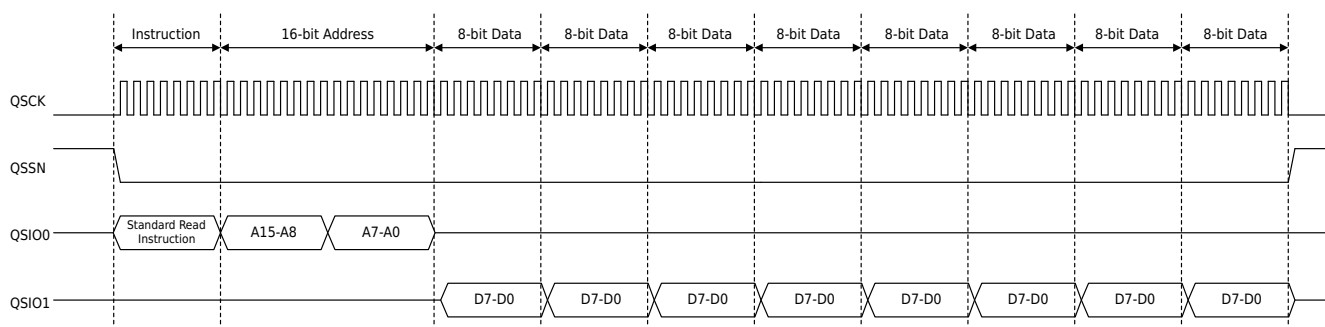


Figure 32-28 Diagram of Data Reading Operation when Pre-fetch Function Enable

### 32.4.5.3 Pre-fetch Termination

If a ROM read bus period to other addresses of the serial flash memory occurs during the pre-fetch transfer, the original pre-fetch action will be terminated and a new QSPI bus period will start. Normally, the pre-fetch operation will be terminated after the current byte transmission is completed, but if the QSPI\_CR.PFSAE bit is set to 1, QSPI will immediately stop the pre-fetch operation without waiting until the current byte transmission is completed. Use of this feature requires the serial flash device to support the instant stop action feature.

### 32.4.5.4 Pre-fetch Status Monitoring

Reading data from a low-speed serial flash memory will increase the load on the system, because the internal bus needs to be in a waiting state until the QSPI bus period of the receiving signal is completed. QSPI provides pre-fetch status monitoring to reduce this load.

The QSPI\_SR.PFAN bit shows the current pre-fetch working status, the QSPI\_SR.PFFUL bit indicates that the pre-fetch data buffer is full, and QSPI\_SR.PFNUM bit shows that the buffer has been read the number of bytes of data in the zone. Through these status bits, it is convenient to determine the current pre-fetch status through a CPU instruction.



#### Note

When executing a pre-fetch status monitoring program, place the program code outside the target serial flash area or enable instruction cache. Otherwise, the pre-fetch object will frequently switch between the object data area and the instruction area, losing the meaning of pre-fetch, and the monitoring program will enter an infinite loop because the pre-fetch cannot be completed.

### 32.4.5.5 Flash Read Using QSPI Bus Cycle Stretching

If the QSPI\_CSCR.SSNW bit is set to a value other than 0b00, the QSPI bus period will be in a hold state after receiving the data and wait for the next data to be read. At this time, the QSSN signal will remain in a low-level active state, while the QSCK is in a stopped state. If the next flash memory read instruction arrives, if the address of the read object is sequentially incremented next to the current address, QSPI will restart QSCK and directly accept data. If it is a discontinuous address, the QSSN signal will be set to a high-level inactive state to end the previously maintained QSPI bus period, and then start a new SPI bus period.

This function can prevent the system from repeatedly sending instruction codes and addresses when performing discontinuous reading of continuously increasing addresses, thereby improving reading efficiency.

The extension time of the QSPI bus period can be set by QSPI\_CSCR.SSNW. When the next reading does not occur after the set time, QSSN will be automatically set to a high-level inactive state to end the QSPI bus period. If QSPI\_CSCR.SSNW is set to 0b11, the QSSN will be extended indefinitely, and the QSPI bus period will always be in the hold state, but this will increase the power consumption of the serial flash memory.

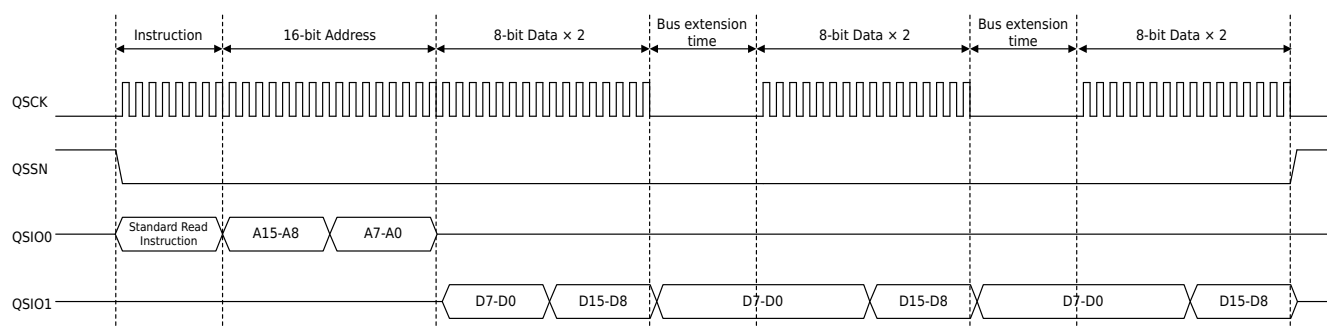


Figure 32-29 Diagram of Data Read Operation by QSPI Bus Cycle Extension Function

## 32.4.6 XIP Control

### 32.4.6.1 XIP Mode Overview

Some serial flash devices can reduce latency by omitting to receive a read instruction. This function is selected by a mode code sent during the dummy period.

During dummy periods at fast speed instructions, QSPI controls the XIP mode of the serial flash by sending XIP mode codes during the first two periods. Different serial flash memories have different XIP mode codes, which can be set specifically through the QSPI\_XCMD.XIPMC bit. Please refer to the figure below for details.



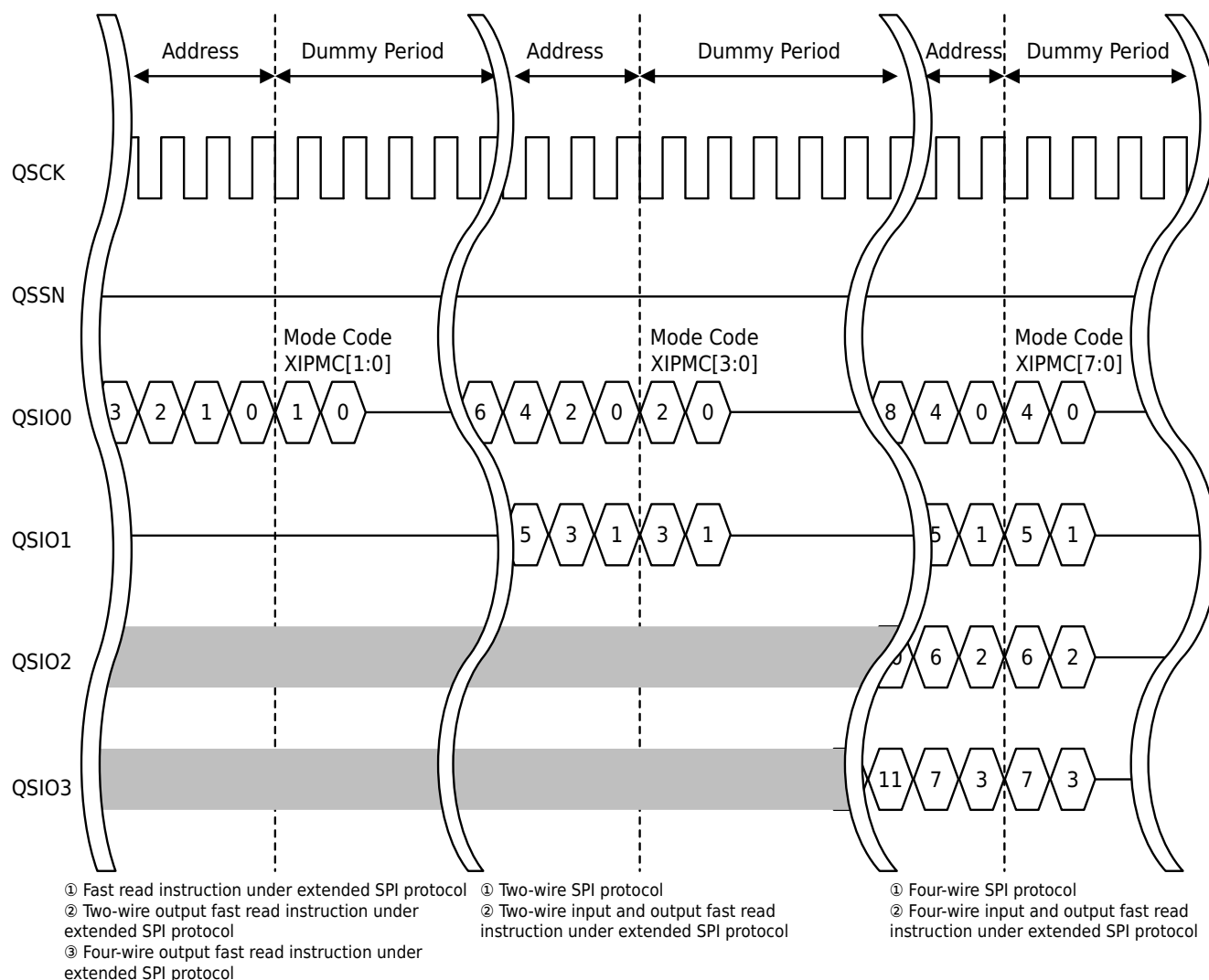


Figure 32-30 XIP Mode Control Diagram

### 32.4.6.2 XIP Mode Setting

When the XIP mode code of the corresponding serial flash memory is written to `QSPI_XCMD.XIPMC` and the `QSPI_CR.XIPE` bit is set to 1, when the next fast read instruction occurs, the set mode code will be transmitted to the target serial flash memory during the first two periods of the virtual period. After the mode code is received, the serial flash memory and its control part will start the XIP mode. Whether the system has entered XIP mode can be confirmed by checking the `QSPI_SR.XIPF` bit.



#### Note

To enable the XIP mode of the serial flash memory, the corresponding mode code must be set in `QSPI_XCMD.XIPMC`. However, for the control part, XIP mode can be enabled simply by setting the `XIPE` bit to 1, regardless of the value in `QSPI_XCMD.XIPMC`.

### 32.4.6.3 Exit From XIP Mode

When the exit XIP mode code of the corresponding serial flash memory is written to `QSPI_XCMD.XIPMC` and the `QSPI_CR.XIPE` bit is set to 0, when the next fast read instruction occurs, the set exit mode code will be transmitted to the target serial flash during the first two periods of the dummy period. After the exit mode code is received, the serial flash memory and its control part will exit the XIP mode. Whether the system has exited XIP mode can be confirmed by checking the `QSPI_SR.XIPF` bit.

**Note**

To exit the XIP mode of the serial flash memory, the corresponding exit mode code must be set in QSPI\_XCMD.XIPMC. However, for the control part, XIP mode can be disabled simply by setting the XIPE bit to 0, regardless of the value in QSPI\_XCMD.XIPMC.

### 32.4.7 Pin Status of QSI02 and QSI03

The state of the QSI02 and QSI03 pins depends on the serial read mode set by the MDSEL[2:0] bits in the QSPI\_CR register.

Table 32-4 Pin Status of QSI02 and QSI03

| QSPI_CR.MDSEL | QSI02 Status                                                                                                | QSI03 Status                                                        | Remark                              |
|---------------|-------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------|-------------------------------------|
| 0b000         | Output status, the output level is determined by the QSPI_FCR.WPOL bit, and the initial output is low level | Output high level                                                   | Standard read (initial status)      |
| 0b001         |                                                                                                             |                                                                     | Fast read                           |
| 0b010         |                                                                                                             |                                                                     | Two-wire output fast read           |
| 0b011         |                                                                                                             |                                                                     | Two-wire input and output fast read |
| 0b100         | As the third data line for input or output, the idle state is Hi-Z                                          | As the fourth data line for input or output, the idle state is Hi-Z | Four-wire output fast read          |
| 0b101         |                                                                                                             |                                                                     | Four-wire input output fast read    |
| 0b110         | Refer to the specific protocol settings for each stage                                                      | Refer to the specific protocol settings for each stage              | Custom protocol standard read       |
| 0b111         |                                                                                                             |                                                                     | Custom protocol fast read           |

**Note**

- The QSI02 pin can also be used as the WP# function of the serial flash.
- The QSI03 pin can also be used as the HOLD# or RESET# function of the serial flash.

### 32.4.8 Direct Communication Mode

#### 32.4.8.1 About Direct Communication Mode

QSPI can read serial flash memory by automatically converting the MCU's external ROM read bus period to QSPI bus period. But the serial flash memory also has many different additional functions, such as ID information reading, erasing, writing, and status information reading. These functions do not have a set of standard instructions to set, and with the rapid increase of new functions of serial flash memory, the correspondence at the hardware level becomes more and more difficult.

In response to this situation, QSPI provides a direct communication mode, and users can directly control the serial flash memory through software. This mode software can generate any desired QSPI bus period.

### 32.4.8.2 Direct Communication Mode Setting

Setting the QSPI\_CR.DCOME bit to 1 can enter direct communication mode. Once it enters the direct communication mode, it will not be able to perform normal flash memory read operations. If you want to perform regular flash memory read operations, you need to clear the DCOME bit to exit the direct communication mode.

**Note**

If the system is in XIP mode, it is necessary to exit the XIP mode first before activating the direct communication mode.

### 32.4.8.3 QSPI Bus Period Generation in Direct Communication Mode

A complete QSPI bus period in direct communication mode starts from the first operation on the QSPI\_DCOM register and ends after a write operation on the QSPI\_CR register. During this period, QSPI\_DCOM.DCOMPRSL and QSPI\_DCOM.DCOM can be operated multiple times.

Direct communication mode does support multi-line actions.

- When DCOMPRSL[1:0] is set to 0b00, the write of DCOM[7:0] is converted into a single-byte data transmission of the QSPI bus extended SPI protocol, and the read of DCOM[7:0] is converted into a single-byte data reception of the QSPI bus extended SPI protocol.
- When DCOMPRSL[1:0] is set to 0b01, the write of DCOM[7:0] is converted into a single-byte data transmission of the QSPI bus two-wire SPI protocol, and the read of DCOM[7:0] is converted into a single-byte data reception of the QSPI bus two-wire SPI protocol.
- When DCOMPRSL[1:0] is set to 0b10, the write of DCOM[7:0] is converted into a single-byte data transmission of the QSPI bus four-wire SPI protocol, and the read of DCOM[7:0] is converted into a single-byte data reception of the QSPI bus four-wire SPI protocol.

From the first operation on the QSPI\_DCOM register to the last write operation on the QSPI\_CR register, the QSSN signal always remains in a active low state.

**Note**

- Writing to any QSPI register other than QSPI\_DCOM in direct communication mode will exit direct communication mode. Exiting direct communication mode by any method other than "QSPI\_CR register write operation" may cause unexpected situations, so it is not recommended.
- In direct communication mode, one-wire, two-wire or four-wire mode can be freely switched by rewriting DCOMPRSL[1:0]. If the line mode needs to be switched before the direct read operation, please note that writing DCOMPRSL[1:0] will result in a dummy data transmission.
- When writing to the QSPI\_DCOM register, you must use 32-bit or 16-bit operations and write to both the DCOMPRSL[1:0] and DCOM[7:0] bits. When reading the QSPI\_DCOM register, 32-bit, 16-bit, or 8-bit operations can be used.

### 32.4.9 Interrupt

When a read access operation to the ROM is detected in direct communication mode, the QSPI\_SR.RAER bit is set to 1, and the QSPI will generate a bus hardware error interrupt. The interrupt request will be held until the RAER bit is cleared. For more information, refer to "Interrupt Controller (INTC)" chapter.

## 32.5 Register Description

### 32.5.1 Register List

QSPI base address: 0x9C000000

Table 32-5 Quad Serial Peripheral Interface (QSPI) Register Offset

| Offset | Register                  | Description                                    | Width |
|--------|---------------------------|------------------------------------------------|-------|
| 0x0000 | <a href="#">QSPI_CR</a>   | QSPI Control Register                          | 32    |
| 0x0004 | <a href="#">QSPI_CSCR</a> | QSPI Chip Select Control Register              | 32    |
| 0x0008 | <a href="#">QSPI_FCR</a>  | QSPI Format Control Register                   | 32    |
| 0x000C | <a href="#">QSPI_SR</a>   | QSPI Status Register                           | 32    |
| 0x0010 | <a href="#">QSPI_DCOM</a> | QSPI Direct Communication Instruction Register | 32    |
| 0x0014 | <a href="#">QSPI_CCMD</a> | QSPI Instruction Code Register                 | 32    |
| 0x0018 | <a href="#">QSPI_XCMD</a> | QSPI XIP Mode Code Register                    | 32    |
| 0x0024 | <a href="#">QSPI_CLR</a>  | QSPI Flag Clear Register                       | 32    |
| 0x0804 | <a href="#">QSPI_EXAR</a> | QSPI External Extended Address Register        | 32    |

## 32.5.2 QSPI Control Register (QSPI\_CR)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |     |    |    |    |          |    |       |    |       |    |       |    |        |   |      |   |       |   |       |   |     |   |       |  |
|--------|--------------|----|----|----|----|----|----|----|----|----|-----|----|----|----|----------|----|-------|----|-------|----|-------|----|--------|---|------|---|-------|---|-------|---|-----|---|-------|--|
| 0x0000 | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21  | 20 | 19 | 18 | 17       | 16 | 15    | 14 | 13    | 12 | 11    | 10 | 9      | 8 | 7    | 6 | 5     | 4 | 3     | 2 | 1   | 0 |       |  |
| Reset  | 0x003F0000   |    |    |    |    |    |    |    |    |    |     |    |    |    |          |    |       |    |       |    |       |    |        |   |      |   |       |   |       |   |     |   |       |  |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    | DIV |    |    |    | Reserved |    | DPRSL |    | APRSL |    | IPRSL |    | SPIMD3 |   | XIPE |   | DCOME |   | PFSAE |   | PFE |   | MDSEL |  |
| Access |              |    |    |    |    |    |    |    |    |    | RW  |    |    |    |          |    | RW    |    | RW    |    | RW    |    | RW     |   | RW   |   | RW    |   | RW    |   | RW  |   | RW    |  |

| Bit   | Name     | Function                                            | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | Access |
|-------|----------|-----------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:22 | Reserved | Reserved                                            | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | -      |
| 21:16 | DIV      | Reference clock selection                           | Serial interface reference clock period configuration:<br><ul style="list-style-type: none"> <li>● 0b000000: 2 HCLK periods</li> <li>● 0b000001: 2 HCLK periods</li> <li>● 0b000010: 3 HCLK periods</li> <li>● .....</li> <li>● 0b111110: 63 HCLK periods</li> <li>● 0b111111: 64 HCLK periods</li> </ul> <div>  <b>Note</b><br/>           The actual value used by the hardware is the programmed value plus 1.         </div> | RW     |
| 15:14 | Reserved | Reserved                                            | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | -      |
| 13:12 | DPRSL    | SPI protocol selection in data receiving phase      | <ul style="list-style-type: none"> <li>● 0b00: Extended SPI protocol</li> <li>● 0b01: Two-wire SPI protocol</li> <li>● 0b10: Four-wire SPI protocol</li> <li>● 0b11: Setting prohibited</li> </ul>                                                                                                                                                                                                                                                                                                                  | RW     |
| 11:10 | APRSL    | SPI protocol selection in address sending phase     | <ul style="list-style-type: none"> <li>● 0b00: Extended SPI protocol</li> <li>● 0b01: Two-wire SPI protocol</li> <li>● 0b10: Four-wire SPI protocol</li> <li>● 0b11: Setting prohibited</li> </ul>                                                                                                                                                                                                                                                                                                                  | RW     |
| 9:8   | IPRSL    | SPI protocol selection in instruction sending phase | <ul style="list-style-type: none"> <li>● 0b00: Extended SPI protocol</li> <li>● 0b01: Two-wire SPI protocol</li> <li>● 0b10: Four-wire SPI protocol</li> <li>● 0b11: Setting prohibited</li> </ul>                                                                                                                                                                                                                                                                                                                  | RW     |
| 7     | SPIMD3   | SPI Mode selection                                  | <ul style="list-style-type: none"> <li>● 0b0: SPI mode 0</li> <li>● 0b1: SPI mode 3</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                      | RW     |
| 6     | XIPE     | XIP mode permission                                 | <ul style="list-style-type: none"> <li>● 0b0: XIP mode disabled</li> <li>● 0b1: XIP mode enabled</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                         | RW     |

| Bit | Name  | Function                            | Description                                                                                                                                                                                                                                                                                                                                                                          | Access |
|-----|-------|-------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 5   | DCOME | Direct communication permission     | QSPI bus communication mode selection:<br><ul style="list-style-type: none"> <li>● 0b0: ROM access mode</li> <li>● 0b1: Direct communication mode</li> </ul>                                                                                                                                                                                                                         | RW     |
| 4   | PFSAE | Pre-fetch immediate stop permission | Select where to reset the pre-fetch operation:<br><ul style="list-style-type: none"> <li>● 0b0: The current pre-fetch operation is aborted on a byte boundary</li> <li>● 0b1: The current pre-fetch operation is immediately terminated</li> </ul>                                                                                                                                   | RW     |
| 3   | PFE   | Pre-fetch permission                | <ul style="list-style-type: none"> <li>● 0b0: Pre-fetch function disabled</li> <li>● 0b1: Pre-fetch function enabled</li> </ul>                                                                                                                                                                                                                                                      | RW     |
| 2:0 | MDSEL | QSPI read mode selection            | <ul style="list-style-type: none"> <li>● 0b000: Standard read</li> <li>● 0b001: Fast read</li> <li>● 0b010: Two-wire output fast read</li> <li>● 0b011: Two-wire input and output fast read</li> <li>● 0b100: Four-wire output fast read</li> <li>● 0b101: Four-wire input and output fast read</li> <li>● 0b110: Custom standard read</li> <li>● 0b111: Custom fast read</li> </ul> | RW     |

## 32.5.3 QSPI Chip Select Control Register (QSPI\_CSCR)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |      |   |   |   |      |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|------|---|---|---|------|---|---|---|
| 0x0004 | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7    | 6 | 5 | 4 | 3    | 2 | 1 | 0 |
| Reset  | 0x0000000F   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |      |   |   |   |      |   |   |   |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   | SSNW |   |   |   | SSHW |   |   |   |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   | RW   |   |   |   | RW   |   |   |   |

| Bit  | Name     | Function                           | Description                                                                                                                                                                                                                                                                                                                                                                                                                             | Access |
|------|----------|------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:6 | Reserved | Reserved                           | -                                                                                                                                                                                                                                                                                                                                                                                                                                       | -      |
| 5:4  | SSNW     | QSSN active time extension setting | <p>QSSN active time extension function selection after QSPI bus access:</p> <ul style="list-style-type: none"> <li>● 0b00: Do not extend</li> <li>● 0b01: Extend 32 QSCK periods</li> <li>● 0b10: Extend 128 QSCK periods</li> <li>● 0b11: Infinite extension</li> </ul> <p> <b>Note</b><br/>This bit setting has no effect in direct mode.</p>       | RW     |
| 3:0  | SSHW     | QSSN minimum inactive time setting | <p>QSSN signal minimum inactive time selection:</p> <ul style="list-style-type: none"> <li>● 0b0000: 1 QSCK period</li> <li>● 0b0001: 2 QSCK periods</li> <li>● .....</li> <li>● 0b1110: 15 QSCK periods</li> <li>● 0b1111: 16 QSCK periods</li> </ul> <p> <b>Note</b><br/>The actual value used by the hardware is the programmed value plus 1.</p> | RW     |

## 32.5.4 QSPI Format Control Register (QSPI\_FCR)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |      |          |    |    |    |        |   |   |   |          |      |       |    |       |          |      |    |      |  |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|------|----------|----|----|----|--------|---|---|---|----------|------|-------|----|-------|----------|------|----|------|--|
| 0x0008 | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15   | 14       | 13 | 12 | 11 | 10     | 9 | 8 | 7 | 6        | 5    | 4     | 3  | 2     | 1        | 0    |    |      |  |
| Reset  | 0x00008033   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |      |          |    |    |    |        |   |   |   |          |      |       |    |       |          |      |    |      |  |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | DUTY | Reserved |    |    |    | DMCYCN |   |   |   | Reserved | WPOL | SSNLD |    | SSNHD | Reserved | 4BIC |    | AWSL |  |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RW   |          |    |    |    | RW     |   |   |   |          |      | RW    | RW | RW    |          | RW   | RW |      |  |

| Bit   | Name     | Function                                  | Description                                                                                                                                                                                                                                                                                                                                                                                                                                               | Access |
|-------|----------|-------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:16 | Reserved | Reserved                                  | -                                                                                                                                                                                                                                                                                                                                                                                                                                                         | -      |
| 15    | DUTY     | QSK output waveform duty cycle correction | <ul style="list-style-type: none"> <li>0b0: Do not perform duty correction</li> <li>0b1: Delay the rising edge of QSK by 0.5 HCLK period (active when the frequency selected by QSK is an odd multiple of HCLK)</li> </ul>                                                                                                                                                                                                                                | RW     |
| 14:12 | Reserved | Reserved                                  | -                                                                                                                                                                                                                                                                                                                                                                                                                                                         | -      |
| 11:8  | DMCYCN   | Dummy period setting                      | Dummy Period Selection When Using Fast Read Instructions: <ul style="list-style-type: none"> <li>0b0000: 3 QSK periods</li> <li>0b0001: 4 QSK periods</li> <li>.....</li> <li>0b1110: 17 QSK periods</li> <li>0b1111: 18 QSK periods</li> </ul> <div>  <b>Note</b><br/>           The actual value used by the hardware is the programmed value plus 3.         </div> | RW     |
| 7     | Reserved | Reserved                                  | -                                                                                                                                                                                                                                                                                                                                                                                                                                                         | -      |
| 6     | WPOL     | WP pin (QSIO2) output level setting       | <ul style="list-style-type: none"> <li>0b0: Low level</li> <li>0b1: High level</li> </ul>                                                                                                                                                                                                                                                                                                                                                                 | RW     |
| 5     | SSNLD    | QSSN signal output time delay setting     | <ul style="list-style-type: none"> <li>0b0: Output QSSN 0.5 QSK ahead of the first rising edge of QSK</li> <li>0b1: Output QSSN 1.5 QSK ahead of the first rising edge of QSK</li> </ul>                                                                                                                                                                                                                                                                  | RW     |
| 4     | SSNHD    | QSSN signal release time delay setting    | <ul style="list-style-type: none"> <li>0b0: Release QSSN 0.5 QSK after the last rising edge of QSK</li> <li>0b1: Release QSSN 1.5 QSK after the last rising edge of QSK</li> </ul>                                                                                                                                                                                                                                                                        | RW     |
| 3     | Reserved | Reserved                                  | -                                                                                                                                                                                                                                                                                                                                                                                                                                                         | -      |



| Bit | Name | Function                                       | Description                                                                                                                                                                                                                         | Access |
|-----|------|------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 2   | 4BIC | 4-byte address read instruction code selection | Read instruction code selection when the address width is 4 bytes <ul style="list-style-type: none"><li>● 0b0: Not use 4-byte address to read instruction code</li><li>● 0b1: Use 4-byte address to read instruction code</li></ul> | RW     |
| 1:0 | AWSL | Serial interface address width selection       | <ul style="list-style-type: none"><li>● 0b00: 1 byte</li><li>● 0b01: 2 bytes</li><li>● 0b10: 3 bytes</li><li>● 0b11: 4 bytes</li></ul>                                                                                              | RW     |

## 32.5.5 QSPI Status Register (QSPI\_SR)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |      |       |          |    |       |    |   |      |      |   |          |   |   |      |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|------|-------|----------|----|-------|----|---|------|------|---|----------|---|---|------|---|---|
| 0x000C | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15   | 14    | 13       | 12 | 11    | 10 | 9 | 8    | 7    | 6 | 5        | 4 | 3 | 2    | 1 | 0 |
| Reset  | 0x00008000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |      |       |          |    |       |    |   |      |      |   |          |   |   |      |   |   |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |      |       |          |    | PFNUM |    |   |      |      |   | Reserved |   |   |      |   |   |
|        |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | PFAN | PFFUL | Reserved |    |       |    |   | RAER | XIPF |   |          |   |   | BUSY |   |   |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RO   | RO    |          | RO |       |    |   | RO   | RO   |   |          |   |   | RO   |   |   |

| Bit   | Name     | Function                                               | Description                                                                                                                                                                                                                                                             | Access |
|-------|----------|--------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:16 | Reserved | Reserved                                               | -                                                                                                                                                                                                                                                                       | -      |
| 15    | PFAN     | Pre-fetch action status                                | <ul style="list-style-type: none"> <li>0b0: Pre-fetch enabled</li> <li>0b1: Pre-fetch disabled</li> </ul>                                                                                                                                                               | RO     |
| 14    | PFFUL    | Pre-fetch buffer status                                | <ul style="list-style-type: none"> <li>0b0: Pre-fetch buffer is not full</li> <li>0b1: Pre-fetch buffer data is full</li> </ul>                                                                                                                                         | RO     |
| 13    | Reserved | Reserved                                               | -                                                                                                                                                                                                                                                                       | -      |
| 12:8  | PFNUM    | Number of bytes of data stored in the pre-fetch buffer | <ul style="list-style-type: none"> <li>0b00000: 0 byte</li> <li>0b00001: 1 byte</li> <li>.....</li> <li>0b10001: 17 bytes</li> <li>0b10010: 18 bytes</li> <li>Other: Reserved</li> </ul>                                                                                | RO     |
| 7     | RAER     | ROM access error flag                                  | Error flag bit for ROM access in direct communication mode: <ul style="list-style-type: none"> <li>0b0: No ROM access detected</li> <li>0b1: ROM access detected</li> </ul> Clear condition: <ul style="list-style-type: none"> <li>QSPI_CLR's RAERCLR set 1</li> </ul> | RO     |
| 6     | XIPF     | XIP mode flag                                          | <ul style="list-style-type: none"> <li>0b0: Non-XIP mode</li> <li>0b1: XIP Mode</li> </ul>                                                                                                                                                                              | RO     |
| 5:1   | Reserved | Reserved                                               | -                                                                                                                                                                                                                                                                       | -      |
| 0     | BUSY     | Bus busy flag                                          | QSPI bus working status flag bit in direct communication mode: <ul style="list-style-type: none"> <li>0b0: Bus idle, no serial transmission process</li> <li>0b1: Bus busy, serial transmission process is in progress</li> </ul>                                       | RO     |

## 32.5.6 QSPI Direct Communication Instruction Register (QSPI\_DCOM)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |         |   |      |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---------|---|------|---|---|---|---|---|---|---|
| 0x0010 | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9       | 8 | 7    | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0XXXXXXXXX   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |         |   |      |   |   |   |   |   |   |   |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | DCOMPSL |   | DCOM |   |   |   |   |   |   |   |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RW      |   |      |   |   |   |   |   |   |   |

| Bit   | Name     | Function                                            | Description                                                                                                                                                                                                                                                                                               | Access |
|-------|----------|-----------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:10 | Reserved | Reserved                                            | -                                                                                                                                                                                                                                                                                                         | -      |
| 9:8   | DCOMPRSL | SPI protocol selection in direct communication mode | <ul style="list-style-type: none"> <li>0b00: Extended SPI protocol</li> <li>0b01: Two-wire SPI protocol</li> <li>0b10: Four-wire SPI protocol</li> <li>0b11: Setting prohibited</li> </ul>                                                                                                                | RW     |
| 7:0   | DCOM     | Direct communication mode instruction               | The interface in the direct communication mode performs direct communication through the QSPI bus. Read and write accesses to this interface are translated into a corresponding QSPI bus period. This interface is only valid in direct communication mode, and access in ROM access mode is prohibited. | RW     |

## 32.5.7 QSPI Instruction Code Register (QSPI\_CCMD)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |     |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|-----|---|---|---|---|---|---|---|
| 0x0014 | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7   | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |     |   |   |   |   |   |   |   |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   | RIC |   |   |   |   |   |   |   |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |     |   |   |   |   |   |   |   |

| Bit  | Name     | Function                 | Description                                                         | Access |
|------|----------|--------------------------|---------------------------------------------------------------------|--------|
| 31:8 | Reserved | Reserved                 | -                                                                   | -      |
| 7:0  | RIC      | Replace instruction code | Serial flash instruction codes used to replace default instructions | RW     |

## 32.5.8 QSPI XIP Mode Code Register (QSPI\_XCMD)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |       |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|-------|---|---|---|---|---|---|---|
| 0x0018 | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7     | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x000000FF   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |       |   |   |   |   |   |   |   |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   | XIPMC |   |   |   |   |   |   |   |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   | RW    |   |   |   |   |   |   |   |

| Bit  | Name     | Function      | Description                                       | Access |
|------|----------|---------------|---------------------------------------------------|--------|
| 31:8 | Reserved | Reserved      | -                                                 | -      |
| 7:0  | XIPMC    | XIP mode code | Mode code for serial flash memory. (Set XIP mode) | RW     |

## 32.5.9 QSPI Flag Clear Register (QSPI\_CLR)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |         |   |          |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---------|---|----------|---|---|---|---|---|
| 0x0024 | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7       | 6 | 5        | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0xFFFFFFFF   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |         |   |          |   |   |   |   |   |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   | RAERCLR |   | Reserved |   |   |   |   |   |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   | WO      |   |          |   |   |   |   |   |

| Bit  | Name     | Function        | Description                                                  | Access |
|------|----------|-----------------|--------------------------------------------------------------|--------|
| 31:8 | Reserved | Reserved        | -                                                            | -      |
| 7    | RAERCLR  | RAER flag clear | Write 1 to clear the QSPI_SR.RAER bit, write 0 has no effect | WO     |
| 6:0  | Reserved | Reserved        | -                                                            | -      |

## 32.5.10 QSPI External Extended Address Register (QSPI\_EXAR)

| Offset | Bit Position |    |    |    |    |    |          |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0x0804 | 31           | 30 | 29 | 28 | 27 | 26 | 25       | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |          |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | EXADR        |    |    |    |    |    | Reserved |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|        |              |    |    |    |    |    |          |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access | RW           |    |    |    |    |    |          |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit   | Name     | Function                       | Description                                                                                                                                                       | Access |
|-------|----------|--------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:26 | EXADR    | External Extended Address Code | The upper 6 bits of the QSPI external address are set, and the ROM access window address of the QSPI can access a maximum of 64MB×64 blocks of external ROM space | RW     |
| 25:0  | Reserved | Reserved                       | -                                                                                                                                                                 | -      |

## 32.6 Precautions

### 32.6.1 QSPI Registers Setting

If the QSPI control register is dynamically set or changed during QSPI communication, it may cause new communication to start before the register is fully set. Therefore, it is not recommended to dynamically modify the QSPI control register during communication.

### 32.6.2 Module Stop Bit Settings

QSPI is in the module stop state after system reset. When the QSPI module is required to work, please clear the QSPI module control bit in the PWC\_FCG register first. For specific descriptions, please refer to the relevant sections of the "Action Mode and Low Power Mode" chapter in the "Power Control (PWC) " chapter.

## 33 Inter\_IC Sound Bus (I2S)

### 33.1 Introduction

I2S (Inter\_IC Sound Bus), integrated circuit built-in sound bus, which is dedicated to data transmission between sound devices.

### 33.2 Features

Table 33-1 I2S Main Features

| Function             | Main Features                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|----------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Communication mode   | <ul style="list-style-type: none"> <li>● Support full-duplex and half-duplex communication</li> <li>● Support master mode or slave mode operation</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| Data format          | <ul style="list-style-type: none"> <li>● Optional channel length: 16/32 bits</li> <li>● Optional transmission data length: 16/24/32 bits</li> <li>● Data shift order: MSB start</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                      |
| Baud rate            | <ul style="list-style-type: none"> <li>● 8-bit programmable linear prescaler for precise audio sampling frequency</li> <li>● Support sampling frequencies 192kHz, 96kHz, 48kHz, 44.1kHz, 32kHz, 22.05kHz, 16kHz, 8kHz</li> <li>● Can output driving clock to drive external audio components, the ratio is fixed at <math>256 \times F_s</math> (<math>F_s</math> is the audio sampling frequency)</li> </ul>                                                                                                                                                                                   |
| Support I2S protocol | <ul style="list-style-type: none"> <li>● I2S Philips Standard</li> <li>● MSB Alignment Standard</li> <li>● LSB Alignment Standard</li> <li>● PCM Standard</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                            |
| Data buffer          | <ul style="list-style-type: none"> <li>● With 4-word deep, 32-bit wide input and output FIFO buffer area</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| Clock source         | <ul style="list-style-type: none"> <li>● Can use internal I2SCLK (PLLAR/PLLAQ/PLLAP/PLLHR/PLLHQ/PLLHP)</li> <li>● Can be provided by an external clock on the I2S_EXCK pin</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                           |
| Interrupt            | <ul style="list-style-type: none"> <li>● An interrupt is generated when the active space of the send buffer reaches the alarm threshold</li> <li>● An interrupt is generated when the active space of the receive buffer reaches the alarm threshold</li> <li>● The receiving data area is full and there is still a request to write data, the receiving overflow</li> <li>● The transmit data area is empty and there are still sending requests, the sending underflow</li> <li>● The transmit data area is full and there is still a request to write data, the sending overflow</li> </ul> |

33.3 Functional Description

33.3.1 System Block Diagram

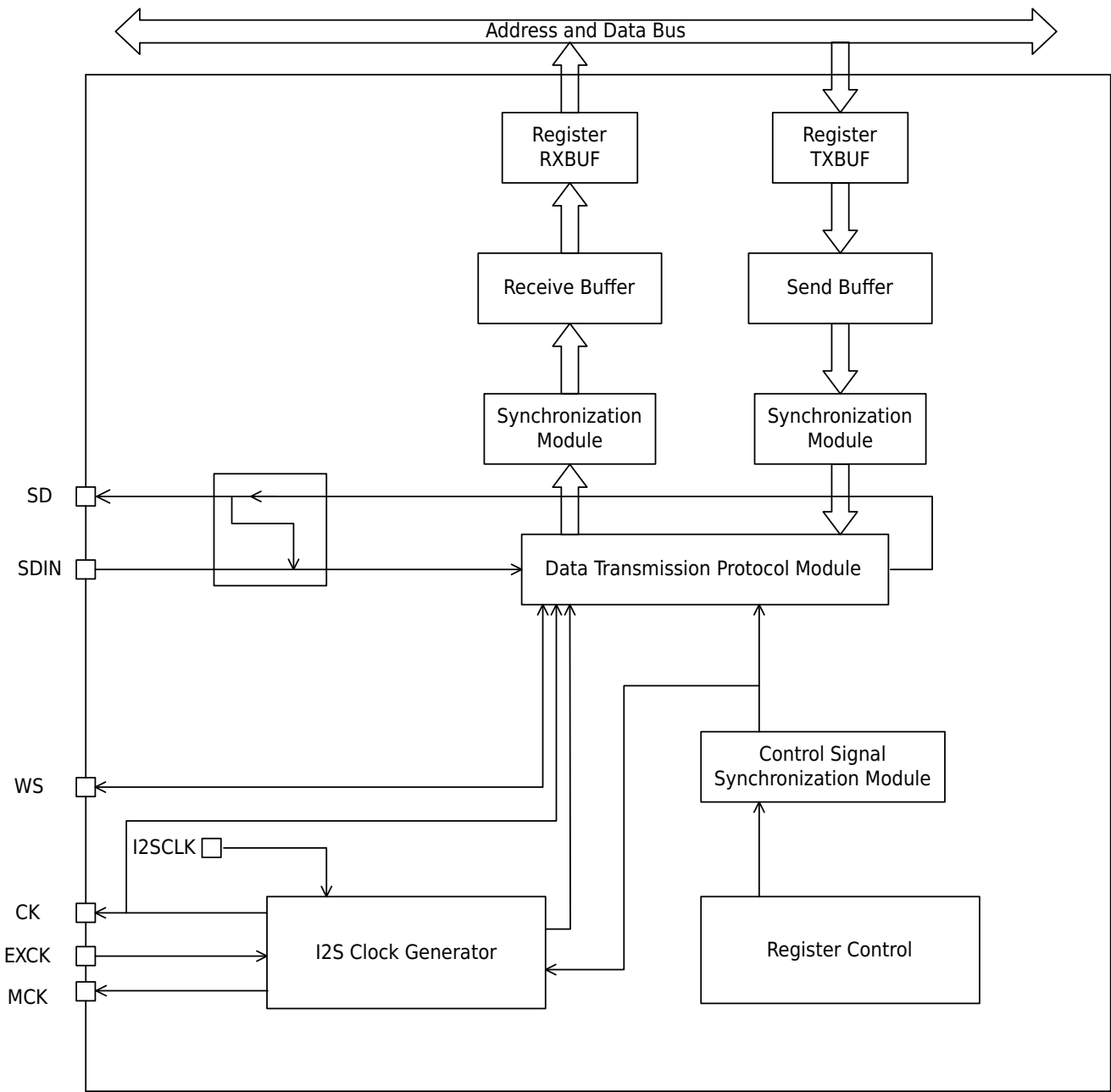


Figure 33-1 I2S System Block Diagram

33.3.2 Pin Description

Table 33-2 I2S Pin Description

| Pin Name | Direction    | Function Description                                                                                                            |
|----------|--------------|---------------------------------------------------------------------------------------------------------------------------------|
| I2Sn_CK  | Input/Output | Serial clock, which is the serial clock output in master mode and the serial clock input in slave mode.                         |
| I2Sn_WS  | Input/Output | Word selection, which is the data control signal output in the master mode and the data control signal input in the slave mode. |



| Pin Name  | Direction    | Function Description                                                                                                                                                                                                                                                                                                                                                                                                        |
|-----------|--------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| I2Sn_SD   | Input/Output | Serial data, for half-duplex mode data input, or half/full-duplex mode data output.                                                                                                                                                                                                                                                                                                                                         |
| I2Sn_SDIN | Input        | Full-duplex audio data input. Pin used for serial data input in I2S full-duplex mode.                                                                                                                                                                                                                                                                                                                                       |
| I2Sn_EXCK | Input        | The external clock source is that the clock generator in the master mode selects an external clock as the divided clock source.                                                                                                                                                                                                                                                                                             |
| I2Sn_MCK  | Output       | Drive clock, when I2S is configured as master mode (and MCKOE bit is set to 1), use the drive clock (separate mapping) to output this additional clock. The clock output frequency is $256 \times F_s$ , where $F_s$ is the audio signal sampling frequency. I2S uses its own clock generator to generate the communication clock in the master mode. This clock generator is also the source that drives the clock output. |

**Note**

The value of n is 1-4.

### 33.3.3 Communication Method

It supports half-duplex and full-duplex communication modes, which can be selected through the I2S\_CTRL.DUPLEX bit.

- When the I2S\_CTRL.DUPLEX bit is 0, I2S operates in the half-duplex communication mode, uses the I2Sn\_SD pin as the output data pin when only sending, and uses the I2Sn\_SD pin as the input data pin when only receiving.
- When the I2S\_CTRL.DUPLEX bit is 1, I2S operates in full-duplex communication mode. At this time, the I2Sn\_SDIN pin is used as the input data pin, and the I2Sn\_SD pin is used as the output data pin to realize full-duplex communication.

### 33.3.4 Supported Audio Protocols

#### 33.3.4.1 Audio Protocol Overview

There are four data and frame format combinations to send data in the following formats:

- 16-bit data encapsulated in 16-bit frame
- 16-bit data encapsulated in 32-bit frame
- 24-bit data encapsulated in 32-bit frame
- 32-bit data encapsulated in 32-bit frame

When using 16 bits of data in a 32-bit frame, the first 16 bits (MSB) are valid and the 16 LSB are forced to 0 without any software action (just one read/write operation).

When using 24 bits of data in a 32-bit frame, the first 24 bits (MSB) are valid and the 8 LSB are forced to 0 without any software action (just one read/write operation).

For all data formats and communication standards, the most significant bit is always sent first (MSB first).

The I2S interface supports 4 audio protocols, which can be configured using the I2S\_CFGR.I2SSTD and I2S\_CFGR.PCMSYNC bit.

### 33.3.4.2 I2S Philips Standard

Use the WS signal to indicate which channel the data currently being sent belongs to. This signal is enable one clock before the first bit (MSB) of the current channel's data.

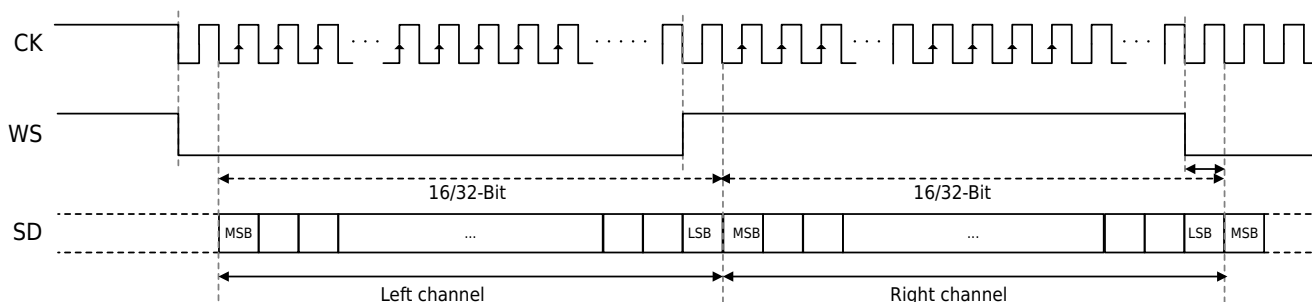


Figure 33-2 I2S Philips Protocol Waveform (16/32 bit Full Precision)

16 bits are encapsulated in a 16-bit frame. In transmit mode, write to I2S\_TXBUF register 0xFFFF3344, SD outputs serial data 0x3344; in receive mode, SD inputs serial data 0xEEDD, and I2S\_RXBUF register reads data 0x0000EEDD.

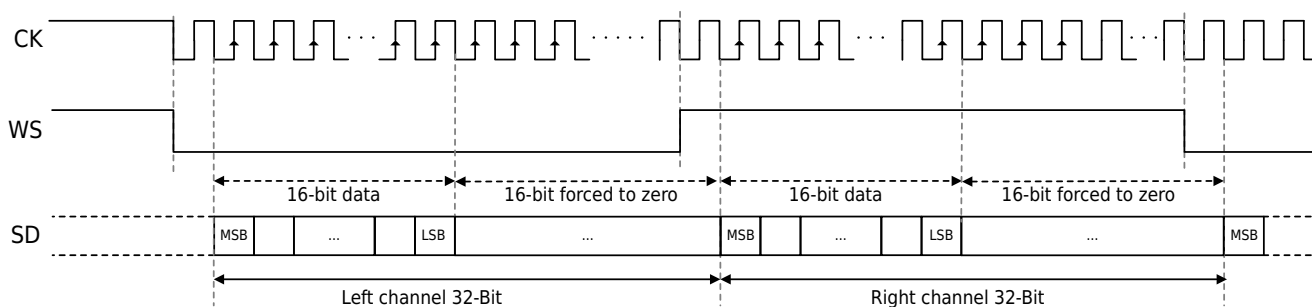


Figure 33-3 I2S Philips Protocol Waveform (16-bit Data Encapsulated in 32-bit Frame)

16 bits are encapsulated in a 32-bit frame. In transmit mode, write to I2S\_TXBUF register 0xFFFF3344, SD outputs serial data 0x33440000; in receive mode, SD inputs serial data 0xEEDDXXXX, and I2S\_RXBUF register reads data 0x0000EEDD.

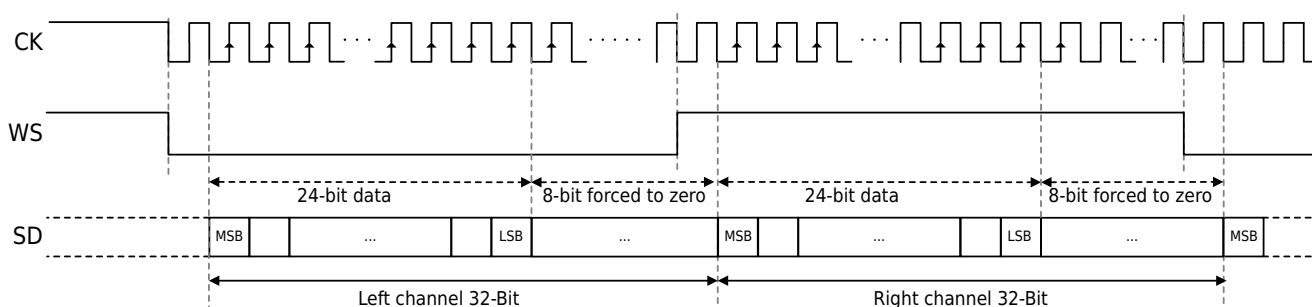


Figure 33-4 I2S Philips Protocol Waveform (24-bit Data Encapsulated in 32-bit Frame)

24 bits are encapsulated in a 32-bit frame. In transmit mode, write to I2S\_TXBUF register 0xFF223344, SD outputs serial data 0x22334400; in receive mode, SD inputs serial data 0xEEDD11XX, and I2S\_RXBUF register reads data 0x00EEDD11.

### 33.3.4.3 MSB Alignment Standard

This standard generates both the WS signal and the first data bit (ie MSBit).

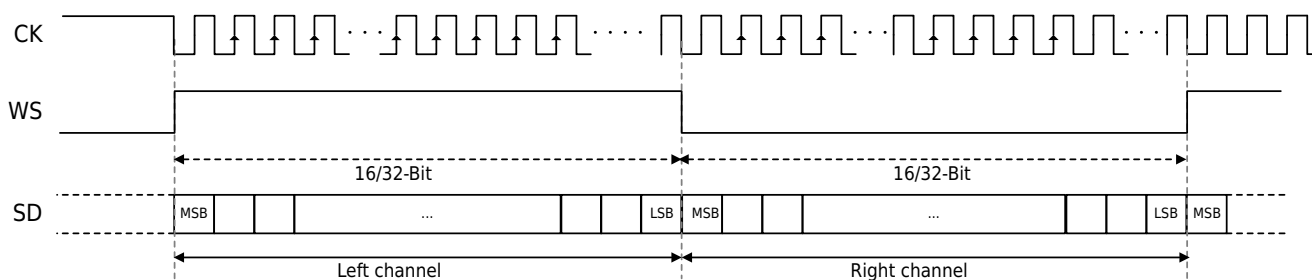


Figure 33-5 I2S MSB Protocol Waveform (16/32-bit Full Precision)

The sender changes data on the falling edge of the clock signal; the receiver reads data on the rising edge.

16 bits are encapsulated in a 16-bit frame. In transmit mode, write to I2S\_TXBUF register 0xFFFF3344, SD outputs serial data 0x3344; in receive mode, SD inputs serial data 0xEEDD, and I2S\_RXBUF register reads data 0x0000EEDD.

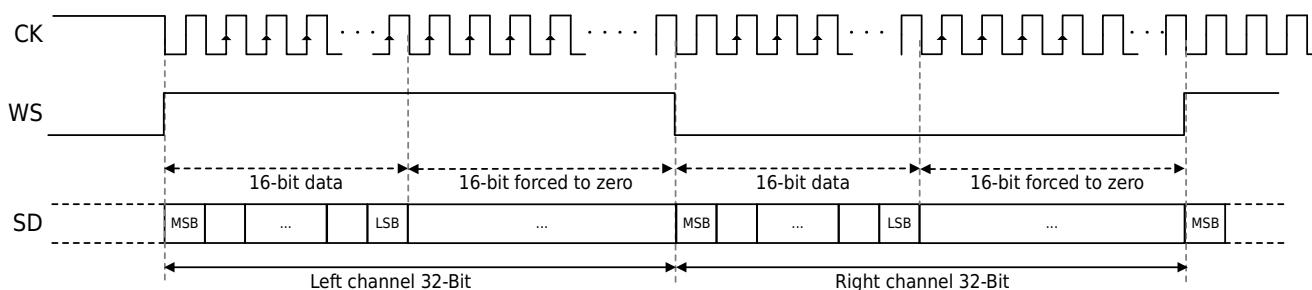


Figure 33-6 I2S MSB Protocol Waveform (16-bit Data Encapsulated in 32-bit Frame)

16 bits are encapsulated in a 32-bit frame. In transmit mode, write to I2S\_TXBUF register 0xFFFF3344, SD outputs serial data 0x33440000; in receive mode, SD inputs serial data 0xEEDDXXXX, and I2S\_RXBUF register reads data 0x0000EEDD.

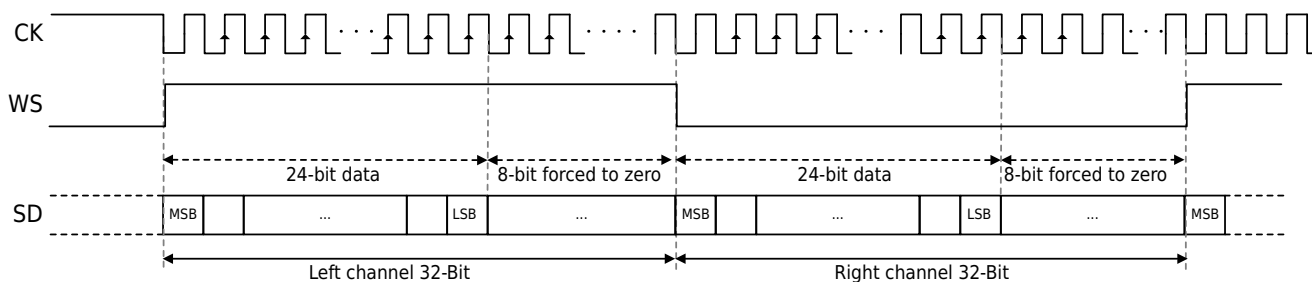


Figure 33-7 I2S MSB Protocol Waveform (24-bit Data Encapsulated in 32-bit Frame)

24 bits are encapsulated in a 32-bit frame. In transmit mode, write to I2S\_TXBUF register 0xFF223344, SD outputs serial data 0x22334400; in receive mode, SD inputs serial data 0xEEDD11XX, and I2S\_RXBUF register reads data 0x00EEDD11.

#### 33.3.4.4 LSB Alignment Standard

This standard is similar to the MSB alignment standard (for 16-bit and 32-bit full-precision frame formats, there is no difference).

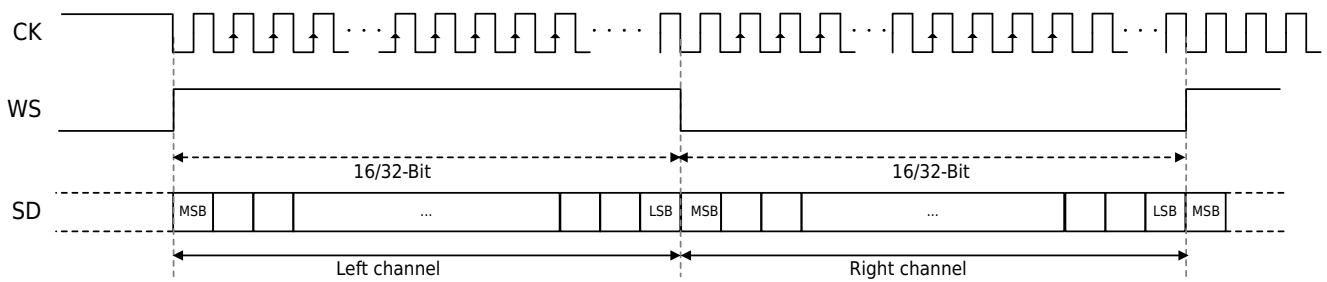


**Figure 33-8 I2S LSB Protocol Waveform (16/32-bit Full Precision)**

16 bits are encapsulated in a 16-bit frame. In transmit mode, write to I2S\_TXBUF register 0xFFFF3344, SD outputs serial data 0x3344; in receive mode, SD inputs serial data 0xEEDD, and I2S\_RXBUF register reads data 0x0000EEDD.

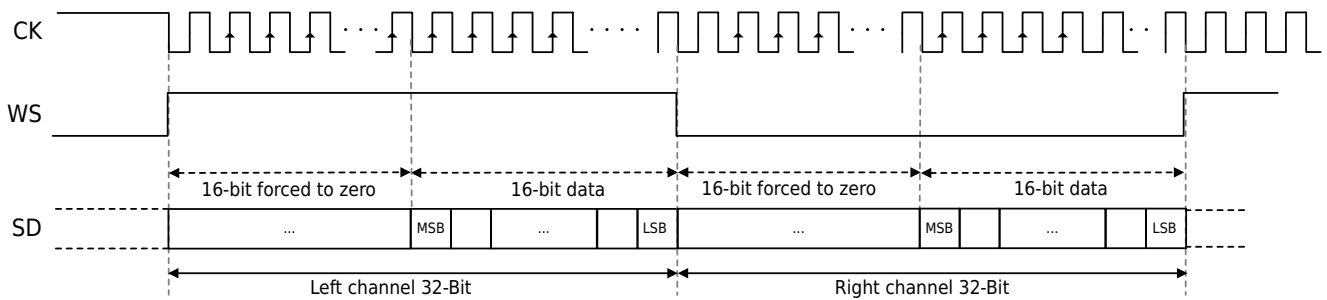


**Figure 33-9 I2S LSB Protocol Waveform (16-bit Data Encapsulated in 32-bit Frame)**

16 bits are encapsulated in a 32-bit frame. In transmit mode, write to I2S\_TXBUF register 0xFFFF3344, SD outputs serial data 0x00003344; in receive mode, SD inputs serial data 0xFFFF1122, and I2S\_RXBUF register reads data 0x00001122.

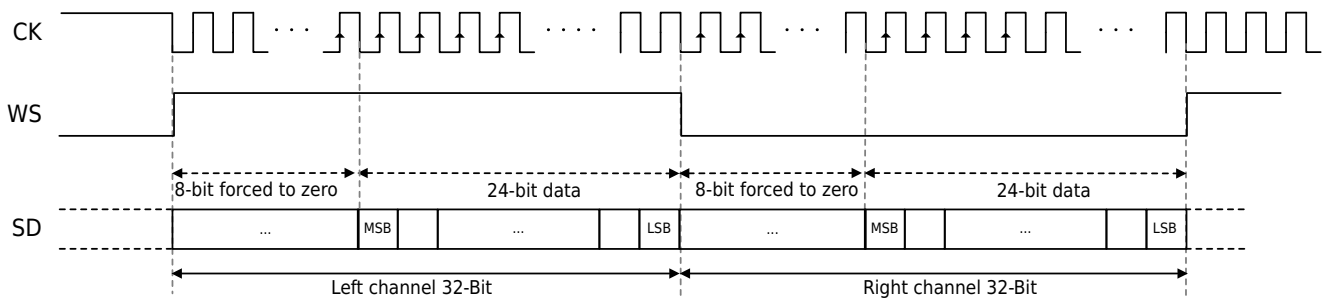


**Figure 33-10 I2S LSB Protocol Waveform (24-bit Data Encapsulated in 32-bit Frame)**

24 bits are encapsulated in a 32-bit frame. In transmit mode, write to I2S\_TXBUF register 0xFF223344, SD outputs serial data 0x00223344; in receive mode, SD inputs serial data 0xFFDD1122, and I2S\_RXBUF register reads data 0x00DD1122.

### 33.3.4.5 PCM Standard

For the PCM standard, there is no need to use channel information. There are two PCM modes (Short Frame and Long Frame) and are configurable using the in I2S\_CFGR.PCMSYNC bit.

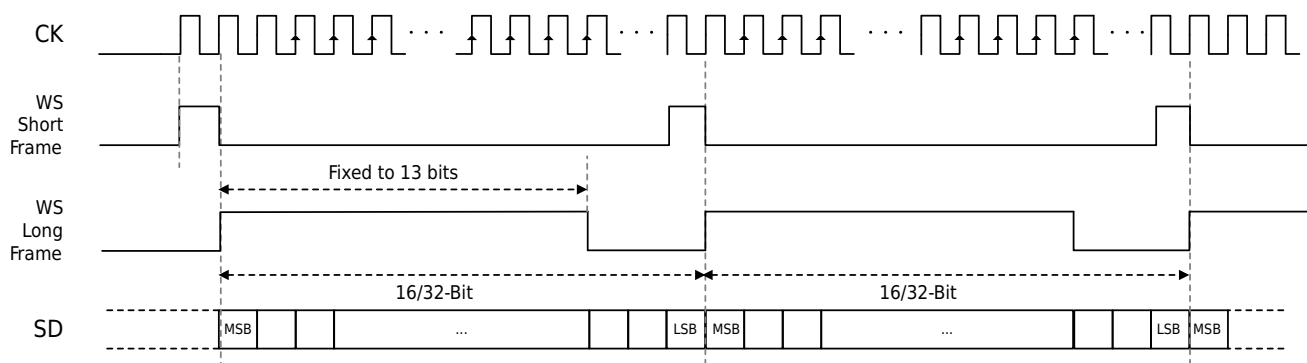


Figure 33-11 I2S PCM Protocol Waveform (16/32-bit Full Precision)

For long frame synchronization, the WS signal is maintained for 13 periods in master mode.

For short frame synchronization, the duration of the WS synchronization signal is only one period.

16 bits are encapsulated in a 16-bit frame. In transmit mode, write to I2S\_TXBUF register 0xFFFF3344, SD outputs serial data 0x3344; in receive mode, SD inputs serial data 0xEEDD, and I2S\_RXBUF register reads data 0x0000EEDD.

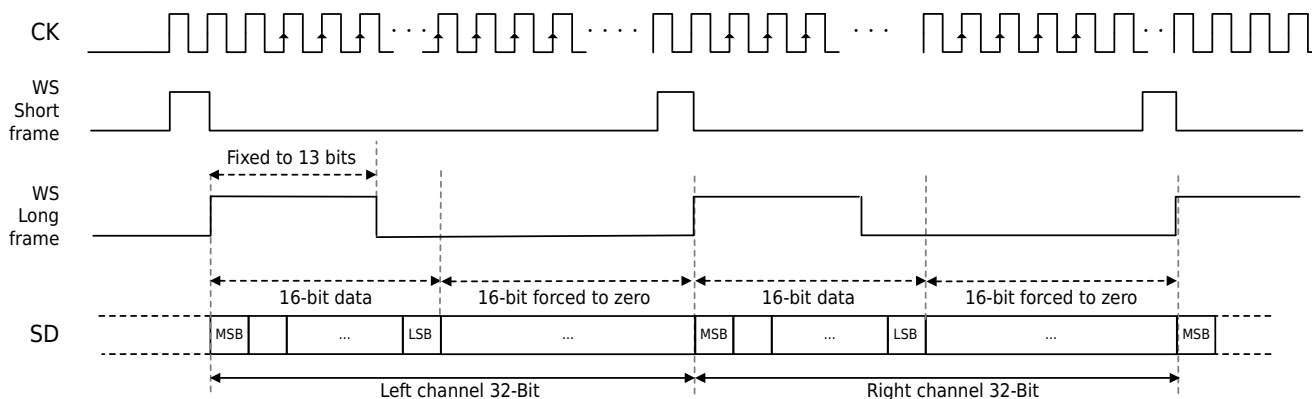


Figure 33-12 I2S PCM Protocol Waveform (16-bit Data Encapsulated in 32-bit Frame)

16 bits are encapsulated in a 32-bit frame. In transmit mode, write to I2S\_TXBUF register 0xFFFF3344, SD outputs serial data 0x00003344; in receive mode, SD inputs serial data 0xEEDDXXXX, and I2S\_RXBUF register reads data 0x0000EEDD.

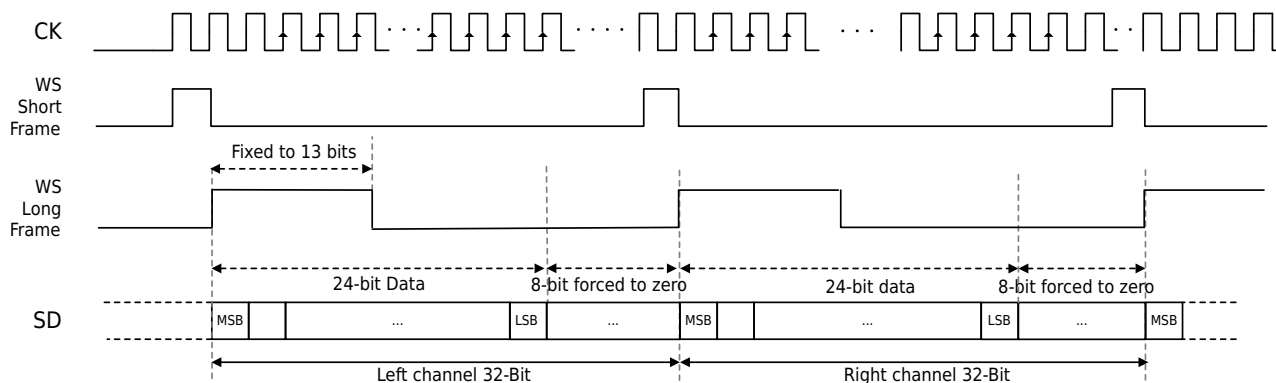


Figure 33-13 I2S PCM Protocol Waveform (24-bit Data Encapsulated in 32-bit Frame)

24 bits are encapsulated in a 32-bit frame. In transmit mode, write to I2S\_TXBUF register 0xXX223344, SD outputs serial data 0x00223344; in receive mode, SD inputs serial data 0xEEDD11XX, and I2S\_RXBUF register reads data 0x00EEDD11.



#### Note

For two modes (master/slave mode) and two synchronizations (short/long synchronization), even in slave mode, it need to specify the number of bits (I2S\_CFGR.DATLEN bit and I2S\_CFGR.CHLEN bit in I2S\_CFGR register) between two sets of continuous data (and two synchronization signals).

### 33.3.5 Clock Generator

The I2S bit rate is used to determine the data flow on the I2S data line and the I2S clock signal frequency.

I2S bit rate=number of bits per channel×number of channels×audio sampling frequency

For 16-bit dual-channel audio, the calculation formula of I2S bit rate is as follows: I2S bit rate=16×2×Fs.  
If the frame is 32 bits wide, then the I2S bit rate=32×2×Fs.

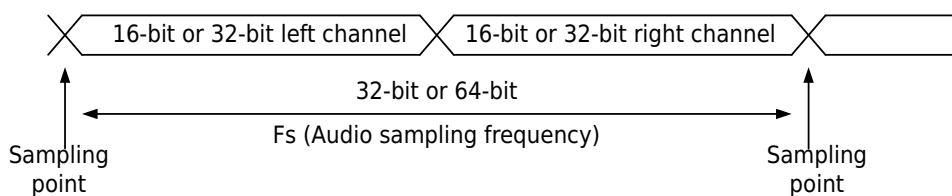


Figure 33-14 Audio Sampling Frequency Definition

When configuring master mode, the linear divider needs to be set up correctly to communicate at the desired audio frequency.

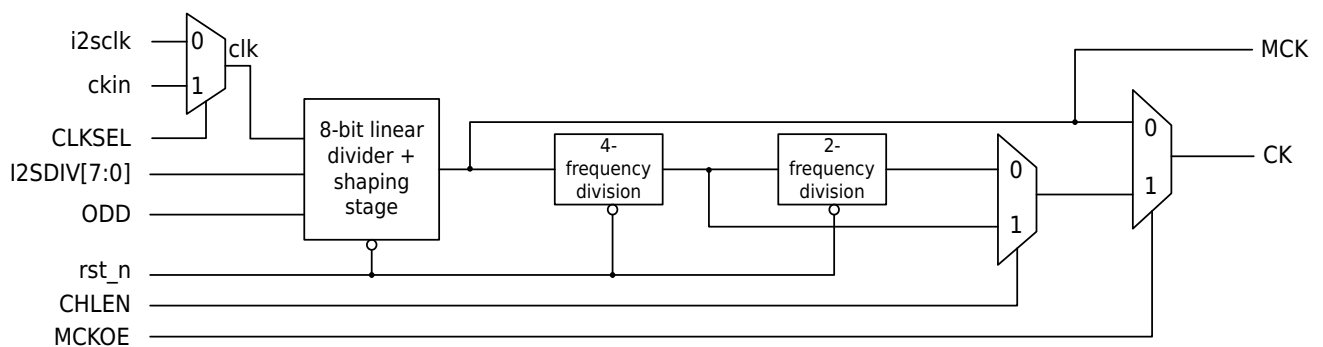


Figure 33-15 Clock Generator Architecture

The clk clock source can be I2SCLK output or external clock.

The audio sampling frequency may be 192kHz, 96kHz or 48kHz etc. To achieve the desired frequency, the current divider needs to be programmed according to the following formula:

- **When output drive clock (mckoe is set to 1) :**

$F_s = \text{clk} / [(16 \times 2) * ((2 * I2SDIV + ODD) * 8)]$  (when the channel frame width is 16 bits)

$F_s = \text{clk} / [(32 \times 2) * ((2 * I2SDIV + ODD) * 4)]$  (when the channel frame width is 32 bits)

- **When turn off drive clock (mckoe is cleared) :**

$F_s = \text{clk} / [(16 \times 2) * (2 * I2SDIV + ODD)]$  (when the channel frame width is 16 bits)

$F_s = \text{clk} / [(32 \times 2) * (2 * I2SDIV + ODD)]$  (when the channel frame width is 32 bits)

The table below provides example accuracy values for different clock configurations. There are other configurations that achieve better clock accuracy.

**Table 33-3 Audio Frequency Accuracy (For VCO Input Frequency=1MHz)**

| Drive Clock     | Target Fs (Hz) | Data Format | Frequency Multiplication Coefficient | Output Division Ratio | I2SDIV | ODD | Real-time fS(Hz) | Error   |
|-----------------|----------------|-------------|--------------------------------------|-----------------------|--------|-----|------------------|---------|
| Output Disabled | 8000           | 16-bit      | 288                                  | 3                     | 187    | 1   | 8000             | 0.0000% |
|                 |                | 32-bit      | 256                                  | 4                     | 62     | 1   | 8000             | 0.0000% |
|                 | 16000          | 16-bit      | 256                                  | 4                     | 62     | 1   | 16000            | 0.0000% |
|                 |                | 32-bit      | 256                                  | 2                     | 62     | 1   | 16000            | 0.0000% |
|                 | 32000          | 16-bit      | 256                                  | 2                     | 62     | 1   | 32000            | 0.0000% |
|                 |                | 32-bit      | 256                                  | 5                     | 12     | 1   | 32000            | 0.0000% |
|                 | 48000          | 16-bit      | 384                                  | 10                    | 12     | 1   | 48000            | 0.0000% |
|                 |                | 32-bit      | 384                                  | 5                     | 12     | 1   | 48000            | 0.0000% |
|                 | 96000          | 16-bit      | 384                                  | 5                     | 12     | 1   | 96000            | 0.0000% |
|                 |                | 32-bit      | 424                                  | 3                     | 11     | 1   | 96014.49219      | 0.0151% |
|                 | 22050          | 16-bit      | 290                                  | 3                     | 68     | 1   | 22049.87695      | 0.0006% |
|                 |                | 32-bit      | 302                                  | 2                     | 53     | 1   | 22050.23438      | 0.0011% |
|                 | 44100          | 16-bit      | 302                                  | 2                     | 53     | 1   | 44100.46875      | 0.0011% |
|                 |                | 32-bit      | 429                                  | 4                     | 19     | 0   | 44099.50781      | 0.0011% |
|                 | 192000         | 16-bit      | 424                                  | 3                     | 11     | 1   | 192028.9844      | 0.0151% |
|                 |                | 32-bit      | 258                                  | 3                     | 3      | 1   | 191964.2813      | 0.0186% |
| Output Enabled  | 8000           | -           | 256                                  | 5                     | 12     | 1   | 8000             | 0.0000% |
|                 | 16000          | -           | 426                                  | 4                     | 13     | 0   | 16000.60059      | 0.0038% |
|                 | 32000          | -           | 426                                  | 4                     | 6      | 1   | 32001.20117      | 0.0038% |
|                 | 48000          | -           | 258                                  | 3                     | 3      | 1   | 47991.07031      | 0.0186% |
|                 | 96000          | -           | 344                                  | 2                     | 3      | 1   | 95982.14063      | 0.0186% |
|                 | 22050          | -           | 429                                  | 4                     | 9      | 1   | 22049.75291      | 0.0011% |
|                 | 44100          | -           | 271                                  | 2                     | 6      | 0   | 44108.07422      | 0.0183% |



#### Note

The frequency multiplication coefficient can be PLLAN/PLLHN, when the frequency multiplication coefficient is PLLAN, the output frequency division ratio can be PLLAP/PLLAQ/PLLAR, and when the frequency multiplication coefficient is PLLHN, the output frequency division ratio can be PLLHP/PLLHQ/PLLHR. For specific settings, refer to "CMU PLLA Configuration Register (CMU\_PLLACFGR)" and "CMU PLLH Configuration Register (CMU\_PLLHCFGR)" of the "Clock Module Unit (CMU)" chapter.

## 33.3.6 Master/Slave Mode

### 33.3.6.1 I2S Master Mode

I2S can be configured as follows:

- Transmit master or receive master (using I2S half-duplex mode).
- Simultaneous transmit and receive master (using I2S full-duplex mode).

I2S works in the master mode, the serial clock is output by the pin CK, and the word selection signal is generated by the pin WS. It can choose to output or not output the drive clock (MCK) by setting the I2S\_CTRL.MCKOE bit.

**Steps:**

- Step1.** Set the pins that I2S needs to use.
- Step2.** Select the clock source through the I2S\_CTRL.CLKSEL, I2S\_CTRL.I2SPLLSEL bit.
- Step3.** Set the I2S\_PR.I2SDIV bit and the I2S\_CTRL.ODD bit to define the serial data baud rate for proper audio sampling frequency.
- Step4.** Set the I2S\_CFGR register, select the I2S standard through the I2SSTD and PCMSYNC bit, select the data length through the DATLEN bit and select the number of bits per channel by configuring the CHLEN bit.
- Step5.** If it need to use interrupt, set the interrupt register of the system.
- Step6.** If it need to use DMA, set the relevant register for DMA.
- Step7.** Set I2S\_CTRL register, including working mode setting, communication mode setting, clock output permission setting, data output permission setting, FIFO reset setting, sending/receiving buffer threshold setting, etc., working mode WMS bit select I2S master mode.
- Step8.** Set the interrupt permission bit.
- Step9.** Set I2S\_CTRL.TXE and I2S\_CTRL.RXE, and the action starts.

**Note**

When using TXIRQOUT interrupt to write communication data to TXBUF, if each interrupt writes two data, first turn off the transmit interrupt enable flag TXIE after the interrupt starts, and then enable TXIE after writing the data. Or write only one data per interrupt.

**Transmit Sequence**

The I2S\_CTRL.TXE bit is set to 1 to enable transmission. The transmit sequence begins as soon as the data is written to the transmit buffer. A complete frame means that the left channel data is sent first and then the right channel data is sent. There are no partial frames where only the left channel is sent. During the first bit transmission, the data is loaded into the shift register in parallel, then shifted in serial and output to the SD pin (MSB first). See [Supported Audio Protocols](#) for more details on write operations in the various I2S standard modes.

**Receive Sequence**

This working mode is basically the same as the transmit mode, only the transmitting and receiving settings of the I2S\_CTRL register are different, and the RXE bit is set to 1 to allow reception. See [Supported Audio Protocols](#) for more details on read operations in the various I2S standard modes. If new data is received while previously received data has not been read, an overflow error will be generated and the I2S\_ER.RXERR flag will be set to 1. If the I2S\_CTRL.EIE bit is set to 1, an interrupt will be generated to indicate the error.

**33.3.6.2 I2S Slave Mode**

I2S can be configured as follows:

- Transmit slave or receive slave (using I2S half-duplex mode).
- Simultaneous transmit and receive slave (using I2S full-duplex mode).

The rules followed by this working mode are basically the same as the I2S master mode. In slave mode, the I2S interface does not generate a clock. Clock and WS signals are input from an external master device connected to the I2S interface. This way, the user does not need to configure the clock.

**Steps:**

- Step1.** Set the pins that I2S needs to use.



- Step2.** Set the I2S\_CFGR register, select the I2S standard through the I2SSTD and PCMSYNC bits, select the data length through the DATLEN bit and select the number of bits per channel by configuring the CHLEN bit.
- Step3.** If it need to use interrupt, set the interrupt register of the system.
- Step4.** If it need to use DMA, set the relevant register for DMA.
- Step5.** If it want to transmit data, 1-4 data to be sent should be pre-written into I2S\_TXBUF.
- Step6.** Set I2S\_CTRL register, including working mode setting, communication mode setting, clock output permission setting, data output permission setting, FIFO reset setting, transmitting/receiving buffer threshold setting, etc., working mode WMS bit select I2S slave mode.
- Step7.** Set the interrupt permission bit.
- Step8.** Set I2S\_CTRL.TXE and I2S\_CTRL.RXE, and the action starts.

### Transmit Sequence

The transmit sequence begins when WMS is set to 1, TXE is set to 1, and the external master transmits the clock and requests data transfer via the WS signal. When communication starts, data is transferred from the transmit buffer to the shift register. During first bit transmission, data is loaded from the internal bus into the shift register in parallel, then shifted in serial and output to the SD pin (MSB first). Every time the transmit buffer FIFO space is larger than the set threshold, if the I2S\_CTRL.TXIE bit is set to 1, an interrupt will be generated. See [Supported Audio Protocols](#) for more details on write operations in the various I2S standard modes.

In order to ensure continuous audio data transmission, the next data to be transmitted must be written into the TX FIFO before the end of the current data transmission. If the first clock edge of the next data communication arrives when the data has not been written into the TX FIFO, a transmit underflow will occur, the I2S\_ER.TXERR flag will be set to 1 and an interrupt may be generated. If the I2S\_CTRL.EIE bit is set to 1, an interrupt will be generated when the I2S\_ER.TXERR flag becomes 1.

### Receive Sequence

This working mode is basically the same as the transmitting mode, only the transmitting and receiving settings of the I2S\_CTRL register are different, and the RXE bit is set to 1 to allow reception. See [Supported Audio Protocols](#) for more details on read operations in the various I2S standard modes. If new data is received while previously received data has not been read, a receive overflow error will be generated and the RXERR flag will be set to 1. If the I2S\_CTRL.EIE bit is set to 1, an interrupt will be generated to indicate the error.

#### 33.3.7 Interrupt

The interrupt sources of I2S include transmitting buffer active space greater than the alarm threshold, receiving buffer active space less than the alarm threshold, receiving overflow, transmit underflow and transmit overflow. The transmit underflow and transmit overflow are integrated into the I2S transmit error interrupt TXERR, so it is necessary to judge the actual interrupt source through the flag. The specific description of the I2S interrupt sources is shown in the following table. Once the interrupt condition is established, the corresponding interrupt request is generated.

Table 33-4 I2S Interrupt Request

| Interrupt Event                                                             | Event flag | Enable Control Bit |
|-----------------------------------------------------------------------------|------------|--------------------|
| Active space of the transmitting buffer is greater than the alarm threshold | TXBA       | TXIE               |

| Interrupt Event                                                                                     | Event flag | Enable Control Bit |
|-----------------------------------------------------------------------------------------------------|------------|--------------------|
| Active space of the receiving buffer is less than the alarm threshold                               | RXBA       | RXIE               |
| Receiving data area is full and there is still a request to write data, and the receiving overflows | RXERR      | EIE                |
| Transmitting data area is empty and there are still transmitting requests, transmitting underflow   | TXERR      | EIE                |
| Transmitting data area is full and there is still a request to write data, transmitting overflow    | TXERR      | EIE                |

Users can write the vector corresponding to the event trigger source into different trigger object registers to implement various event trigger functions. For the vectors corresponding to the above event trigger sources, please refer to the "Interrupt Event List" of "Interrupt Controller (INTC) " chapter.

## 33.4 Register Description

### 33.4.1 Register List

- I2S1 base address: 0x4001E000
- I2S2 base address: 0x4001E400
- I2S3 base address: 0x40022000
- I2S4 base address: 0x40022400

Table 33-5 Inter\_IC Sound Bus (I2S) Register Offset

| Offset | Register  | Description                            | Width |
|--------|-----------|----------------------------------------|-------|
| 0x000  | I2S_CTRL  | I2S Control Register                   | 32    |
| 0x004  | I2S_SR    | I2S Status Register                    | 32    |
| 0x008  | I2S_ER    | I2S Error Status Register              | 32    |
| 0x00C  | I2S_CFGR  | I2S Configuration Register             | 32    |
| 0x010  | I2S_TXBUF | I2S Transmit Buffer FIFO Data Register | 32    |
| 0x014  | I2S_RXBUF | I2S Receive Buffer FIFO Data Register  | 32    |
| 0x018  | I2S_PR    | I2S Prescaler Register                 | 32    |



#### Note

Only 32-bit register writing is supported.

## 33.4.2 I2S Control Register (I2S\_CTRL)

| Offset | Bit Position |    |         |      |          |    |        |        |      |        |      |           |          |       |          |          |    |          |          |    |       |     |     |     |      |     |      |     |    |    |    |    |
|--------|--------------|----|---------|------|----------|----|--------|--------|------|--------|------|-----------|----------|-------|----------|----------|----|----------|----------|----|-------|-----|-----|-----|------|-----|------|-----|----|----|----|----|
| 0x000  | 31           | 30 | 29      | 28   | 27       | 26 | 25     | 24     | 23   | 22     | 21   | 20        | 19       | 18    | 17       | 16       | 15 | 14       | 13       | 12 | 11    | 10  | 9   | 8   | 7    | 6   | 5    | 4   | 3  | 2  | 1  | 0  |
| Reset  | 0x00004400   |    |         |      |          |    |        |        |      |        |      |           |          |       |          |          |    |          |          |    |       |     |     |     |      |     |      |     |    |    |    |    |
| Name   | Reserved     |    | SYNCSEL | SRST | Reserved |    | CLKSEL | DUPLEX | CKOE | LRCKOE | SDOE | I2SPLLSEL | Reserved | FIFOR | Reserved | RXBIRQWL |    | Reserved | TXBIRQWL |    | MCKOE | ODD | WMS | EIE | RXIE | RXE | TXIE | TXE |    |    |    |    |
| Access |              |    | RW      | RW   |          |    | RW     | RW     | RW   | RW     | RW   | RW        |          | RW    |          | RW       |    |          | RW       |    | RW    | RW  | RW  | RW  | RW   | RW  | RW   | RW  | RW | RW | RW | RW |

| Bit   | Name      | Function                              | Description                                                                                                                                                                                                                                                                                                                                                                                        | Access |
|-------|-----------|---------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:29 | Reserved  | Reserved                              | -                                                                                                                                                                                                                                                                                                                                                                                                  | -      |
| 28    | SYNCSEL   | 4-way synchronous control             | <ul style="list-style-type: none"> <li>0b0: 4-way synchronous control disabled</li> <li>0b1: 4-way synchronous control enabled</li> </ul> <div>  <b>Note</b><br/>           Only unit 1 supports the SYNCSEL function bit. This bit of other units is reserved, please retain the initial value 0.         </div> | RW     |
| 27    | SRST      | Software reset                        | <ul style="list-style-type: none"> <li>0b0: Release reset</li> <li>0b1: Software reset</li> </ul>                                                                                                                                                                                                                                                                                                  | RW     |
| 26:24 | Reserved  | Reserved                              | -                                                                                                                                                                                                                                                                                                                                                                                                  | -      |
| 23    | CLKSEL    | Clock source selection                | <ul style="list-style-type: none"> <li>0b0: Select I2SPLL</li> <li>0b1: Select external clock</li> </ul>                                                                                                                                                                                                                                                                                           | RW     |
| 22    | DUPLEX    | Communication method selection        | <ul style="list-style-type: none"> <li>0b0: Half-Duplex</li> <li>0b1: Full-duplex</li> </ul>                                                                                                                                                                                                                                                                                                       | RW     |
| 21    | CKOE      | Communication clock output permission | <ul style="list-style-type: none"> <li>0b0: Output disabled</li> <li>0b1: Output enabled</li> </ul>                                                                                                                                                                                                                                                                                                | RW     |
| 20    | LRCKOE    | Channel clock output permission       | <ul style="list-style-type: none"> <li>0b0: Output disabled</li> <li>0b1: Output enabled</li> </ul>                                                                                                                                                                                                                                                                                                | RW     |
| 19    | SDOE      | Data output permission                | <ul style="list-style-type: none"> <li>0b0: Output disabled</li> <li>0b1: Output enabled</li> </ul>                                                                                                                                                                                                                                                                                                | RW     |
| 18    | I2SPLLSEL | I2SPLL input selection                | <ul style="list-style-type: none"> <li>0b0: Input disabled</li> <li>0b1: Input enabled</li> </ul>                                                                                                                                                                                                                                                                                                  | RW     |
| 17    | Reserved  | Reserved                              | -                                                                                                                                                                                                                                                                                                                                                                                                  | -      |
| 16    | FIFOR     | FIFO reset                            | <ul style="list-style-type: none"> <li>0b0: Release reset</li> <li>0b1: Software reset</li> </ul>                                                                                                                                                                                                                                                                                                  | RW     |
| 15    | Reserved  | Reserved                              | -                                                                                                                                                                                                                                                                                                                                                                                                  | -      |

| Bit   | Name            | Function                                | Description                                                                                                                                                                                                                                                                                                                                                                                      | Access |
|-------|-----------------|-----------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 14:12 | RXBIRQWL        | Receive buffer interrupt request level  | Interrupt request triggers when the available space is less than the set value:<br>Can only be set to 0x0-0x4 (FIFO space is 4)                                                                                                                                                                                                                                                                  | RW     |
| 11    | <i>Reserved</i> | <i>Reserved</i>                         | -                                                                                                                                                                                                                                                                                                                                                                                                | -      |
| 10:8  | TXBIRQWL        | Transmit buffer interrupt request level | Interrupt request triggers when the available space is more than the set value:<br>Can only be set to 0x0-0x4 (FIFO space is 4)                                                                                                                                                                                                                                                                  | RW     |
| 7     | MCKOE           | Drive clock output enabled              | <ul style="list-style-type: none"> <li>0b0: Disable drive clock output</li> <li>0b1: Enable drive clock output</li> </ul> <div>  <b>Note</b><br/>Only I2S master model support this function.         </div>                                                                                                    | RW     |
| 6     | ODD             | Prescaler odd factor                    | <ul style="list-style-type: none"> <li>0b0: Actual division value=I2SDIV*2</li> <li>0b1: Actual division value=I2SDIV*2+1</li> </ul> <div>  <b>Note</b><br/>Only I2S master model support this function.<br/>To set ODD to 1, set the I2S_PR register first, and then set the I2S_CTRL register.         </div> | RW     |
| 5     | WMS             | I2S working mode selection              | <ul style="list-style-type: none"> <li>0b0: I2S master mode</li> <li>0b1: I2S slave mode</li> </ul>                                                                                                                                                                                                                                                                                              | RW     |
| 4     | EIE             | Communication error interrupt enabled   | <ul style="list-style-type: none"> <li>0b0: Communication error interrupt disabled</li> <li>0b1: Communication error interrupt enabled</li> </ul>                                                                                                                                                                                                                                                | RW     |
| 3     | RXIE            | Receive interrupt enabled               | <ul style="list-style-type: none"> <li>0b0: Receive interrupt disabled</li> <li>0b1: Receive interrupt enabled</li> </ul>                                                                                                                                                                                                                                                                        | RW     |
| 2     | RXE             | Receive enabled                         | <ul style="list-style-type: none"> <li>0b0: Reception disabled</li> <li>0b1: Reception enabled</li> </ul>                                                                                                                                                                                                                                                                                        | RW     |
| 1     | TXIE            | Transmit interrupt enabled              | <ul style="list-style-type: none"> <li>0b0: Transmit interrupt disabled</li> <li>0b1: Transmit interrupt enabled</li> </ul>                                                                                                                                                                                                                                                                      | RW     |
| 0     | TXE             | Transmit enabled                        | <ul style="list-style-type: none"> <li>0b0: Transmission disabled</li> <li>0b1: Transmission enabled</li> </ul>                                                                                                                                                                                                                                                                                  | RW     |

**Note**

Before resetting the FIFO using FIFOR, the I2S module must be disabled using TXE and RXE.

## 33.4.3 I2S Status Register (I2S\_SR)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |    |      |    |      |    |      |    |      |    |      |    |      |    |      |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|----|------|----|------|----|------|----|------|----|------|----|------|----|------|
| 0x004  | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5  | 4    | 3  | 2    | 1  | 0    |    |      |    |      |    |      |    |      |
| Reset  | 0x00000014   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |    |      |    |      |    |      |    |      |    |      |    |      |    |      |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |    | RXID |    | RXBF |    | RXBE |    | TXBF |    | TXBE |    | RXBA |    | TXBA |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   | RO |      | RO |      | RO |      | RO |      | RO |      | RO |      | RO |      |

| Bit  | Name     | Function                                       | Description                                                                                                              | Access |
|------|----------|------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------|--------|
| 31:7 | Reserved | Reserved                                       | -                                                                                                                        | -      |
| 6    | RXID     | Receiving Data ID                              | <ul style="list-style-type: none"> <li>0b0: Left channel data</li> <li>0b1: Right channel data</li> </ul>                | RO     |
| 5    | RXBF     | Receive buffer full                            | <ul style="list-style-type: none"> <li>0b0: Receive buffer not full</li> <li>0b1: Receive buffer full</li> </ul>         | RO     |
| 4    | RXBE     | Receive buffer empty                           | <ul style="list-style-type: none"> <li>0b0: Receive buffer not empty</li> <li>0b1: Receive buffer empty</li> </ul>       | RO     |
| 3    | TXBF     | Transmit buffer full                           | <ul style="list-style-type: none"> <li>0b0: Transmit buffer not full</li> <li>0b1: Transmit buffer full</li> </ul>       | RO     |
| 2    | TXBE     | Transmit buffer empty                          | <ul style="list-style-type: none"> <li>0b0: Transmit buffer not empty</li> <li>0b1: Transmit buffer empty</li> </ul>     | RO     |
| 1    | RXBA     | Receive buffer alarm (related to water level)  | <ul style="list-style-type: none"> <li>0b0: Receive buffer without alarm</li> <li>0b1: Receive buffer alarm</li> </ul>   | RO     |
| 0    | TXBA     | Transmit buffer alarm (related to water level) | <ul style="list-style-type: none"> <li>0b0: Transmit buffer without alarm</li> <li>0b1: Transmit buffer alarm</li> </ul> | RO     |

**Note**

In addition to FIFO and shift register, there are two buffer units inside I2S. The FIFO flag only indicates the data status in the FIFO, not all the stored data in the I2S.

## 33.4.4 I2S Error Status Register (I2S\_ER)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |       |  |       |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|-------|--|-------|
| 0x008  | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0     |  |       |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |       |  |       |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   | RXERR |  | TXERR |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   | RW    |  | RW    |

| Bit  | Name     | Function            | Description                                                                                                                                                                                                                                                                                                                                                                                      | Access |
|------|----------|---------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:2 | Reserved | Reserved            | -                                                                                                                                                                                                                                                                                                                                                                                                | -      |
| 1    | RXERR    | Receive error flag  | <p>Write 1 to clear the flag, write 0 has no effect</p> <p>Read:</p> <ul style="list-style-type: none"> <li>0b0: No receive error occurred</li> <li>0b1: A receive error occurred</li> </ul> <div>  <b>Note</b><br/>           When a receive overflow occurs, RXERR is set to 1.         </div>                | RW     |
| 0    | TXERR    | Transmit error flag | <p>Write 1 to clear the flag, write 0 has no effect</p> <p>Read:</p> <ul style="list-style-type: none"> <li>0b0: No transmit error occurred</li> <li>0b1: A transmit error occurred</li> </ul> <div>  <b>Note</b><br/>           TXERR is set to 1 when a transmit overflow/under-flow occurs.         </div> | RW     |

## 33.4.5 I2S Configuration Register (I2S\_CFGR)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |         |   |       |   |        |   |        |  |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---------|---|-------|---|--------|---|--------|--|
| 0x00C  | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5       | 4 | 3     | 2 | 1      | 0 |        |  |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |         |   |       |   |        |   |        |  |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   | PCMSYNC |   | CHLEN |   | DATLEN |   | I2SSTD |  |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   | RW      |   | RW    |   | RW     |   | RW     |  |

| Bit  | Name     | Function                             | Description                                                                                                                                                                                                                                                                                                                  | Access |
|------|----------|--------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:6 | Reserved | Reserved                             | -                                                                                                                                                                                                                                                                                                                            | -      |
| 5    | PCMSYNC  | PCM frame synchronization            | <ul style="list-style-type: none"> <li>0b0: Short frame synchronization</li> <li>0b1: Long frame synchronization</li> </ul> <div>  <b>Note</b><br/>           This bit is enable only when I2SSTD=0b11 (using PCM standard).         </div> | RW     |
| 4    | CHLEN    | Mono one frame data length selection | <ul style="list-style-type: none"> <li>0b0: 16-bit</li> <li>0b1: 32-bit</li> </ul>                                                                                                                                                                                                                                           | RW     |
| 3:2  | DATLEN   | Transmission data length selection   | <ul style="list-style-type: none"> <li>0b00: 16-bit</li> <li>0b01: 24-bit</li> <li>0b10/0b11: 32-bit</li> </ul>                                                                                                                                                                                                              | RW     |
| 1:0  | I2SSTD   | Communication protocol selection     | <ul style="list-style-type: none"> <li>0b00: Philips protocol</li> <li>0b01: MSB justified protocol (left alignment)</li> <li>0b10: LSB justified protocol (right alignment)</li> <li>0b11: PCM protocol</li> </ul>                                                                                                          | RW     |



## 33.4.6 I2S Transmit Buffer FIFO Data Register (I2S\_TXBUF)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0x010  | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | TXBUF        |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access | WO           |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit   | Name  | Function      | Description                                                                                                                                                                                                                                                                                                       | Access |
|-------|-------|---------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31: 0 | TXBUF | Transmit Data | Store transmitted data <ul style="list-style-type: none"> <li>For 16-bit frame, TXBUF[15:0] stores one frame of left channel or one frame of right channel transmission data.</li> <li>For 32-bit frame, TXBUF[31:0] stores one frame of left channel or one frame of right channel transmission data.</li> </ul> | WO     |

## 33.4.7 I2S Receive Buffer FIFO Data Register (I2S\_RXBUF)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0x014  | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | RXBUF        |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access | RO           |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit  | Name  | Function     | Description                                                                                                                                                                                                                                                                                             | Access |
|------|-------|--------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:0 | RXBUF | Receive Data | Store received data: <ul style="list-style-type: none"> <li>For 16-bit frame, RXBUF[15:0] stores one frame of left channel or one frame of right channel received data.</li> <li>For 32-bit frame, RXBUF[31:0] stores one frame of left channel or one frame of right channel received data.</li> </ul> | RO     |

## 33.4.8 I2S Prescaler Register (I2S\_PR)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |        |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|--------|---|---|---|---|---|---|---|
| 0x018  | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7      | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000002   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |        |   |   |   |   |   |   |   |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   | I2SDIV |   |   |   |   |   |   |   |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   | RW     |   |   |   |   |   |   |   |

| Bit  | Name     | Function                  | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | Access |
|------|----------|---------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:8 | Reserved | Reserved                  | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | -      |
| 7:0  | I2SDIV   | Frequency Division Factor | The actual division value= $I2SDIV*2+I2S\_CTRL.ODD$ . For details, refer to the <a href="#">Clock Generator</a> : <ul style="list-style-type: none"> <li>0x0: Setting prohibited</li> <li>0x1: Setting prohibited</li> <li>0x2: 2 frequency division</li> <li>0x3: 3 frequency division</li> <li>.....</li> <li>0xFE: 254 frequency division</li> <li>0xFF: 255 frequency division</li> </ul> <div>  <b>Note</b><br/>           Only I2S master model support this function.         </div> | RW     |

## 33.5 Precautions

### 33.5.1 Precautions when Using as a Master

1. When I2S acts as the master to transmit data only, if all the data in I2S\_TXBUF has been transmitted and no new data is written, I2S will pause the action after the last data is transmitted, and I2S will no longer generate communication clock at this time, the transmit error flag bit TXERR will be set to 1. At this time, the user can choose to write the I2S\_CTRL.TXE bit as 0 to close the I2S, or write new transmitting data to I2S\_TXBUF to continue transmitting. When continuing to transmit, WS will restart from the left channel (Philips, MSB/LSB mode).

If the I2S\_CTRL.TXE bit is directly written as 0 during the transmitting action, the I2S will be turned off immediately, and the current data transmitting will be terminated. This approach will lead to the inability to grasp the state of the slave, and the slave will receive data confusion when restarting communication without resetting the slave. Therefore, it is recommended that the user write the I2S\_CTRL.TXE bit to 0 to close the I2S when the I2S is in the paused state.

2. When I2S is used as the master to only receive data, if you want to temporarily stop the receiving action, it can write two frames of dummy data in advance, and set I2S\_CTRL.TXE to 1 when it need to pause and count to the corresponding position. When the baud rate is 8kHz-96kHz, advance 4 pieces of data TXE are set to 1, and when the baud rate is 192kHz, 5 pieces of data TXE are set to 1 in advance. After the two frames of dummy data are transmitted, I2S will pause. At this time, I2S will no longer generate a communication clock, and the transmission error flag TXERR will be is set to 1. At this time, the user can choose to clear I2S\_CTRL.TXE and I2S\_CTRL.RXE to close I2S, or write another frame of dummy data to I2S\_TXBUF to restart the communication action, and turn off TXE when restarting to receive the first data after the pause, and WS will restart from the left channel when restart the action (Philips, MSB/LSB mode). When the communication clock is regenerated, clear I2S\_CTRL.TXE to return to the receive-only state. Please refer to the figure below for details.

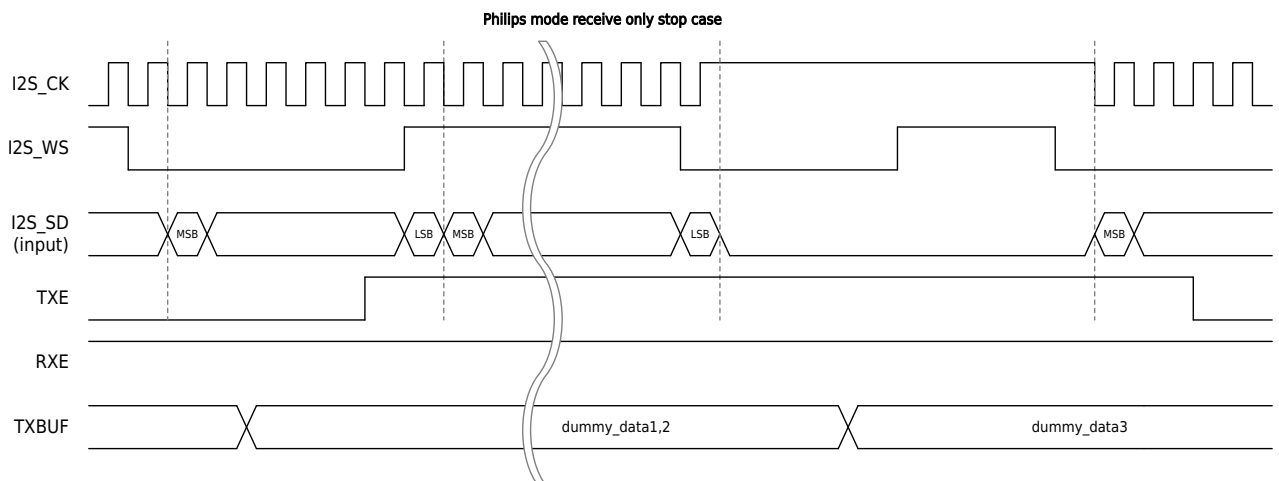


Figure 33-16 Master only Receives and Temporarily Stops Receiving

If the I2S\_CTRL.RXE bit is directly written to 0 during the receiving action; the I2S will be turned off immediately, and the current data reception will be terminated. This approach will lead to the inability to grasp the status of the slave. When communication is restarted without resetting the slave, it causes confusion in the data transmitted by the slave. Therefore, it is recommended that the user clear the I2S\_CTRL.TXE and I2S\_CTRL.RXE bits when the I2S is in a paused state to disable I2S.

3. When I2S acts as the master in full-duplex operation, if all the data in I2S\_TXBUF has been transmitted and no new data is written, I2S will suspend the action after the last data is transmitted, and I2S

will no longer generate the communication clock; the transmit error flag bit TXERR will be set to 1. At this time, the user can choose to clear I2S\_CTRL.TXE and I2S\_CTRL.RXE to close I2S, or write new transmitting data to I2S\_TXBUF to continue transmitting and receiving operations. When continuing to transmit, WS will restart from the left channel (Philips, MSB/LSB mode).

If the I2S\_CTRL.TXE and I2S\_CTRL.RXE bits are directly written as 0 during full-duplex operation; the I2S will be turned off immediately, and the current data transmission and reception will be terminated. This approach will lead to the inability to grasp the status of the slave, and the communication data will be confused when restarting the communication without resetting the slave. Therefore, it is recommended that the user clear the I2S\_CTRL.TXE and I2S\_CTRL.RXE bits to close I2S when the I2S is in the paused state.

4. When I2S is used as the master to transmit PCM short frame data, when I2S is suspended because I2S\_TXBUF has no new data writing action, there are two setting methods to restart the transmission, which method to choose depends on the data reception specification of the slave to determine. If the slave needs to detect the status of WS every time it receives data, then after I2S pauses, you need to set I2S\_CTRL.TXE to 0 first, then write new send data to I2S\_TXBUF and then set I2S\_CTRL.TXE to 1 to restart the transmission. The specific actions are shown in the figure below.

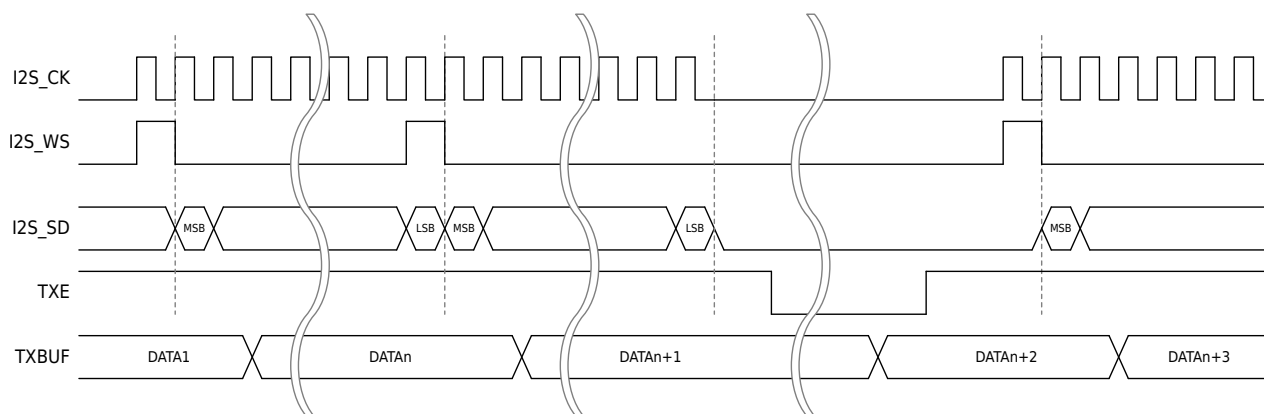


Figure 33-17 PCM Short Frame Master Resends After Suspension of Transmission Method 1

If the slave only detects the WS state when receiving the first frame of data, after the I2S pauses, it can directly write new transmitting data to I2S\_TXBUF to restart the transmission. The specific actions are shown in the figure below.

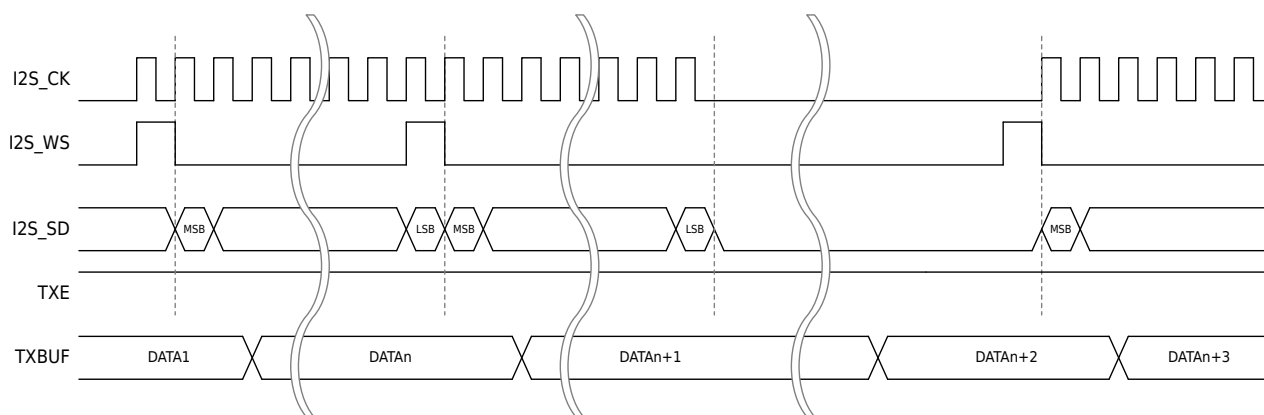


Figure 33-18 PCM Short Frame Master Resends After Suspension of Transmission Method 2

### 33.5.2 Precautions when Using as a Slave

1. When I2S acts as a slave, it is necessary to ensure that after all register configurations are completed, set I2S\_CTRL.TXE or I2S\_CTRL.RXE to start the slave operation.
2. When I2S starts the slave operation in Philips, MSB/LSB mode, it is necessary to ensure that the WS signal is at the right channel level state when starting. When I2S starts the slave operation in PCM mode, it is necessary to ensure that the WS signal is low level status when starting.
3. When the I2S receives data as a slave, the data received each time can only be read after the start of the next frame of data reception. Therefore, when the communication is paused or terminated, the last frame of data received by the I2S can only be read when the next communication begins.
4. When I2S performs data reception as a slave in Philips, MSB/LSB mode, it will check whether WS is the left channel level when receiving each frame of left channel data. When I2S is receiving data as a slave in PCM mode, before each frame of data is received, it will check whether WS has generated a active level according to the standard communication protocol.
5. When I2S is acting as a slave in full-duplex mode, if all data in I2S\_TXBUF has been transmitted and no new data has been written, the internal transceiver circuit of I2S will suspend operation (i.e., it will not be able to receive data) after the last data is sent, and the transmit error flag TXERR will be set to 1. At this time, the user can choose to use I2S\_CTRL.TXE and I2S\_CTRL.RXE to disable I2S, reset FIFO and set SRST for soft reset, then clear the error flag, and finally enable I2S to transmit and receive new data.

### 33.5.3 Anti-interference Support Function

1. In slave mode, when the received WS signal indication is inconsistent with the configured data channel length, data of each data channel starts to be received according to the WS signal indication to avoid accumulated errors caused by transmission interference.
2. Due to transmission interference, when the WS signal is less than 7 data channel lengths, packet loss may occur.

### 33.5.4 Data Channel Identification Function

1. In Philips/MSB/LSB slave mode, if you need to know whether the received data belongs to the left or right channel, you can first read I2S\_SR.RXID to get the data channel identifier, and then read the I2S\_RXBUF register to get the channel data.
2. Each read of the I2S\_RXBUF register automatically updates the I2S\_SR.RXID at the same time.

### 33.5.5 Four-way Synchronous Control Function

The process of using the four-way synchronization function is as follows:

- Step1.** Disable the clock and disable I2S through the I2S\_CTRL.I2SPLLSEL, I2S\_CTRL.TXE and I2S\_CTRL.RXE bits corresponding to the four-way I2S.
- Step2.** Make the configuration of the four-way I2S consistent and keep I2S\_CTRL.I2SPLLSEL, I2S\_CTRL.TXE and I2S\_CTRL.RXE closed.
- Step3.** Set I2S\_CTRL.SYNCSEL of I2S1.
- Step4.** Enable the clocks of the four-way I2S through I2S\_CTRL.I2SPLLSEL of I2S1.
- Step5.** Enable the TX and RX of four-way I2S through I2S\_CTRL.TXE and I2S\_CTRL.RXE of I2S1.
- Step6.** Read and write the buffer FIFO data registers corresponding to the four-way I2S to transmit and receive data.

**Note**

Only the selection of the internal clock (I2SPLL) through I2S\_CTRL.CLKSEL is supported.

### 33.5.6 Other Precautions

1. Recommended FIFO clearing process:
  - Disable I2S by I2S\_CTRL.TXE and I2S\_CTRL.RXE;
  - Reset FIFO by I2S\_CTRL.FIFOR;
  - Soft reset via I2S\_CTRL.SRST.
2. Recommended ERR clearing process:
  - Disable I2S by I2S\_CTRL.TXE and I2S\_CTRL.RXE;
  - Reset FIFO by I2S\_CTRL.FIFOR;
  - Soft reset via I2S\_CTRL.SRST;
  - Clear the ERR flag.
3. Description of each reset bit in the control register:
  - The I2S\_CTRL.SRST bit is used to reset the working circuit and clear the internal shift register, and will not affect the register configuration;
  - The I2S\_CTRL.FIFOR bit is used to reset the FIFO and will not affect other circuits and registers.
4. In addition to the shift register, the RX/TX circuit also has two caches that will obtain data from the RX/TX FIFO according to the situation. The cache is cleared via FIFOR.

## 34 USB2.0 High-Speed Module (USBHS)

### 34.1 Introduction

The USB High-Speed (USBHS) controller offers a range of USB communication solutions for portable devices. The USBHS controller supports both host mode and device mode. The chip integrates a full-speed PHY internally and supports an external high-speed PHY via the ULPI interface. In host mode, the USBHS controller supports high-speed (HS, 480 Mbit/s), full-speed (FS, 12 Mbit/s), and low-speed (LS, 1.5 Mbit/s) transceivers, while in device mode, it supports only high-speed (HS, 480 Mbit/s) and full-speed (FS, 12 Mbit/s) transceivers. The USBHS controller supports all four types of transfers defined by the USB 2.0 protocol (Control transfer, Bulk transfer, Interrupt transfer, and Isochronous transfer). The USBHS controller supports the LPM (Link Power Management) function.

The chip integrates a full-speed PHY that supports only full-speed (FS, 12 Mbit/s) and low-speed (LS, 1.5 Mbit/s) transfers. By connecting an external high-speed PHY via the ULPI interface, it supports high-speed (HS, 480 Mbit/s), full-speed (FS, 12 Mbit/s), and low-speed (LS, 1.5 Mbit/s) transfers.

The protocol followed is as follows:

- Universal Serial Bus Revision 2.0 Specification
- USB 2.0 Link Power Management Addendum Engineering Change Notice to the USB 2.0 specification, July 16, 2007
- Errata for USB 2.0 ECN: Link Power Management (LPM) - 7/2007

### 34.2 Features

#### 34.2.1 General Features

- Supports two PHY interfaces
  - ▶ Built-in on-chip USB 2.0 full-speed PHY (needs to set USB\_SYCTLREG.USBHS\_FSPHYE=1 to enable).
  - ▶ ULPI interface for connecting an external high-speed PHY.
- Support host mode and device mode
- Supports HS/FS SOF and low-speed "Keep-alive" tokens with the following functions:
  - ▶ SOF pulse pin output function
  - ▶ The SOF pulse can serve as an internal event source to trigger operations of modules such as TIMER and DMA.
  - ▶ Configurable frame period
  - ▶ Configurable frame end interrupt
- The module embeds DMA and supports software-configurable AHB burst transmission types
- Supports power-saving functions, such as USB suspend, stopping RAM clock, and stopping PHY domain clock
- Equipped with a dedicated 8KB RAM featuring advanced FIFO control
  - ▶ The RAM space can be partitioned into different FIFOs for flexible and efficient utilization of RAM
  - ▶ Each FIFO can store multiple data packets
  - ▶ Dynamic allocation of memory area
  - ▶ The FIFO size can be configured to non-power-of-two values for continuous memory unit utilization
- Within one frame, no application intervention is required to achieve maximum USB bandwidth

- The mode (Host Mode or Device Mode) can be automatically determined based on the ID line level

### 34.2.2 Host Mode Features

- Host mode supports USB2.0 high-speed (HS, 480Mbit/s), full-speed (FS, 12Mbit/s), and low-speed (LS, 1.5Mbit/s) transmissions
- The VBUS voltage needs to be generated by an external power supply chip
- Up to 16 host channels (Pipes) : Each channel can be dynamically reconfigured to support any type of USB transfer
- Built-in hardware scheduler capable of:
  - ▶ Store up to 8 interrupt plus isochronous transfer requests in a periodic hardware queue
  - ▶ Store up to 8 control and bulk transfer requests in a non-periodic hardware queue
- Manages a shared RX FIFO, a periodic TX FIFO, and a non-periodic TX FIFO to efficiently utilize USB data RAM

### 34.2.3 Device Mode Features

- Slave mode supports USB2.0 high-speed (HS, 480Mbit/s) and full-speed (FS, 12Mbit/s) transmission
- 1 bidirectional control endpoint 0
- 15 OUT endpoints, configurable to support bulk transfer, interrupt transfer, or isochronous transfer.
- 15 IN endpoints, configurable to support bulk transfer, interrupt transfer, or isochronous transfer.
- Contains 16 transmit FIFOs (one for each IN endpoint) and one receive FIFO (shared by all OUT endpoints)
- Supports remote wake-up function
- Supports soft disconnect function
- The VBUS PIN supports 5V withstand voltage



## 34.3 Functional Description

### 34.3.1 System Block Diagram

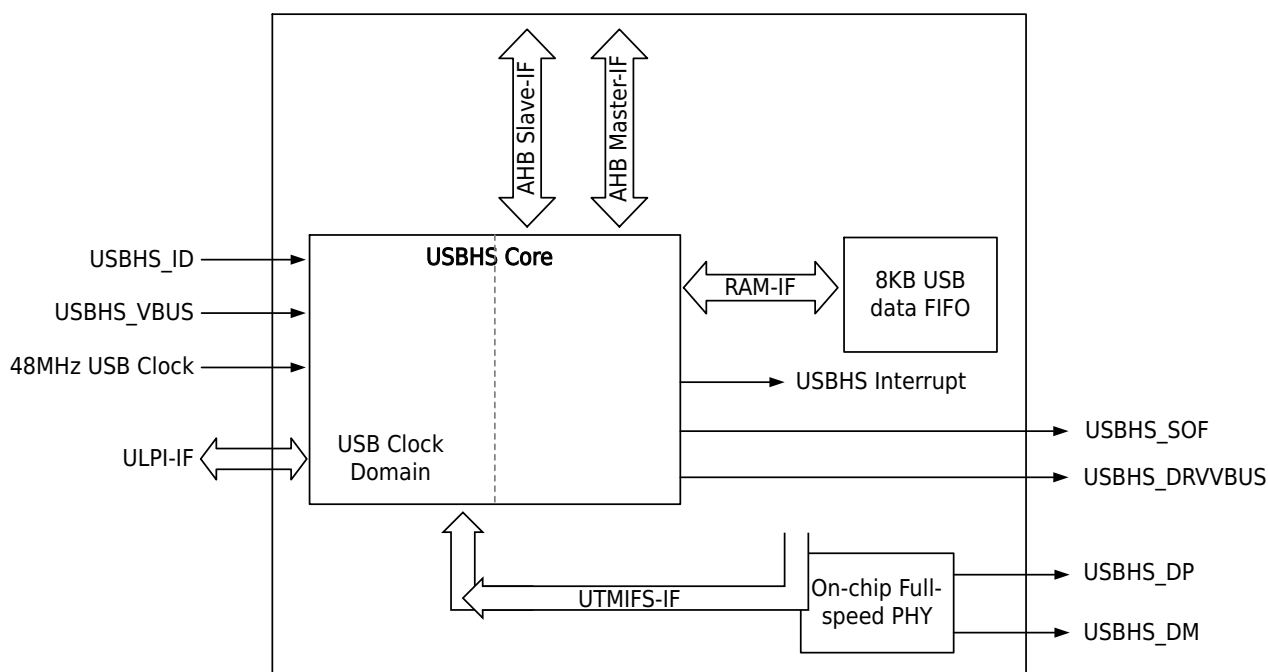


Figure 34-1 USBHS System Block Diagram

### 34.3.2 Pin Description

Table 34-1 USBHS Pin Description

| Pin Name       | Direction    | Applicable Mode | Corresponding Pins | Function Description              | Applicable PHY                     |
|----------------|--------------|-----------------|--------------------|-----------------------------------|------------------------------------|
| USBHS_VBUS     | Input        | Equipment       | PB13               | Power port, 5V with-stand voltage | On-chip full-speed PHY             |
| USBHS_DP       | Input/Output | Host, de-vice   | PB15               | Differential data D+              | On-chip full-speed PHY             |
| USBHS_DM       | Input/Output | Host, de-vice   | PB14               | Differential data D-              | On-chip full-speed PHY             |
| USBHS_DRVVBUS  | Output       | Host            | PD8                | External power chip enable        | On-chip full-speed PHY             |
| USBHS_ID       | Input        | Host            | PB12               | USB A-B device identification     | On-chip full-speed PHY             |
| USBHS_SOF      | Output       | Host, de-vice   | PA4                | SOF output pulse                  | On-chip full-speed PHY<br>ULPI PHY |
| USBHS_ULPI_CLK | Input        | Host, de-vice   | PE12, PA5          | ULPI Clock                        | ULPI PHY                           |
| USBHS_ULPI_DIR | Input        | Host, de-vice   | PC2, PI11          | ULPI Direction                    | ULPI PHY                           |

| Pin Name       | Direction    | Applicable Mode | Corresponding Pins | Function Description | Applicable PHY |
|----------------|--------------|-----------------|--------------------|----------------------|----------------|
| USBHS_ULPI_NXT | Input        | Host, device    | PC3, PH4           | ULPI Next            | ULPI PHY       |
| USBHS_ULPI_STP | Output       | Host, device    | PC0                | ULPI Stop            | ULPI PHY       |
| USBHS_ULPI_D0  | Input/Output | Host, device    | PE13, PA3          | ULPI Data 0          | ULPI PHY       |
| USBHS_ULPI_D1  | Input/Output | Host, device    | PE14, PB0          | ULPI Data 1          | ULPI PHY       |
| USBHS_ULPI_D2  | Input/Output | Host, device    | PE15, PB1          | ULPI Data 2          | ULPI PHY       |
| USBHS_ULPI_D3  | Input/Output | Host, device    | PB10               | ULPI Data 3          | ULPI PHY       |
| USBHS_ULPI_D4  | Input/Output | Host, device    | PB11, PD9          | ULPI Data 4          | ULPI PHY       |
| USBHS_ULPI_D5  | Input/Output | Host, device    | PB12               | ULPI Data 5          | ULPI PHY       |
| USBHS_ULPI_D6  | Input/Output | Host, device    | PB13               | ULPI Data 6          | ULPI PHY       |
| USBHS_ULPI_D7  | Input/Output | Host, device    | PB5, PE11          | ULPI Data 7          | ULPI PHY       |

Since the USBHS\_DP and USBHS\_DM pins are multiplexed with universal GPIO, it is recommended to disable the digital function of the corresponding pins when using the USBHS on-chip full-speed PHY. The USBHS\_DP and USBHS\_DM functions are analog functions, independent of the associated PFSR register settings. For details, see the relevant description in the "General-Purpose I/O (GPIO) " chapter. In addition, when the USBHS function is not used, the digital function pin corresponding to the USBHS\_DP and USBHS\_DM pins consumes additional current when flipped.

### 34.3.3 Clock and Work Mode

The clocks used by USBHS are as follows:

- For a 48MHz clock. The internal PLL circuit generates this 48MHz clock. The PLL clock source must select the external high-speed oscillator. Before using the USBHS module, the USBHS clock must be properly configured in the CMU module.
- For a 60MHz clock, use the UPLI interface to connect to the input clock of the high-speed PHY. It is used only when the external PHY is connected.
- The module clock PCLK1 needs to be configured in the CMU module, and the clock must be greater than or equal to 60MHz.

USBHS can operate as either a host or device, and includes an on-chip full-speed PHY and a ULPI interface for an external high-speed PHY.

The pull-up and pull-down resistors are integrated inside the on-chip full-speed PHY, and USBHS can automatically select them based on the current mode and connection status.

When USBHS uses the on-chip full-speed PHY, the VCC voltage range is 3.0-3.63V.

### 34.3.4 Mode Decision

USBHS determines the current operating mode using the following two methods:

- Method 1: Automatic identification based on the status of the USBHS\_ID line. When the USBHS\_ID line is detected at high level, the module works in the device mode. When the USBHS\_ID line is detected at low level, the module works in host status.
- Method 2: Force host/device mode. By setting the FDMOD or FHMOD bit in the USBHS\_GUSBCFG register to 1, the module will ignore the USBHS\_ID line level and forcibly worked in device or host mode.

### 34.3.5 Host Function

#### Function Introduction

When USBHS works in host mode, VBUS is the 5V power supply pin specified by the USB protocol. The internal PHY does not support 5V power supply, so an external USB power chip is required to power the device. USBHS\_DRVVBUS is used to enable the external USB power chip. The overcurrent detection of the external power supply chip can be implemented through the external interrupt IRQ of this MCU. In host mode, USBHS\_VBUS can be used as GPIO.

A typical USB host mode system architecture diagram is as follows:

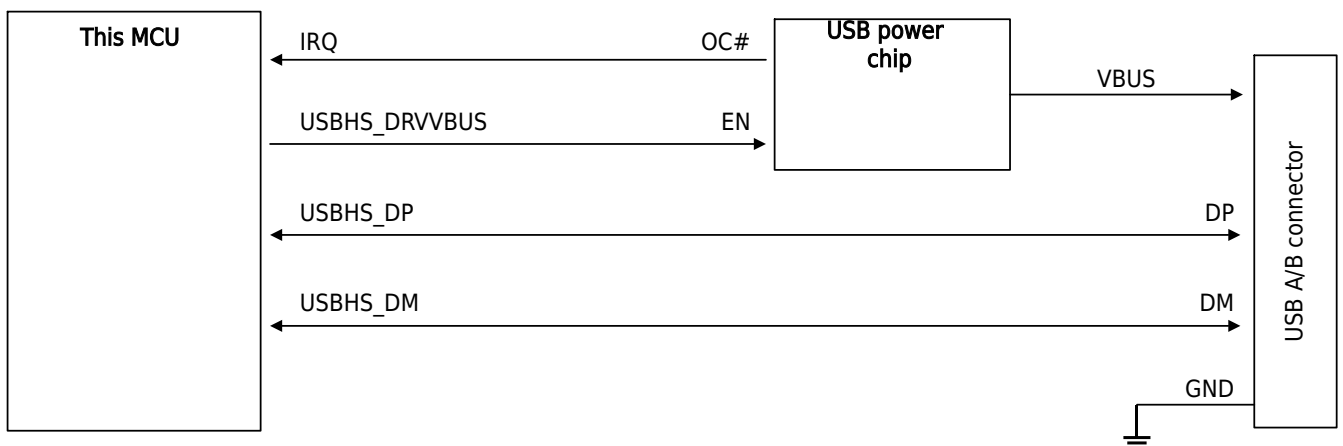


Figure 34-2 USBHS host Mode System Architecture Diagram

#### Host Port Power Supply

This MCU cannot output 5V to supply VBUS. To achieve this, a USB power chip or a basic power switch (if the circuit board provides 5V power) must be added externally to the microcontroller to drive the 5V VBUS line. The external power chip can be driven by any GPIO output or USBHS\_DRVVBUS. When the application determines to use GPIO to control an external device for providing VBUS, the port power bit (USBHS\_HPRT.PWPR) in the host port control and status register must still be set to 1.

#### Host Detects Device Connection and Disconnection

The USB device will be detected immediately after connection. The USBHS module will issue a host port interrupt signal, which is triggered by the device connection bit (USBHS\_HPRT.PCDET) in the host port control and status register.

A device disconnect event will trigger a disconnect detection interrupt (USBHS\_GINTSTS.DISCINT).

## Host Enumeration

After a device connection is detected, if a new device is connected, the host must initiate the enumeration process by sending a USB reset and configuration command to the new device.

The application sets the port reset bit (USBHS\_HPRT.PRST) in the host port control and status register to 1, maintaining this state for a minimum of 10ms and a maximum of 20ms, thereby driving the USB reset signal (single-ended zero) through the USB driver. The application calculates the duration of this process, then clears the port reset bit.

After the USB reset sequence is completed, the port enable/disable change bit (USBHS\_HPRT.PENCHNG) triggers a host port interrupt immediately, notifying the application that the enumerated device speed can be read from the port speed field (PSPD in USBHS\_HPRT) in the host port control and status register, and that the host has started driving SOF (HS/FS) or Keep-alive tokens (LS). At this point the host is ready to enumerate the device by sending commands to the device.

## Host Suspend

The application suspends the USB bus by setting the port suspend bit (PSUSP in USBHS\_HPRT) in the host port control and status register to 1. The USBHS module stops sending SOF and enters suspend status.

The bus can exit suspend status through autonomous activity (remote wake-up) from a remote device. In this scenario, the remote wake-up signal will trigger a remote wake-up interrupt (USBHS\_GINTSTS.WKUPINT). The hardware automatically resets the port resume bit (USBHS\_HPRT.PRES) in the host port control and status register and drives the resume signal via USB autonomously. The application must time the resume window, then clear the port resume bit to exit the suspend status and restart SOF.

If the host initiates exit from suspend status, the application must set the port resume bit to 1 to activate the resume signaling on the host port, time the resume window, and finally clear the port resume bit.

## Host Channel

The USBHS module implements 16 host channels. Each host channel can be used for USB host transfers (USB pipes). The host can handle up to 8 transfer requests at the same time. If the application has more than 8 transfer requests pending, the host controller driver (HCD) must reassign the channel for unprocessed transfer requests after the channel is released from the previous task (i.e., upon receiving the transmission complete and channel halt interrupts).

Each host channel can be configured to support input/output and periodic/non-periodic transactions. Each host channel uses dedicated control (HCCHARx) register, transfer configuration (HCTSIZx) register/interrupt (HCINTx) register, and its associated interrupt mask register (HCINTMSKx).

### Host Channel Control

The application can control the host channel through the host channel x characteristic register (HCCHARx) with the following settings:

- Channel enabled/disabled
- Set the target USB device speed: HS/FS/LS
- Set the target USB device address
- Set the endpoint number on the target USB device for communication with this channel
- Sets the transfer direction on this channel: IN/OUT

- Set the type of USB transfer on this channel: control/bulk/interrupt/isochronous
- Set the maximum packet length of the device endpoint for communication with this channel
- Set the frame for periodic transfer: odd frame/even frame

### Host Channel Transfer

The host channel transfer size register (HCTSIZx) allows applications to program the transmission size parameter and read the transfer status. The settings for this register must be completed before setting the channel enable bit to 1 in the host channel characteristics register. After enabling the endpoint, the packet count field immediately becomes read-only, while the USBHS module updates this field based on the current transmission status.

The following transfer parameters can be programmed:

- Transfer size in bytes
- The number of packets that make up the entire transfer size
- Initial data PID

### Host Channel Status/Interrupt

Host channel x interrupt register (HCINTx) indicates the endpoint status upon occurrence of USB and AHB related events. When the host channel interrupt bit (USBHS\_GINTSTS.HCINT) in the interrupt register is set to 1, the application must read these registers to obtain detailed information. Before reading these registers, the application must first read the host all channels interrupt (HCAINT) register to obtain the channel number of the host channel-x interrupt register. The application must clear the corresponding bit in this register to clear the corresponding bits in the HCAINT and GINTSTS registers. The USBHS\_HCINTMSKx register also provides mask bits for each interrupt source on every channel.

The host module provides the following status check and interrupt generation functions:

- Transfer complete interrupt, indicating both the application (AHB) and USB side have completed data transfer
- The channel stops due to transmission completion, USB transaction error, or a disable command issued by the application
- The associated transmit FIFO is half-empty or in full empty status (IN endpoint)
- ACK response received
- NAK response received
- STALL response received
- USB transaction error due to CRC check failure, timeout, bit stuffing error, or incorrect EOP
- Crosstalk error
- Frame overflow
- Error in toggling bit for data synchronization

## 34.3.6 Device Function

### Function Introduction

When the USBHS in device mode, VBUS is the 5V power supply pin specified by the USB protocol and is a 5V withstand voltage pin. This module continuously monitors the VBUS line level status to connect or disconnect devices. The USBHS\_VBUS port must be connected to the VBUS power supply through a resistor of at least 1kΩ.

A typical USB device mode system block architecture diagram is as follows:

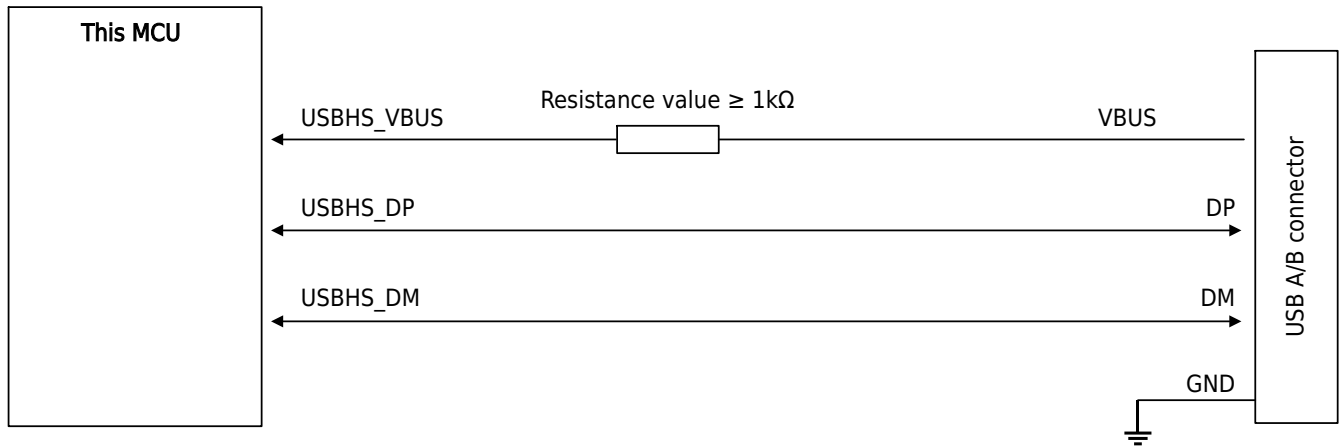


Figure 34-3 USBHS Device Mode System Architecture Diagram

### Power Supply State of Device

When the module detects that USBHS\_VBUS is at a high level, it enables the USB device to enter the power supply state. Then, USBHS automatically connects the DP pull-up resistor, signals a full-speed device connection to the host, and generates a session request interrupt (USBHS\_GINTSTS.VBUSINT), indicating entry into the powered state.

In addition, the USBHS\_VBUS input ensures the host provides a active VBUS level during USB operation. If a low level is detected on VBUS (e.g., caused by power supply interference or host port shutdown), USBHS will automatically disconnect.

Under power supply, the USBHS expects to receive a reset signal from the host. Other USB operations cannot be performed. Upon receiving the reset signal, a reset interrupt is immediately generated (USBHS\_GINTSTS.USB\_RST). After the reset signal ends, an enumeration complete interrupt (USBHS\_GINTSTS.ENUMDNE) will be generated, and USBHS will then enter the default state.

### Device Default State

By default, USBHS expects to receive a SET\_ADDRESS command from the host. Other USB operations cannot be performed. When a active SET\_ADDRESS command is decoded on USB, the application writes the corresponding address value to the device address field (USBHS\_DCFG.DAD) in the device configuration register. The USBH immediately enters the address state and is ready to respond to host transactions using the configured USB address.

### Device Suspend State

The USBHS device continuously monitors USB activity. After the USB idle time reaches 3ms, an early suspend interrupt (USBHS\_GINTSTS.ESUSP) will be issued, and after 3ms, the suspend interrupt (USBHS\_GINTSTS.USBSUSP) will confirm the device enters suspend status. Then, the device suspend bit (USBHS\_DSTS.SUSPSTS) in the devices status register is automatically set to 1, and USBHS immediately enters the suspend status.

The device can exit the suspend status by itself. In this case, the application will set the remote wake-up signal bit (USBHS\_DCTL.RWUSIG) in the device control register to 1 and clear it within 1ms to 15ms.

However, if the device detects a resume signal from the host, it will generate a resume interrupt (USBHS\_GINTSTS.WKUPINT), and the device suspend bit will be automatically cleared.

### Device Soft-off

The power supply state can be exited via software using the soft disconnect function. Setting the soft disconnect bit (USBHS\_DCTL.SDIS) in the device control register to 1 removes the DP pull-up resistor. At

this point, even though the USB cable is not physically unplugged from the host port, the host will still detect a device disconnect and trigger an interrupt.

## Device Endpoint

### Endpoint Type

The USBHS module implements the following USB endpoints:

- Control endpoint 0:
  - ▶ Bidirectional and handles only control messages
  - ▶ Use a separate set of registers to handle IN and OUT transactions
  - ▶ Dedicated control (USBHS\_DIEPCTL0/USBHS\_DOEPCTL0) register, transfer configuration (USBHS\_DIEPTSIZ0/USBHS\_DOEPSIZ0) register, and status interrupt (USBHS\_DIEPINTx/USBHS\_DOEPINT0) register. The available bit groups in the control and transmission size registers differ slightly from those in other endpoints
- 15 IN endpoints:
  - ▶ Each endpoint can be configured to support isochronous, bulk, or interrupt transfer types.
  - ▶ Each endpoint has dedicated control (USBHS\_DIEPCTLx) register, transfer configuration (USBHS\_DIEPTSIZx) register, and status interrupt (USBHS\_DIEPINTx) register
  - ▶ The device IN endpoint universal interrupt mask register (USBHS\_DIEPMSK) can be used to enable/disable the same type of interrupt sources on all IN endpoints (including EP0).
  - ▶ Supports interrupt for incomplete isochronous IN transfer (USBHS\_GINTSTS.IISOIXFR). This interrupt is triggered when at least one isochronous IN endpoint has an incomplete transfer in the current frame. This interrupt is triggered together with the periodic frame interrupt (USBHS\_GINTSTS/EOPF)
- 15 OUT endpoints:
  - ▶ Each endpoint can be configured to support isochronous, bulk, or interrupt transfer types.
  - ▶ Each endpoint has dedicated control (USBHS\_DOEPCTLx) register, transfer configuration (USBHS\_DOEPSIZx) register, and status interrupt (USBHS\_DOEPINTx) register.
  - ▶ The device OUT endpoint universal interrupt mask register (USBHS\_DOEPMSK) can be used to enable/disable the same type of interrupt sources on all OUT endpoints (including EP0).
  - ▶ Supports for incomplete isochronous OUT transfer interrupts (USBHS\_GINTSTS.INCOMPISOOUT), which will trigger when at least one transfer on an isochronous OUT endpoint is incomplete in the current frame. This interrupt is triggered together with the periodic frame interrupt (USBHS\_GINTSTS/EOPF).

### Endpoint Control

Applications can control the endpoint through the device endpoint x IN/OUT control register (DIEPCTLx/DOEPCTLx) with the following operations:

- Endpoint enable/disable
- Enable endpoint in current configuration
- Set USB transfer type (Isochronous, Bulk, and Interrupt)
- Supported packet size setting
- Set the Tx-FIFO number associated with the IN endpoint
- Configure the desired data0/data1 PID for receiving or transmitting (applicable only to bulk/interrupt transfers)
- Set the corresponding odd/even frame for receive or transmit transactions (isochronous transfer only)



- The NAK Bit can be set to reply NAK to the host request regardless of the current FIFO status
- The STALL bit can be set to make the hardware respond with STALL to all host tokens for this endpoint
- The OUT endpoint can be configured in listen mode, where CRC checking is disabled for received data

### Endpoint Transfer

The device endpoint x transmission size register (DIEPTSIZE<sub>x</sub>/DOEPTSIZE<sub>x</sub>) allows the application to program transfer size parameters and read transfer status. The setting of this register must be completed before setting the endpoint enable bit in the endpoint control register to 1. After enabling the endpoint, these fields immediately become read-only, while the USBHS block updates them based on the current transmission status.

The following transfer parameters can be programmed:

- Transfer size in bytes
- The number of packets that make up the entire transfer

### Endpoint Status/Status

Device endpoint x interrupt register (DIEPINT<sub>x</sub>/DOEPINT<sub>x</sub>) indicates the endpoint status upon occurrence of USB and AHB related events. When the OUT endpoint interrupt bit or IN endpoint interrupt bit (corresponding to the USBHS\_GINTSTS.OEPINT/IEPINT bit) in the module interrupt register is set to 1, the application must read these registers to obtain detailed information. Before an application can read these registers, it must first read the device overall endpoint interrupt (USBHS\_DAINTE) register to obtain the endpoint number of the device endpoint x interrupt register. The application must clear the corresponding bit in this register to clear the corresponding bits in the DAINTE and GINTSTS registers.

The module provides the following status check and interrupt generation functions:

- Transfer complete interrupt, indicating to the application that both the AHB and the USB side have completed the data transfer
- Setup phase completed (only for OUT endpoints of control transfer type)
- The associated transmit FIFO is half-empty or in full-empty status (IN endpoint)
- A NAK reply has been sent to the host (IN endpoint for isochronous transfers only)
- IN token received while TxFIFO is empty (only for IN endpoints of bulk and interrupt transfer types)
- Received OUT token when endpoint is not enabled
- Babble error detected
- Application closes endpoint enable
- The application sets NAK to take effect for the endpoint (only applicable to IN endpoints of isochronous transfer type)
- Received more than 3 consecutive setup packets (only applicable to control-type OUT endpoints)
- Timeout detected (only for IN endpoints of control transfer type)

### 34.3.7 SOF Pulse Pin Output Function

In both host and device modes, USBHS can monitor, track, and configure SOF frames, and also features SOF pulse output functionality. The SOF pulse is output through the USBHS\_SOF pin, with a pulse width of 16 HCLK clock periods.

#### Host SOF

In host mode, the number of PHY clocks occurring between two consecutive SOF (HS/FS) or keep-alive (LS) tokens can be programmed in the host frame interval register (HFIR), allowing the application to control the SOF frame period. An interrupt is generated at the start of frame (SOF bit in



USBHS\_GINTSTS). The current frame number and the remaining time until the next SOF can be tracked by the application in the host frame number register (HFNUM).

Using the SOFEN bit in the USBHS system control register USBHS\_SYCTLREG), a SOF pulse signal with a width of 16 HCLK system clock periods can be output from the USBHS\_SOF pin simultaneously with any issued SOF token.

In addition, the SOF pulse can also serve as an internal event to trigger operations of external modules such as DMA transfers and TIMER counting.

### Device SOF

In device mode, each time the USB receives a SOF token, it will trigger a start-of-frame interrupt (SOF bit in USBHS\_GINTSTS). The corresponding frame number can be read from the device status register (FNSOF bit in USBHS\_DSTS). Using the SOFEN bit in the USBHS system control register USBHS\_SYCTLREG, a 16 HCLK system clock period SOF pulse signal can be generated and output on the USBFS\_SOF pin for external availability.

In addition, the SOF pulse can also work as an internal event to trigger operations of external modules such as DMA transfers and TIMER counting.

The periodic frame end interrupt (GINTSTS/EOPF) is used to notify the application when 80%, 85%, 90%, or 95% of the frame interval time has elapsed, depending on the periodic frame interval field in the device configuration register (PFIVL bit in the USBHS\_DCFG). This function can be used to determine whether all isochronous communications for this frame are completed.

### 34.3.8 Power Consumption Control

When the USBHS module is not used, the HCLK and PHY clocks of the USBHS module can be stopped through the CMU module to reduce power consumption.

When using the USB module, if the device USB session has not started or the device is not connected, power reduction techniques can be employed in the suspend status.

- Stop PHY clock (STPPCLK bit in USBHS\_GCCTL)

When the stop PHY clock bit in the clock gating control register is set to 1, most 48MHz internal clock domains in the USBHS full-speed module are gated off by clock gating. Even if the application still provides a clock input, the dynamic power consumption caused by clock signal toggling in the block will be saved. Additionally, most units of the transceiver will be turned off, with only the portion responsible for detecting asynchronous recovery events or remote wake-up events remaining active.

- HCLK gating (GATEHCLK bit in USBHS\_GCCTL)

When the GATEHCLK bit in the clock gating control register is set to 1, most HCLK system clock domains inside the USBHS module are gated off by clock gating. Only the register read and write interface remains active. Even when the application still provides a clock input, the dynamic power consumption caused by clock signal toggling in the block is saved.

To save dynamic power consumption, the clock is only provided to the USB data FIFO when it is accessed by the USBHS module.

### 34.3.9 Dynamic Update of USBHS\_HFIR Register

In host mode, the USB module has the function to dynamically fine-tune the frame period, enabling synchronization between the external device and the SOF frame. If the USBHS\_HFIR register is modified within the current SOF frame, the SOF period will be adjusted accordingly in the next frame. For detailed specifications, refer to the figure below.

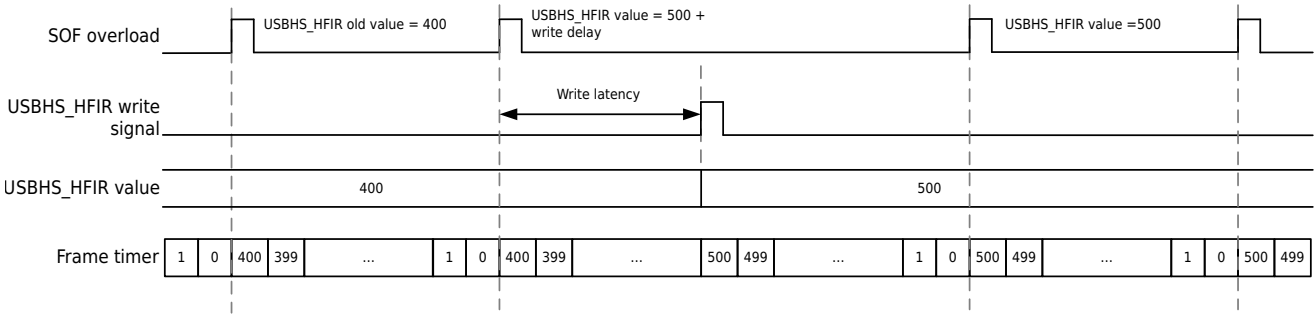


Figure 34-4 Schematic Diagram of USBHS Dynamic Update for USBHS\_HFIR Register

34.3.10 Data FIFO

The USBHS system features 8KB of dedicated RAM and employs an efficient FIFO control mechanism. In the USBHS module, the packet FIFO controller divides the RAM space into multiple TxFIFOs (Before USB transfer, the application pushes data into it for temporary storage) and a single RxFIFO (The data received from the USB is briefly stored in it before being read by the application).

The number and organization of FIFOs built in RAM depend on the device's role. In device mode, a TxFIFO is configured for each active IN endpoint. The FIFO size is software-configurable to better meet application requirements.

34.3.11 Host FIFO Architecture

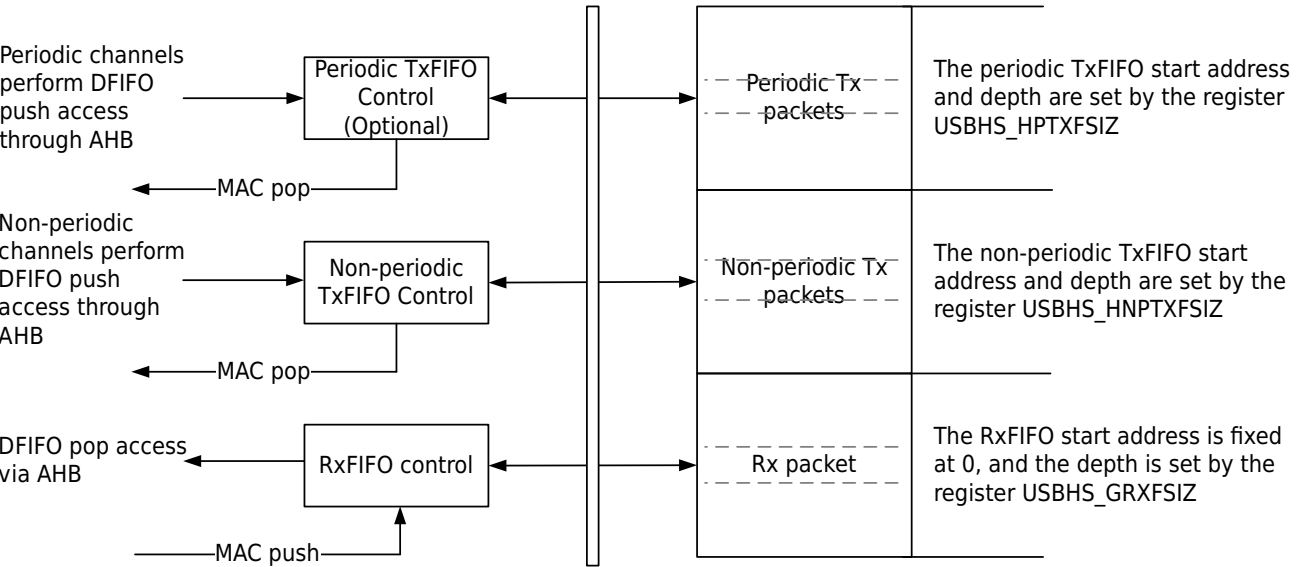


Figure 34-5 FIFO Architecture Diagram in Host Mode

Host RxFIFO

The host uses a receive FIFO to handle all periodic and non-periodic transactions. The FIFO serves as a receive buffer to store data received from USB (the data portion of received packets) until it is transferred to system memory. As long as there is space in the FIFO, data packets from the device's IN endpoint are received and stored sequentially. The status of each received data packet (including the host target channel, byte count, data PID, and checksum of the received data) is also stored in the FIFO. The size of the receive FIFO is configured in the receive FIFO size register (GRXFSIZ).

The single receive FIFO architecture enables the USB host to efficiently fill the receive data buffer:

- All IN configuration host channels share the same RAM buffer (shared FIFO)

- For any sequence of IN tokens driven by host software, the USBHS module can fill the receive FIFO up to its limit.

The application receives an RxFIFO non-empty interrupt whenever at least one data packet is available for reading in the RxFIFO. The application reads packet information from the receiving status and out-stack register and finally retrieves the data from the RxFIFO.

### Host TxFIFO

The host uses one transmit FIFO for all non-periodic (control and bulk) OUT transactions and the other transmit FIFO for all periodic (isochronous and interrupt) OUT transactions. FIFO serves as a transmit buffer to store data (transmit packets) for USB transmission. The periodic (non-periodic) TxFIFO size is configured in the host periodic (non-periodic) transmit FIFO size (HPTXFSIZ/HNPTXFSIZ) register.

The two TxFIFOs operate based on priority, with periodic communication having higher priority. Therefore, within one USB frame, periodic communication is processed first. At the start of the frame, the built-in host scheduler first processes the periodic request queue, then handles the non-periodic request queue.

The dual transmit FIFO architecture enables the USB host to separately optimize the management of periodic and non-periodic transmit data buffers:

- All host channels configured to support periodic (non-periodic) OUT transactions share the same RAM buffer (shared FIFO).
- For any sequence of OUT tokens driven by host software, the USBHS module can fill the periodic (non-periodic) transmit FIFO up to the limit.

The USBHS module issues a periodic TxFIFO empty interrupt (USBHS\_GINTSTS.PTXFE) whenever the periodic TxFIFO is half-empty or full-empty, depending on the value of the periodic Tx-FIFO empty level bit (USBHS\_GAHBCFG.PTXFELVL) in the AHB configuration register. As long as there is free space in both the periodic TxFIFO and the periodic request queue, the application can pre-write transmit data. The available space can be determined by reading the host periodic transmit FIFO and queue status register (HPTXSTS).

Whenever the non-periodic TxFIFO is half-empty or full-empty, the USBHS module generates a non-periodic TxFIFO empty interrupt (USBHS\_GINTSTS.NPTXFE), depending on the non-periodic TxFIFO empty level bit (USBHS\_GAHBCFG.TXFELVL) in the AHB configuration register. As long as there is free space in both the non-periodic TxFIFO and the non-periodic request queue, the application can write to transmit data. The available space of both can be checked by reading the host non-periodic transmit FIFO and queue status register (HNPTXSTS).

### 34.3.12 Device FIFO Architecture

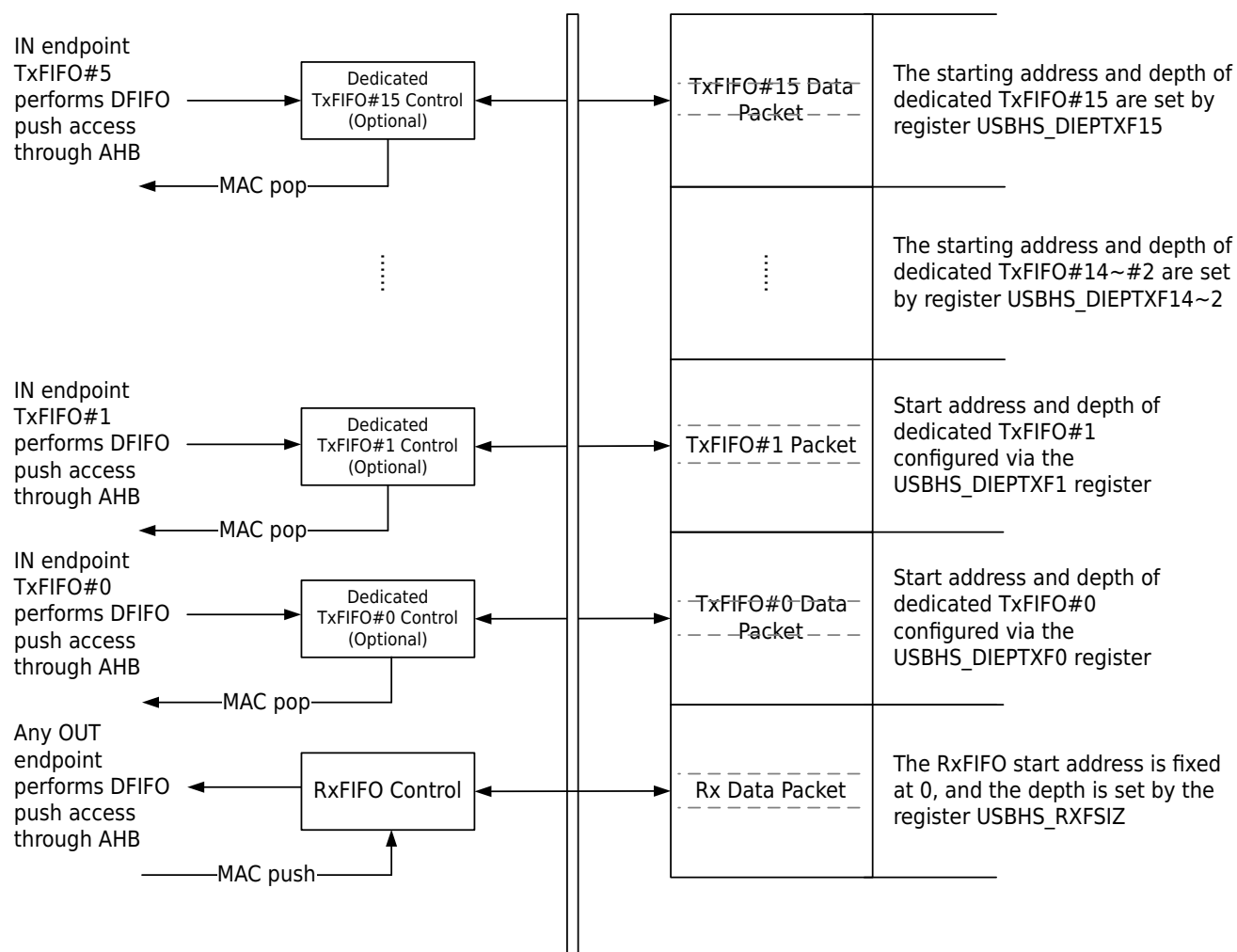


Figure 34-6 Schematic Diagram of FIFO Architecture in USBHS Device Mode

#### Device Rx FIFO

USBHS device uses a single receive FIFO to receive data sent to all OUT endpoints. As long as there is free space in the Rx FIFO, the received data packets are sequentially filled into the Rx FIFO. In addition to active data, the received data packet status (including OUT endpoint target number, byte count, data PID, and activation of the received data) is also stored by the module. When no available space exists, the device responds with a NAK to the host transaction and triggers an interrupt on the addressed endpoint. The size of the receive FIFO is configured in the receive FIFO size register (GRXFSIZ).

The single receive FIFO architecture enables USB devices to more efficiently fill the receive RAM buffer:

- All OUT endpoints share the same RAM buffer (shared FIFO)
  - For any host sequence OUT token, the USBHS module can fill the receive FIFO up to its limit
- The application will continuously receive the Rx FIFO non-empty interrupt (RXFNE bit in **USBHS\_GINTSTS**) as long as at least one data packet is available for reading in the Rx FIFO. The application reads data packet information from the receive status read and out-stack register (**GRXSTSP**), and finally retrieves the corresponding data from the receive FIFO by reading the out-stack address associated with the endpoint.

## Device Tx FIFO

The module provides dedicated FIFO for each IN endpoint. The application configures the FIFO size for IN endpoint 0 through Device Endpoint 0 Transmit FIFO Size Register (USBHS\_DIEPTXF0), and configures the FIFO size for IN endpoint x through Device IN Endpoint Transmit FIFO Size Register (USBHS\_DIEPTXFx).

### 34.3.13 FIFO RAM Allocation

#### Host Mode

##### Receive FIFO RAM Allocation

Status information is written to the FIFO with each receive data packet. Therefore, at least (maximum data packet size/4) + 2 space must be allocated for receive data packets. If multiple isochronous channels are enabled, the allocated space for receiving continuous data packets must be at least twice the (maximum data packet size/4) + 2. Generally, the recommended space is twice the (maximum data packet size/4 + 2). This allows the USB to receive subsequent data packets while the previous packet is being transferred to the CPU.

The transmission completion status information and the last received data packet of this endpoint are written together into the FIFO. Therefore, a location must be allocated for this purpose.

When running in DMA mode, the DMA address register of each host channel will be stored in the FIFO. Therefore, one location must be reserved in the FIFO for each channel to store its address register.

##### Transmit FIFO RAM Allocation

The minimum RAM required for the host non-periodic transmit FIFO is the size of the largest data packet transmitted across all supported non-periodic OUT channels.

Typically, the recommended space is twice the maximum packet size, so that while USB is transmitting the current packet, AHB can fill the next packet into the transmit FIFO.

The minimum RAM required for the host to periodically send FIFO is the size of the largest data packet transmitted across all supported periodic OUT channels. If there is at least one isochronous OUT endpoint, the space must be at least twice the maximum packet size in this channel.

When running in DMA mode, the DMA address register of each host channel will be stored in the FIFO. Therefore, one location must be reserved in the FIFO for each channel to store its address register.

#### Device Mode

##### Receive FIFO RAM Allocation

The application should allocate RAM for SETUP packets: 13 locations must be reserved in the receive FIFO to receive SETUP data packets on the control endpoint. The USBHS module will not write any other data to these bits reserved for SETUP data packets. A location will be allocated for the global OUT NAK. Status information is written to the FIFO with each receive data packet. Therefore, at least (maximum packet size/4) + 1 space must be allocated for receive data packets. If multiple isochronous endpoints are enabled, the allocated space for receiving consecutive data packets must be at least twice the (maximum packet size/4) + 1. Generally, the recommended space is twice the (Maximum packet size/4 + 1). This allows the USB to receive subsequent data packets while the previous data packet is being transferred to the CPU.

The transfer completion status information and the last received data packet of this endpoint are pushed into the FIFO together. Normally, it is recommended to allocate one bit for each OUT endpoint.

##### Transmit FIFO RAM Allocation

The minimum RAM space required for each IN endpoint's transmit FIFO is the maximum data packet size of that specific IN endpoint.

#### 34.3.14 System Performance

With a large-capacity RAM buffer, highly configurable FIFO size, 32-bit FIFO fast access via AHB push/pop registers, and especially the advanced FIFO control mechanism, optimal USB and system performance can be achieved. In fact, regardless of the current USB sequence, the USBHS can efficiently fill the available RAM space through this mechanism. With these features:

- The application has enough margin to calculate and correct the CPU load, thereby optimizing CPU bandwidth utilization:
  - ▶ The application can first accumulate a large amount of data to transmit, then send it out via USB.
  - ▶ Provides sufficient timing margin to read data from the receive FIFO.
- The USB module can keep working at full-speed, that is, to provide the maximum full-speed bandwidth (as much hardware as possible to run automatically, and as little software as possible to participate) :
  - ▶ The USB module can accumulate a large amount of sending data in advance at its disposal, enabling automatic management of USB data transmission.
  - ▶ There is ample free space in the receive buffer, which can be automatically filled with data from USB.

The USBHS module can efficiently fill the 8KB RAM buffer, and the 8KB transmit/receive data is sufficient to accommodate the data volume of a full-speed frame. Therefore, the USB system can achieve maximum USB bandwidth within a single frame without application intervention.

#### 34.3.15 Interrupt and Event

There are four types of USBHS interrupts: SOTP mode wake-up interrupt `USBHS_WKUP`, endpoint 1 OUT interrupt `USBHS_EP1_OUT`, endpoint 1 IN interrupt `USBHS_EP1_IN`, and USBHS global interrupt `USBHS_GLB`.

- **USBHS\_WKUP Interrupt**

The `USBHS_WKUP` interrupt is used in STOP mode to wake up the system's STOP mode through `USBHS_DP` or `USBHS_DM`. The interrupt enable bit is `INT_WUPEN`. `USH_WUEN`.

Before using USBHS to wake up the system STOP, you need to ensure that the USBHS controller is in the suspend status, and set the corresponding filter range and enable the filter function in the register `USBHS_SYCTLREG`.

- **USBHS\_EP1\_OUT Interrupt**

The `USBHS_EP1_OUT` interrupt is the OUT interrupt of device mode endpoint 1 that the software needs to handle. The interrupt flag can be queried in `USBHS_DEACHINT`.

- **USBHS\_EP1\_IN Interrupt**

The `USBHS_EP1_IN` interrupt is used for software to handle the IN interrupt of endpoint 1 in device mode. The interrupt flag can be queried in `USBHS_DEACHINT`.

- **USBHS\_GLB Interrupt**

The `USBHS_GLB` interrupt is the main interrupt that the software needs to handle. The flag of the global interrupt can be read in the `USBHS_GINTSTS` register.

Table 34-2 USBHS\_GLB Interrupt Event List

| Interrupt Flag     | Description                                                      | Operating Mode | Internal Event Source |
|--------------------|------------------------------------------------------------------|----------------|-----------------------|
| WKUPINT            | Resume/Remote wake-up interrupt                                  | Host or device | -                     |
| VBUSVINT           | VBUS active interrupt                                            | Device         | -                     |
| DISCINT            | Disconnect interrupt                                             | Host           | -                     |
| CIDSCHG            | Connector ID line status change interrupt                        | Host or device | -                     |
| PTXFE              | Periodic TxFIFO empty interrupt                                  | Host           | -                     |
| LPMINT             | LPM interrupt                                                    | Host or device | -                     |
| HCINT              | Host channel interrupt                                           | Host           | -                     |
| HPRTINT            | Host port interrupt                                              | Host           | -                     |
| DATAFSUSP          | Data acquisition suspend                                         | Device         | -                     |
| IPXFR/INCOMPISOOUT | Incomplete periodic transfer/Incomplete OUT isochronous transfer | Device         | -                     |
| IISOIXFR           | Incomplete IN isochronous transfer                               | Device         | -                     |
| OEPINT             | OUT endpoint interrupt                                           | Device         | -                     |
| IEPINT             | IN endpoint interrupt                                            | Device         | -                     |
| EOPF               | Periodic frame end interrupt                                     | Device         | -                     |
| ISOODRP            | Discard synchronization OUT data packet interrupt                | Device         | -                     |
| ENUMDNE            | Enumeration complete                                             | Device         | -                     |
| USBRST             | USB reset interrupt                                              | Device         | -                     |
| USBSUSP            | USB suspend interrupt                                            | Device         | -                     |
| ESUSP              | Early suspend interrupt                                          | Device         | -                     |
| GONAKEFF           | Global OUT NAK active interrupt                                  | Device         | -                     |
| GINAKEFF           | Global non-periodic IN NAK active interrupt                      | Device         | -                     |
| NPTXFE             | Non-periodic TxFIFO empty interrupt                              | Host           | -                     |
| RXFNE              | RxFIFO not empty interrupt                                       | Host or device | -                     |
| SOF                | Frame start interrupt                                            | Host or device | Yes                   |
| MMIS               | Mode mismatch interrupt                                          | Host or device | -                     |



## 34.4 Programming Module

### 34.4.1 Module Initialization

#### Context

The application must perform the module initialization sequence. For details on how to determine the mode, see [Mode Decision](#).

This section describes the initialization process after power-on of the USBHS controller. Whether working in host mode or device mode, the application must follow the initialization sequence. Initialize all module global registers based on the module configuration:

#### Procedure

- Step1.** Program the following fields in the USBHS\_GAHBCFG register:
  - Global interrupt mask bit GINTMSK=1
  - RxFIFO not empty (RXFNE bit in USBHS\_GINTSTS)
  - Periodic TxFIFO empty threshold
- Step2.** Program the following fields in the USBHS\_GUSBCFG register:
  - FS timeout calibration field
  - USB turnaround time field
- Step3.** The software must unmask the following bits in the USBHS\_GINTMSK register:
  - Mode mismatch interrupt mask
- Step4.** By reading the CMOD bit in USBHS\_GINTSTS, the software can determine whether the USBHS controller is operating in host mode or device mode.

### 34.4.2 Host Initialization

#### Context

To initialize the module as the host, the application must perform the following steps:

#### Procedure

- Step1.** Program USBHS\_GINTMSK.HPRTIM bit to 1 to unmask the interrupt.
- Step2.** Program the USBHS\_HCFG register to select full-speed host.
- Step3.** Program the PWPR bit in USBHS\_HPRT to 1 to supply VBUS to the USB bus.
- Step4.** Wait for the PCDET interrupt in USBHS\_HPRT. This indicates that a device is connected to the host port.
- Step5.** Program the PRST bit in USBHS\_HPRT to 1 to issue a reset signal on the USB bus.
- Step6.** Wait at least 10ms for allow the reset process to complete.
- Step7.** Program the PRST bit in USBHS\_HPRT to 0.
- Step8.** Wait for the PENCHNG interrupt in USBHS\_HPRT.
- Step9.** Read the PSPD bit in USBHS\_HPRT to obtain the enumeration speed.
- Step10.** Use the selected PHY clock and configure the HFIR register accordingly.
- Step11.** Program the FSLSPCS field in the USBFS\_HCFG register according to the device speed detected in Step 9. If the FSLSPCS is modified, a port reset must be performed.
- Step12.** Program the USBHS\_GRXFSIZ register to select the receive FIFO size.
- Step13.** Program the USBFS\_HNPTXFSIZ register to select the size and start address of the non-periodic transmit FIFO for non-periodic communication transactions.
- Step14.** Program the USBFS\_HPTXFSIZ register to select the size and start address of the periodic communication transmit FIFO for periodic communication transactions.



## Post Request

To communicate with the device, the system software must initialize and enable at least one channel.

### 34.4.3 Device Initialization

#### Context

During power on or after switching from host mode to device mode, the application must perform the following steps to initialize the module as a device.

#### Procedure

- Step1.** Program the following fields in the USBHS\_DCFG register:
  - Device speed
  - Non zero length status OUT handshake signal
- Step2.** Program the USBHS\_GINTMSK register to unmask the following interrupts:
  - USB reset
  - Enumeration complete
  - Early suspend
  - USB suspend
  - SOF
- Step3.** Wait for the VBUSVINT interrupt in USBHS\_GINTSTS, indicating the power supply state is entered.
- Step4.** Wait for the USBRST interrupt in USBHS\_GINTSTS. This indicates that a reset signal has been detected on the USB, and the reset process lasts approximately 10ms after receiving this interrupt.
- Step5.** Wait for the ENUMDNE interrupt in USBHS\_GINTSTS. This interrupt indicates the end of the reset process on USB. Upon receiving this interrupt, the application must read the USBHS\_DSTS register to determine the enumeration speed and perform the steps listed in the endpoint initialization on enumeration completion.

#### Results

At this point, the device is ready to receive SOF data packets and perform control transfers on control endpoint 0.

### 34.4.4 DMA Mode

The USB uses the AHB host interface to fetch transmit data packet data (AHB to USB) and receive data updates (USB to AHB). The AHB host interface uses programmed DMA address (HCDMAx register in host mode and DIEPDMAx/DOEPDMAx registers in device mode) to access data buffers.

### 34.4.5 Host Programming Model

#### 34.4.5.1 Channel Initialization

##### Context

The application must initialize one or more channels before communicating with the connected devices.

To initialize and enable the channel, the application must perform the following steps:

##### Procedure

- Step1.** Program the USBHS\_GINTMSK register to unmask the interrupt for the following bits:

- The non-periodic transmit FIFO for OUT transactions is empty (applicable when working at the pipeline transaction level and the programmed packet count field value is greater than 1).
- The non-periodic transmit FIFO for OUT transactions is half-empty (applicable when working at the pipeline transaction level and the programmed packet count field value is greater than 1).

**Step2.** Program the USBHS\_HAINTMSK register to enable the selected channel interrupt.

**Step3.** Program the USBHS\_HCINTMSK register to enable the interrupts related to communication transactions reflected in the host channel interrupt register.

**Step4.** Program the selected channel's USBHS\_HCTSIZx register to specify the total transmission size in bytes and the expected number of packets, including short packets. The application must program the PID field with the initial data PID (used for the first OUT transaction or expected to be obtained from the first IN transaction).

**Step5.** Program the selected channel's USBHS\_HCCHARx register to specify the endpoint features of the device, such as type, speed, direction, etc. (Only when the application is ready to send or receive data packets can the channel be enabled by setting the channel enable bit to 1.)

#### 34.4.5.2 Channel Halted

The application can disable any channel by programming the USBHS\_HCCHARx registers by setting the CHDIS and CHENA bits to 1. This will cause the USBHS host to clear any previously issued requests on this channel (if any) and generate a channel stop interrupt. The application must wait for the CHH interrupt in USBHS\_HCINTx before reassigning the channel to other communication transactions. The USBHS host will not interrupt communication transactions already initiated on the USB.

Before disabling a channel, the application must ensure that there is at least one free space in the non-periodic request queue (when disabling non-periodic channels) or the periodic request queue (when disabling periodic channels). The application can clear the request queue by programming the USBHS\_HCCHARx register to set the CHDIS bit to 1 and clear the CHENA bit to 0 when the request queue is full (before disabling the channel). The application will disable the channel when any of the following conditions occur:

- STALL, TXERR, BBERR or DTERR interrupt received in USBHS\_HCINTx of IN or OUT channel. The application must be capable of receiving other interrupts (DTERR, Nak, Data, TXERR) from the same channel before receiving the channel stop signal.
- Received a DISCINT (disconnect device) interrupt in USBHS\_GINTSTS. (The application will disable all enabled channels).
- The application aborted the transfer before it could complete normally.

When in DMA mode, the application cannot stop an integral periodic transfer by writing to the register.

#### 34.4.5.3 Ping Protocol

When the USBHS host operates in high-speed mode and the application needs to communicate with a high-speed bulk or control (data and status stage) OUT endpoint, the ping protocol must be used.

When the application receives a NAK/NYET/TXERR interrupt, the ping protocol must be applied. Upon receiving any of the above responses, the USBHS host discontinues any communication transactions on that endpoint. It discards all issued or fetched OUT requests (from the request queue) and clears the corresponding data from the transmit FIFO.

This is only applicable in device mode. In device mode, the application can send a ping token in two ways: By setting the DOPING bit in the HCTSIZx register before enabling the channel; or by writing

to the HCTSIZx register and setting the DOPING bit after the channel has already been enabled. This enables the USBHS host to write the ping request into the request queue. The application must wait for the device's response to the ping token (NAK, ACK, or TXERR interrupt) before proceeding with the communication transaction or sending another ping token. The application may only proceed with data communication transactions after receiving an ACK response from the device's OUT endpoint to the ping token. When operating in DMA mode, for bulk/control OUT transactions, the application does not need to set the DOPING bit in HCTSIZx upon receiving a NAK/NYET response. The USBHS host automatically sets bit 1 of the DOPING field in the HCTSIZx register, and then issues a ping token during bulk/control OUT transfers. The USBHS host continues to send ping tokens until an ACK is received, after which it automatically switches to the data communication transaction.

### 34.4.6 Device Programming Model

#### 34.4.6.1 Endpoint Initialization on USB Reset

##### Procedure

- Step1.** Set the NAK bit to 1 for all OUT endpoints
  - In USBHS\_DOEPCTLx, SNAK=1 (for all OUT endpoints)
- Step2.** Unmask the following interrupt bits
  - In USBHS\_DAINMSK, INEP0=1 (control 0 IN endpoint)
  - In USBHS\_DAINMSK, OUTEP0=1 (control 0 OUT endpoint)
  - In DOEPMSK, STUP=1
  - In DOEPMSK, XFRC=1
  - In DIEPMSK, XFRC=1
  - In DIEPMSK, TOC=1
- Step3.** Configure data FIFO RAM for each FIFO
  - Program the USBHS\_GRXFSIZ register to enable receiving OUT data and setup data for control transfers. This register must be at least equal to 1 maximum packet size for control endpoint 0+2 words (for control OUT packet status) + 10 words (for SETUP packet).
  - Program the USBHS\_TX0FSIZ register (depending on the selected FIFO number) to enable transmit control IN data. The register must be at least equal to 1 maximum packet size of control endpoint 0.
- Step4.** Program the following fields in the endpoint-related register to enable control OUT endpoint 0 to receive SETUP packets
  - In USBHS\_DOEPTSIZE0, STUPCNT=3 (receives up to 3 consecutive SETUP packets)

##### Results

At this point, all initialization work required for receiving SETUP packets is completed.

#### 34.4.6.2 Endpoint Initialization Upon USB Enumeration Completion

##### Procedure

- Step1.** Upon enumeration completion interrupt (ENUMDNE in USBHS\_GINTSTS), read the USBHS\_DSTS register to determine the device's enumeration speed.
- Step2.** Program the MPSIZ field in USBHS\_DIEPCTL0 to set the maximum packet size. This step configures control endpoint 0. The maximum packet size of the control endpoint depends on the enumeration speed.

## Results

At this point, the device is ready to receive SOF packets and is configured to perform control transfers on control endpoint 0.

### 34.4.6.3 Endpoint Initialization Upon Receiving the SetAddress Command

#### Context

This section describes the operations that the application must perform when receiving a SetAddress command in the SETUP packet.

#### Procedure

- Step1.** Program the USBHS\_DCFG register using the device address received in the SetAddress command.
- Step2.** Program the module to issue an IN data packet during the status phase.

### 34.4.6.4 Endpoint Initialization When a SetConfiguration/SetInterface Command is Received

#### Context

This section describes the operations that the application must perform upon receiving a SetConfiguration or SetInterface command in the SETUP packet.

#### Procedure

- Step1.** Upon receiving the SetConfiguration command, the application must program the endpoint registers to configure them using the features of active endpoints in the new configuration.
- Step2.** Upon receiving the SetInterface command, the application must program the endpoint register of the endpoint specified by the command.
- Step3.** Endpoints that were active in previous configurations or other settings become inactive in new configurations or other settings. These inactive endpoints must be disabled.
- Step4.** Use the USBHS\_DAINMSK register to enable interrupts for active endpoints and mask interrupts for inactive endpoints.
- Step5.** Set up data FIFO RAM for each FIFO.
- Step6.** After configuring all required endpoints, the application must program the module to transmit IN packets for the status stage.

## Results

At this point, the device module is ready to receive and transmit any type of data packet.

### 34.4.6.5 Endpoint Activation

#### Context

This section describes the steps required to activate a device endpoint or configure an existing device endpoint as a new type.

#### Procedure

- Step1.** In the USBHS\_DIEPCTLx register (for IN or bidirectional endpoints) or USBHS\_DOEPCTLx register (for OUT or bidirectional endpoints), program the features of the desired endpoint in the following fields.
  - Maximum packet size
  - USB active endpoint bit set to 1
  - Endpoint initial data synchronization bit (for interrupt and bulk endpoints)
  - Endpoint type
  - TxFIFO number

- Step2.** After the endpoint is enabled, the module starts decoding tokens sent to the endpoint and replies with a active handshake signal if the received token is enabled.

#### 34.4.6.6 Endpoint Disabled

##### Context

This section describes the steps required to disable an existing endpoint.

##### Procedure

- Step1.** To disable an endpoint, clear the USB active endpoint bit in the USBHS\_DIEPCTLx register (for IN or bidirectional endpoints) or the USBHS\_DOEPCTLx register (for OUT or bidirectional endpoints).
- Step2.** When an endpoint is disabled, the module ignores tokens sent to it, causing a USB timeout.

## 34.5 Operational Module

### 34.5.1 SETUP and OUT Data Transmission

This section describes the internal data flow and application operation steps during data OUT transfers and SETUP transactions.

#### Packet Reading

This section describes how to read data packets (OUT and SETUP data packets) from the receive FIFO.

1. When the RXFNE interrupt (USBHS\_GINTSTS register) is captured, the application must read the receive status pop up register (USBHS\_GRXSTSP).
2. The application can mask the RXFNE interrupt (in USBHS\_GINTSTS) by writing RXFNE=0 (in USBHS\_GINTMSK) until it reads the data packet from the receive FIFO.
3. If the byte counting of the received packet is not 0, pop up these data from the receive data FIFO and store them in memory. If the byte counting of the received packet is 0, no data will be popped from the receive data FIFO.
4. The packet status read from the receive FIFO has the following states:
  - Global OUT NAK:  
PKTSTS=Global OUT NAK, BCNT=0x000, the values of EPNUM and DPID are irrelevant.  
These data indicate that the global OUT NAK bit has taken effect.
  - SETUP packet:  
PKTSTS=SETUP, BCNT=0x008, EPNUM=Control EP number, DPID=D0. These data indicate that SETUP packets received on the specified endpoint can now be read from the receive FIFO.
  - Establishment phase completed:  
PKTSTS=Setup phase complete, BCNT=0x0, EPNUM=Control EP number, DPID value is irrelevant.  
This data indicates that the setup phase of the specified endpoint is complete and the data phase has been initiated. After this status entry pops from the receive FIFO, the module will generate a setup interrupt on the corresponding control OUT endpoint.
  - OUT packet:  
PKTSTS=DataOUT, BCNT=Size of received OUT data packet (BCNT: 0-1024), EPNUM=Endpoint number of the received data packet, DPID=Actual data PID.
  - Data transmission complete:  
PKTSTS=OUT data transmission complete, BCNT=0x0, EPNUM=OUT EP number that completed data transmission, DPID value is irrelevant.  
This data indicates the completion of OUT data transmission at the specified OUT endpoint. After this status entry pops from the receive FIFO, the module will trigger a "Transmission Complete" interrupt on the designated OUT endpoint.
5. After popping data from the receive FIFO, the RXFNE interrupt must be unmasked (USBHS\_GINTSTS).
6. Each time the application detects an RXFNE interrupt in USBHS\_GINTSTS, steps 1 through 5 will be repeated. Reading an empty receive FIFO may result in undefined module behavior.

#### SETUP Transaction

Describe the block's method for handling SETUP packets and the sequence for application processing of SETUP transactions.

#### Application Requirements:

1. To receive a SETUP packet, the control OUT endpoint STUPCNT field (USBHS\_DOEPTSIZE) must be programmed to a non-zero value. If the application programs the STUPCNT field to a non-zero value, the module will receive the SETUP packet and write it to the receive FIFO, regardless of the NAK status and the EPENA bit setting in USBHS\_DOEPTCTL. For each SETUP packet received by the control endpoint, the STUPCNT field decrements. If the STUPCNT field is not programmed to the appropriate value before receiving the SETUP packet, the module can still receive the SETUP packet and decrement the STUPCNT field. However, the application may not be able to determine the correct number of SETUP packets received during the setup phase of the control transmission.
  - In USBHS\_DOEPTSIZE, STUPCNT=3
2. The application must always allocate some additional space in the receive data FIFO to accommodate up to three consecutive SETUP packets on the control endpoint.
  - Reserved space of 10 words. The first SETUP packet requires 3 words, the "Setup Phase Complete" status double-word requires 1 word, and an additional 6 words are needed to store two extra SETUP packets.
  - Each SETUP packet requires 3 words to store 8 bytes of SETUP data and 4 bytes of SETUP status. The module will reserve these spaces in the receive FIFO.
  - This FIFO is exclusively used for storing SETUP packets and will never utilize this space for data packets.
3. The application must read 2 words of SETUP data packet from the receive FIFO.
4. The application must read and discard the "Establishment Phase Complete" status word from the receive FIFO.

#### Internal Data Flow:

1. Upon receiving a SETUP packet, the module writes the received data into the receive FIFO without checking the available space in the receive FIFO, regardless of the endpoint's NAK and STALL bit settings.
  - The module internally sets the IN NAK and OUTNAK bits of the control IN/OUT endpoints that receive the SETUP packet to 1.
2. For each SETUP packet received over USB, the module writes 3 words of data into the receive FIFO and decrements the STUPCNT field by 1.
  - The first word contains internal control information used by the module.
  - The second word contains the first 4 bytes of the SETUP command.
  - The third word contains the last 4 bytes of the SETUP command.
3. When the setup phase ends and the data IN/OUT phase starts, the block writes a status entry ("Setup Phase Complete" word) into the receive FIFO, indicating the setup phase is complete.
4. On the AHB side, the SETUP packet is read by the application.
5. When the application pops the "Setup Phase Complete" word from the receive FIFO, the module will interrupt the application using the STUP interrupt (USBHS\_DOEPTINT), indicating that it can process the received SETUP packet.
  - The module will clear the endpoint enable bit of control OUT endpoint.

#### Application Programming Sequence:

1. Program the USBHS\_DOEPTSIZE register.
  - STUPCNT=3
2. Wait for the RXFNE interrupt (USBHS\_GINTSTS) and read the data packet from the receive FIFO.
3. The STUP interrupt trigger (USBHS\_DOEPTINT) indicates the successful completion of SETUP data transmission.



- When this interrupt occurs, the application must read the USBHS\_DOEPTSIZx register to determine the number of received SETUP packets and process the last received SETUP packet.

#### Process More Than Three Consecutive SETUP Packets:

According to the USB2.0 specification, the host typically does not send more than three consecutive SETUP packets to the same endpoint during SETUP packet errors. However, the USB 2.0 specification does not limit the number of consecutive SETUP packets that the host can send to the same endpoint. When this occurs, the USBHS controller will generate an interrupt (B2BSTUP in USBHS\_DOEPINTx).

### Set Global OUT NAK to 1

#### Internal Data Flow:

1. If the application sets the global OUT NAK (SGONAK bit in USBHS\_DCTL) to 1, the module will stop writing any data other than SETUP packets to the receive FIFO. Regardless of the available space size in the receive FIFO, the device will respond with a NAK to asynchronous OUT tokens sent by the host, while directly ignoring synchronous OUT data packets.
2. The module writes the global OUT NAK to the receive FIFO. The application must reserve sufficient space for this.
3. When the application pops the global OUT NAK word from the receive FIFO, the block sets the GONAKEFF interrupt (USBHS\_GINTSTS) to 1.
4. When the application detects this interrupt, it will assume the module is in global OUT NAK mode. The application can clear this interrupt by clearing the SGONAK bit in USBHS\_DCTL.

#### Application Programming Sequence:

1. To stop receiving any type of data into the receive FIFO, the application must program the following fields to set the global OUT NAK bit to 1.
  - In USBHS\_DCTL, SGONAK=1
2. Wait for the GONAKEFF interrupt in USBHS\_GINTST. Once triggered, this interrupt indicates that the module has stopped receiving any type of data except SETUP packets.
  - If the application has set the SGONAK bit in USBHS\_DCTL to 1, the application can receive active OUT packets before the module generates a GONAKEFF interrupt (USBHS\_GINTSTS).
3. Applications can temporarily mask this interrupt by writing to the GINAKEFFM bit in the USBHS\_GINTMSK register.
  - In the USBHS\_GINTMSK register, GINAKEFFM=0
4. When the application is ready to exit the global OUT NAK mode, the SGONAK bit in USBHS\_DCTL must be cleared. This operation also clears the GONAKEFF interrupt (USBHS\_GINTSTS).
  - In CGONAK, USBHS\_DCTL=1
5. If the application has previously masked this interrupt, the interrupt must be unmasked as follows:
  - In GINTMSK, GINAKEFFM=1

### Disable the Enabled OUT Endpoint

The application must disable the enabled OUT endpoint using the following sequence.

#### Application Programming Sequence:

1. Before disabling any OUT endpoint, the application must enable the global OUT NAK mode in the module.
  - In USBHS\_DCTL, SGONAK=1
2. Wait for the GONAKEFF interrupt (USBHS\_GINTSTS)
3. Disable the OUT endpoint by programming the following fields:



- In USBHS\_DOEPTCTLx, EPDIS=1
  - In USBHS\_DOEPTCTLx, SNAK=1
4. Wait for the EPDISD interrupt (USBHS\_DOEPTINTx), which indicates that the OUT endpoint has been completely disabled. When an EPDISD interrupt is triggered, the module will also clear the following bits:
    - In USBHS\_DOEPTCTLx, EPDIS=0
    - In USBHS\_DOEPTCTLx, EPENA=0
  5. The application must clear the global OUT NAK bit to start receiving data from other OUT endpoints that are not disabled.
    - In USBHS\_DCTL, SGONAK=0

### Universal Asynchronous OUT Data Transmission

This section describes a general asynchronous OUT data transfer (control, bulk or interrupt).

#### Application Requirements:

1. Before establishing an OUT transfer, the application must allocate a buffer in memory to hold all the data to be received as part of the OUT transfer.
2. For OUT transfer, the transfer size field in the endpoint's transfer size register must be a multiple of the endpoint's maximum packet size (and word-aligned).
  - $\text{Transfer Size}[\text{EPNUM}] = n \times (\text{MPSIZ}[\text{EPNUM}] + 4 - (\text{MPSIZ}[\text{EPNUM}] \bmod 4))$
  - $\text{Packet Count}[\text{EPNUM}] = n$
  - $n > 0$
3. When an OUT endpoint interrupt occurs, the application must read the endpoint's transfer size register to calculate the amount of active data in memory. The received active data amount may be less than the programmed transfer size.
  - $\text{Active data amount in memory} = \text{Initial transfer amount set by application program} - \text{remaining transfer amount after module update}$
  - $\text{Number of received USB data packets} = \text{Initial number of packets set by application} - \text{remaining number of packets after module update}$

#### Internal Data Flow:

1. The application must set the transfer size and packet count fields in the endpoint-related register, clear the NAK bit, and enable the endpoint to receive data.
2. After the NAK bit is cleared, the module starts to receive data and writes the data into the receive FIFO (as long as there is space in the receive FIFO). For each data packet received over USB, the packet and its status are written to the receive FIFO. Each packet (either a packet with data volume reaching the maximum packet size or a short packet) written to the receive FIFO will decrement the packet count field of this endpoint by 1.
  - If the received packet has an inactive CRC, it is automatically cleared from the receive FIFO.
  - After replying with an ACK for the data packet on USB, the module discards the non-isochronous OUT packet retransmitted by the host due to undetected ACK. The application will not detect multiple consecutive OUT packets on the same endpoint with identical data PIDs. In this case, the packet count does not decrement.
  - If there is no space in the receive FIFO, isochronous or asynchronous packets will be ignored and not written into the receive FIFO. In addition, an asynchronous OUT token will receive a NAK handshake response.

- In all three scenarios mentioned above, the packet count will not decrement as no data is written to the receive FIFO.
- 3. When the packet count becomes 0 or a short packet is received on the endpoint, the NAK bit of that endpoint will be set to 1. When the NAK bit is set to 1, both synchronous and asynchronous data packets will be ignored and not written into the receive FIFO. Additionally, asynchronous OUT tokens will receive a NAK handshake response.
- 4. After data is written to the receive FIFO, the application will read data from the receive FIFO and write it to external storage, one packet at a time, endpoint by endpoint.
- 5. After writing each data packet to the external storage via AHB, the endpoint's transmission size is automatically decremented by the size of the data packet.
- 6. Under the following conditions, the OUT data transmission completion status of the OUT endpoint will be written to the receive FIFO:
  - Transfer size is 0 and packet count is 0
  - The last OUT data packet written to the receive FIFO is a short packet (Packet size: 0 to maximum packet size-1)
- 7. When the application pops this status entry (OUT data transfer completed) and generates a transfer completed interrupt for this endpoint, it also clears the endpoint enable bit.

#### Application Programming Sequence:

1. Program the USBHS\_DOEPTSIZE register using the transfer size and corresponding packet count.
2. Program the USBHS\_DOEPCTL register using endpoint features, and set the EPENA and CNAK bits to 1.
  - In USBHS\_DOEPCTL, EPENA=1
  - In USBHS\_DOEPCTL, CNAK=1
3. Wait for the RXFNE interrupt (in USBHS\_GINTSTS) and read the data packet from the receive FIFO.
  - This step can be repeated multiple times, depending on the transfer size.
4. Trigger the XFRC interrupt (USBHS\_DOEPINT) to indicate the successful completion of an asynchronous OUT data transmission.
5. Read the USBHS\_DOEPTSIZE register to determine the active data amount.

#### Universal Synchronous OUT Data Transmission

This section describes regular synchronous OUT data transmission.

#### Application Requirements:

1. All application requirements for asynchronous OUT data transmissions apply equally to synchronous OUT data transmission
2. For synchronous OUT data transmissions, the transmission size and packet count fields must always be set to the number of packets corresponding to the maximum packet size receivable in a single frame. Synchronous OUT data transmission transactions must be completed within a single frame.
3. Before the periodic frame ends (EOPF interrupt in USBHS\_GINTSTS), the application must read all synchronous OUT packets (data entries and status entries) from the receive FIFO.
4. To receive data in the next frame, a synchronous OUT endpoint must be enabled after EOPF (USBHS\_GINTSTS) and before SOF (USBHS\_GINTSTS).

#### Internal Data Flow:

1. The internal data flow of synchronous OUT endpoints is essentially the same as that of asynchronous OUT endpoints, with minor differences.

2. When enabling a synchronous OUT endpoint by setting the endpoint enable bit to 1 and clearing the NAK bit, the corresponding even/odd frame bit must also be set to 1. The module will receive data from a specific frame on the synchronized OUT endpoint only when the following conditions are met:
  - EONUM (in USBHS\_DOEPTCTLx)=FNSOF[0] (in USBHS\_DSTS)
3. When the application reads a complete synchronous OUT packet (data and status) from the receive FIFO, the module updates the RXDPID field in USBHS\_DOEPTSIz based on the data PID of the last synchronous OUT packet read from the receive FIFO.

#### Application Programming Sequence:

1. Program the USBHS\_DOEPTSIz register using the transmission size and corresponding packet count.
2. Program the USBHS\_DOEPTCTLx register using endpoint features, and set the endpoint enable bit, clear NAK bit, and odd/even frame bit to 1.
  - EPENA1
  - CNAK=1
  - EONUM=(0: Even/1: Odd)
3. Wait for the RXFNE interrupt (in USBHS\_GINTSTS) and read the data packet from the receive FIFO.
  - This step can be repeated multiple times, depending on the transfer size.
4. XFRC interrupt (in USBHS\_DOEPTINTx) indicates the completion of a synchronous OUT data transmission. This interrupt does not necessarily indicate that the data in memory is enabled.
5. For synchronous OUT transfers, the application may not always detect this interrupt. Instead, the application may detect the IISOXFRM interrupt in USBHS\_GINTSTS.
6. Read the USBHS\_DOEPTSIz register to determine the received transmission size and verify the activity of the received data within the frame. The application shall consider the received data in memory as active only when one of the following conditions is met:
  - RXDPID=D0 (in USBHS\_DOEPTSIz) and the number of active USB data packets received=1
  - RXDPID=D1 (in USBHS\_DOEPTSIz) and the number of active USB data packets received=2
  - RXDPID=D2 (in USBHS\_DOEPTSIz) and the number of active USB data packets received=3

The number of USB packets receiving active data=Initial packet count programmed by the application-remaining packet count after module update.

The application can discard inactive packets.

#### Incomplete Synchronous OUT Data Transmission

This section describes the application programming sequence when packet loss occurs in synchronous OUT data packets.

##### Internal Data Flow:

1. For synchronous OUT endpoints, the XFRC interrupt (in USBHS\_DOEPTINTx) may not always be triggered. If the module discards the synchronized OUT data packet, the application may fail to detect the XFRC interrupt (USBHS\_DOEPTINTx) under the following conditions:
  - When the receive FIFO cannot accommodate a complete ISO OUT packet, the module will discard the received ISO OUT data.
  - The received synchronization OUT packet has a CRC error.
  - The module received a corrupted synchronization OUT token.
  - The application reads data from the receive FIFO at a very slow speed.
2. If the block detects a periodic frame end before the transfer completion of all synchronous OUT endpoints, it will trigger an incomplete synchronous OUT data interrupt (IISOXFRM in USBHS\_GINTSTS),

indicating that at least one synchronous OUT endpoint did not trigger an XFRC interrupt (in USBHS\_DOEPINTx). At this time, endpoints with unfinished transfers remain enabled, but there are no ongoing active transfers on this USB endpoint.

#### Application Programming Sequence:

1. Hardware triggers the IISOXFRM interrupt (USBHS\_GINTSTS) indicating that at least one synchronous OUT endpoint has an incomplete transfer in the current frame.
2. If this occurs due to incomplete reading of synchronous OUT data from the endpoint, the application must ensure all synchronous OUT data (including data entries and status entries) are first read from the receive FIFO before proceeding with further processing.
  - After reading all data from the receive FIFO, the application can detect the XFRC interrupt (USBHS\_DOEPINTx). In this case, the application must re-enable the endpoint to receive synchronous OUT data in the next frame.
3. When the application receives the IISOXFRM interrupt (in USBHS\_GINTSTS), it must read the control registers (USBHS\_DOEPCTLx) of all synchronous OUT endpoints to determine which endpoints have incomplete transmission in the current frame. When both of the following conditions are met simultaneously, it indicates that the endpoint transfer is incomplete:
  - EONUM bit (in USBHS\_DOEPCTLx) = FNSOF[0] (in USBHS\_DSTS)
  - EPENA=1 (in USBHS\_DOEPCTLx)
4. Before detecting the SOF interrupt (in USBHS\_GINTSTS), the previous operation must be fully completed to ensure the current frame number has not changed.
5. For synchronous OUT endpoints with incomplete transmission, the application must discard the data in memory and disable the endpoint by setting the EPDIS bit in USBHS\_DOEPCTLx to 1.
6. Wait for the EPDIS interrupt (in USBHS\_DOEPINTx), and enable the endpoint to receive new data in the next frame.
  - Since the module may require some time to disable the endpoint, the application may fail to receive data in the next frame after detecting inactive synchronization data.

#### Disable Asynchronous OUT Endpoint

This section describes how the application can stop an asynchronous endpoint.

1. Place the module in global OUT NAK mode.
2. Disable the required endpoint
  - When disabling an endpoint, please set STALL=1 (in USBHS\_DOEPCTL) instead of setting the SNAK bit to 1 in USBHS\_DOEPCTL. The priority of the STALL bit is always higher than that of the NAK bit.
3. When the application no longer needs the endpoint to respond with a STALL handshake, the STALL bit (in USBHS\_DOEPCTLx) must be cleared to zero.
4. If an application receives a SetFeature.Endpoint Halt or ClearFeature.Endpoint Halt command from the host to set or clear the STALL state of an endpoint, it must set the STALL bit to 1 or clear it on the control endpoint before initiating the status phase transfer.

### 34.5.2 IN Data Transmission

#### Packet Write

This section describes how the application writes data packets to the endpoint FIFO when the dedicated transmit FIFO is enabled.

1. The application can choose between polling mode or interrupt mode.

- In polling mode, the application monitors the status of the endpoint transmit data FIFO by reading the USBHS\_DTXFSTSx register to determine whether there is sufficient space in the data FIFO.
  - In interrupt mode, the application waits for the TXFE interrupt (in USBHS\_DIEPINTx), then reads the USBHS\_DTXFSTSx register to determine if there is sufficient space in the data FIFO.
  - To write a single non-zero length packet, there must be sufficient space in the data FIFO to accommodate the entire packet.
  - To write a zero length packet, the application cannot check the FIFO space.
2. If one of the above methods is used, when the application determines there is sufficient space to write transmit data packets, the application must first perform the corresponding write operation to the endpoint control register before writing data to the data FIFO. Typically, the application must perform a read-modify-write operation on the USBHS\_DIEPCTLx register to avoid modifying other contents in the register while setting the endpoint enable bit to 1.  
If sufficient space is available, the application can write multiple data packets for the same endpoint into the transmit FIFO. For periodic IN endpoints, the application can only write multiple data packets within a single frame at a time. The application will only write all packets to be sent in the next frame after the previous frame's communication transaction has completed.

### Set IN Endpoint NAK to 1

#### Internal Data Flow:

1. When the application sets the IN NAK of a specific endpoint to 1, the module will stop data transmission on that endpoint, regardless of whether data is available in the endpoint's transmit FIFO.
2. The asynchronous endpoint receives the IN token and replies with a NAK handshake response.
  - The synchronous endpoint receives an IN token and returns a zero length data packet.
3. The module triggers the INEPNE (IN endpoint NAK active) interrupt in USBHS\_DIEPINTx in response to the SNAK bit in USBHS\_DIEPCTLx.
4. When the application detects this interrupt, it will consider the endpoint to be in IN NAK mode. The application can clear this interrupt by setting the CNAK bit in USBHS\_DIEPCTLx to 1.

#### Application Programming Sequence:

1. To stop sending any data on a specific IN endpoint, the application must set the IN NAK bit to 1. To set this bit to 1, the following fields must be programmed.
  - SNAK=1 in USBHS\_DIEPCTLx
2. Wait for the INEPNE interrupt in USBHS\_DIEPINTx to trigger. This interrupt indicates the module has stopped transmitting data on the endpoint.
3. The module can send active IN data on the endpoint when the application sets the NAK bit to 1 but the "NAK Active" interrupt has not yet been triggered.
4. The application can temporarily mask this interrupt by writing to the INEPNEM bit in DIEPMSK.
  - In DIEPMSK, INEPNEM=0
5. To exit endpoint NAK mode, the application must clear the NAK status bit (NAKSTS) in USBHS\_DIEPCTLx. This action also clears the INEPNE interrupt (in USBHS\_DIEPINTx).
  - In USBHS\_DIEPCTLx, CNAK=1
6. If the application has masked this interrupt, it must be unmasked as follows:
  - In DIEPMSK, INEPNEM=1

### Disable IN Endpoint

Use the following sequence to disable a previously enabled specific IN endpoint.

#### Application Programming Sequence:

1. The application must first stop writing data on the AHB before disabling the IN endpoint.
2. The application must set the endpoint to NAK mode.
  - SNAK=1 in USBHS\_DIEPCTLx
3. Wait for the INEPNE interrupt in USBHS\_DIEPINTx.
4. The following bits in the USBHS\_DIEPCTLx register of the endpoint that must be disabled must be set to 1.
  - EPDIS=1 in USBHS\_DIEPCTLx.
  - SNAK=1 in USBHS\_DIEPCTLx
5. The EPDISD interrupt in USBHS\_DIEPINTx indicates that the block has completely disabled the specified endpoint. At the same time the interrupt is triggered, the module also clears the following bits:
  - In USBHS\_DIEPCTLx, EPENA=0
  - In USBHS\_DIEPCTLx, EPDIS=0
6. The application must read the USBHS\_DIEPTSIZx register for periodic IN EP to calculate the amount of data transmitted on the endpoint via USB.
7. The application must clear the data in the endpoint transmit FIFO by setting the following fields in the USBHS\_GRSTCTL register to 1:
  - TXFNUM (in USBHS\_GRSTCTL)=Transmit FIFO number of the endpoint
  - TXFFLSH (in USBHS\_GRSTCTL)=1

The application must poll the USBFS\_GRSTCTL register until the module clears the TXFFLSH bit, indicating the FIFO flush operation has ended. To send new data on this endpoint, the application can re-enable the endpoint later.

### Universal Non-periodic IN Data Transmission

#### Application Requirements:

1. Before establishing an IN transmission, the application must ensure that each data packet comprising an IN transmission, can fit into a single buffer.
2. For IN transmission,s, the transmission size field in the endpoint transmission size register indicates the active data amount of the current transmission, which consists of multiple maximum packet sizes and a single short packet. The short data packet is sent at the end of transmission.
  - To send multiple packets of the maximum packet size and append a short packet at the end of transmission.  
 Transmission size [EPNUM]= $x \times \text{MPSIZ}[\text{EPNUM}] + \text{sp}$   
 If ( $\text{sp} > 0$ ), packet count[EPNUM]= $x + 1$ .  
 Otherwise, packet count [EPNUM]= $x$
  - To send a single zero length data packet:  
 Transmission size[EPNUM]=0  
 Packet count[EPNUM]=1
  - To send multiple packets of the maximum packet size and append a zero length packet at the end of transmission, the application must split the transfer into two parts.  
 The first part transmits a packet with the maximum packet size, while the second part transmits only a zero length packet.  
 First transfer: transmission size[EPNUM]= $x \times \text{MPSIZ}[\text{epnum}]$ ; Packet count= $n$ ;  
 Second transfer: transmission size[EPNUM]=0; Packet count=1;



3. After enabling an endpoint for data transmission, the module updates the transmission size register. At the IN transmission end, the application must read the transmission size register to determine how much data in the transmit FIFO has been sent out via USB.
4. The amount of data written to the transmit FIFO=Initial transmission size programmed by the application-Final transmission size after module update.
  - Amount of data sent via USB=(Initial packet count programmed by application-Final packet count after module update) ×MPSIZ[EPNUM]
  - Remaining data to be sent via USB=(Initial transmission size programmed by application-Data already sent via USB)

#### Internal Data Flow:

1. The application must set the transmission size and packet count fields in the specific endpoint's register, and enable the endpoint to transmit data.
2. The application must also write the required data to the transmit FIFO of this endpoint.
3. Each time the application writes a data packet to the transmit FIFO, the transmission size of this endpoint is automatically decremented by the size of the data packet. The application continuously fetches data from memory to write into the transmit FIFO until the transmission size of this endpoint becomes 0. After writing data to the FIFO, the "packet count in FIFO" counter increments (this is a 3-bit counter maintained internally by the module, with one counter per IN endpoint transmit FIFO. In the IN endpoint FIFO, the maximum number of data packets maintained by the module is always eight.) For zero length packets, each FIFO has an additional dedicated flag indicating no data is present in the FIFO.
4. When data is written to the transmit FIFO, the module will send out this data upon receiving an IN token. For each packet sent out and receiving an ACK handshake signal in response, the packet count of this endpoint will decrement by 1 until the packet count reaches 0. Upon timeout, the packet count does not decrement.
5. For zero length packets (indicated by the internal zero length flag), the module issues a zero length packet in response to the IN token and decrements the packet count field value.
6. If an IN token is received for an endpoint whose corresponding FIFO has no data and its packet count field is zero, the module will generate an "IN token received when TxFIFO is empty" (ITTXFE) interrupt for that endpoint (provided the NAK bit of the endpoint is not set to 1). The module returns a NAK handshake signal on this asynchronous endpoint.
7. The module internally resets the FIFO pointer to the beginning and does not generate a timeout interrupt.
8. When the transmission size is 0 and the packet count is 0, the transmission complete (XFRC) interrupt for this endpoint will be generated, and the endpoint enable will be cleared.

#### Application Programming Sequence:

1. Program the USBHS\_DIEPTSIZx register using the transmission size and corresponding packet count.
2. Program the USBHS\_DIEPCTLx register using endpoint features, and set the CNAK and EPENA (Endpoint enable) bits to 1.
3. When sending a non-zero length data packet, the application must poll the USBHS\_DTXFSTSx register (where x is the FIFO number associated with the endpoint) to determine whether there is sufficient space in the data FIFO. Before writing data, the application can also use the TXFE bit (in USBHS\_DIEPINTx).

#### Universal Periodic IN Data Transmission

This section describes typical periodic IN data transmission

**Application Requirements:**

1. The application requirements 1, 2, 3, and 4 for universal non-periodic IN data transmission also apply to periodic IN data transmission (with slight modification to requirement 2).
  - The application can only send a number of data packets with the maximum packet size or several packets with the maximum packet size, plus a short packet at the end of transmission.  
To send multiple packets of the maximum packet size followed by a short packet at the end of transmission, the following condition must be met:  
Transmission size [EPNUM] =  $x \times \text{MPSIZ}[\text{EPNUM}] + \text{sp}$   
(Where x is an integer greater than 0, and the range of sp is 0-MPSIZ[EPNUM]-1)  
If (sp > 0), packet count [EPNUM] = x + 1  
Otherwise, packet count [EPNUM] = x;  
MCNT[EPNUM] = packet count[EPNUM]
  - The application cannot send a zero length packet at the end of transmission. The application can independently send a zero length data packet.
  - To send a single zero length data packet:  
Transmission size[EPNUM] = 0  
Packet count [EPNUM] = 1  
MCNT[EPNUM] = packet count[EPNUM]
2. The application can only schedule one frame of data transmission at a time.
  - $(\text{MCNT}-1) \times \text{MPSIZ} \leq \text{XFERSIZ} \leq \text{MCNT} \times \text{MPSIZ}$
  - PKTCNT = MCNT (in USBHS\_DIEPTSIZE)
  - If  $\text{XFERSIZ} < \text{MCNT} \times \text{MPSIZ}$ , the last transmitted packet will be a short packet.
  - Note: MCNT is in USBHS\_DIEPTSIZE, MPSIZ is in USBHS\_DIEPTLx, PKTCNT is in USBHS\_DIEPTSIZE, XFERSIZ is in USBHS\_DIEPTSIZE.
3. Before receiving the IN token, the application must write the complete data to be sent in the frame into the transmit FIFO. Upon receiving an IN token, the module will perform the operation for an empty transmit FIFO even if only 1 double word of data remains to be written for the frame to be transmitted. When the transmit FIFO is empty:
  - The synchronous endpoint will respond with a zero length data packet.
  - The interrupt endpoint will respond with a NAK handshake signal.

**Internal Data Flow:**

1. The application must set the transmission size and packet count fields in the specific endpoint's register, and enable the endpoint to transmit data.
2. The application must also write the required data to the transmit FIFO associated with this endpoint.
3. Each time the application writes a data packet to the transmit FIFO, the transmission size of this endpoint is automatically decremented by the size of the data packet. The application continuously fetches data from memory to write into the transmit FIFO until the transmission size of this endpoint becomes 0.
4. When the periodic endpoint receives an IN token, the module will start transmitting data from the FIFO (if data exists in the FIFO). If there is no complete data packet in the FIFO for the frame to be transmitted, the module will generate an "IN token received when Tx FIFO is empty" interrupt for this endpoint.
  - The synchronous endpoint will respond with a zero length data packet.
  - The interrupt endpoint will respond with a NAK handshake signal.



5. The packet count for an endpoint is decremented by 1 when:
  - For synchronous endpoints, when transmitting a zero length or non-zero length data packet
  - For interrupt endpoints, decrement when sending the ACK handshake signal.
  - When both the transmission size and packet count are 0, a transmission completed interrupt will be generated for this endpoint, and the endpoint enable bit will be cleared.
6. Within the "Periodic Frame Interval" (controlled by the PFIVL bit in USBFS\_DCFG), the module generates an IISOIXFR interrupt in USBHS\_GINTSTS whenever it detects that data in any synchronous IN endpoint FIFO—which should be empty in the current frame—has not been fully transmitted.

#### Application Programming Sequence:

1. Program the USBHS\_DIEPCTLx register using endpoint features, and set the CNAK and EPENA bits to 1.
2. Write the data to be transmitted in the next frame into the transmit FIFO.
3. Hardware triggers the ITTXFE interrupt (in USBHS\_DIEPINTx) indicating the application has not written all required data to the transmit FIFO.
4. If the interrupt endpoint is already enabled before detecting the interrupt, the interrupt will be ignored. If the interrupt endpoint is not enabled, enable this endpoint so that data can be sent upon receiving the next IN token.
5. When the hardware triggers the XFRC interrupt (in USBHS\_DIEPINTx) and no ITTXFE interrupt is generated in USBHS\_DIEPINTx, it indicates the success of a synchronized IN transmission,. When reading the USBHS\_DIEPTSIZx register, if the transmission size=0 and packet count=0 are always obtained, it indicates all data has been transmitted via USB.
6. Setting the XFRC interrupt (in USBHS\_DIEPINTx) indicates the successful completion of an interrupt IN transmission, regardless of whether the ITTXFE interrupt (in USBHS\_DIEPINTx) is generated. When reading the USBHS\_DIEPTSIZx register, if the transmission size=0 and packet count=0 are always obtained, it indicates all data has been transmitted via USB.
7. In the USBHS\_GINTSTS register, if the incomplete synchronous IN transmission (IISOIXFR) interrupt is set without any of the aforementioned interrupts being generated, it indicates that the module did not receive at least one periodic IN token in the current frame.

#### Incomplete Synchronization IN Data Transmission

This section describes the operations that the application must perform for incomplete synchronous IN data transmission

##### Internal Data Flow:

1. The synchronization IN transmission is considered incomplete when any of the following conditions are met:
  - The module received a corrupted sync IN token on at least one sync IN endpoint. At this point, the application detects an incomplete synchronous IN transmission interrupt (IISOIXFR bit in USBHS\_GINTSTS).
  - The application writes data to the transmit FIFO too slowly, and receives an IN token before completing the full data write to the FIFO. At this point, the application detects the "IN token received when TxFIFO is empty" interrupt in USBHS\_DIEPINTx. The application can ignore this interrupt, as it will eventually generate an incomplete synchronous IN transmission interrupt (IISOIXFR bit in USBHS\_GINTSTS) at the end of the periodic frame. The module will respond to the received IN token by sending a zero length data packet via USB.
2. The application must stop writing data to the transmit FIFO as soon as possible.

3. The application must set the NAK bit and disable bit of the endpoint.
4. The module will disable this endpoint, clear the disable bit, and trigger the "Endpoint Disable" interrupt.

#### Application Programming Sequence:

1. The application can ignore the "IN token received when TxFIFO is empty" interrupt in USBHS\_DIE-PINTx for any isochronous IN endpoint, as this will ultimately generate an incomplete synchronous IN transmission interrupt (in USBHS\_GINTSTS).
2. Hardware-triggered incomplete synchronous IN transmission interrupt (in USBHS\_GINTSTS) indicates there is at least one incomplete synchronous IN transmission, on a synchronous IN endpoint.
3. The application must read the "Endpoint Control" register of all synchronized IN endpoints to detect endpoints with incomplete IN data transmission
4. The application must stop writing data to the "Periodic Transmit FIFO" associated with these endpoints.
5. Program the following fields in the USBHS\_DIEPCTLx register to disable the endpoint:
  - SNAK=1 in USBHS\_DIEPCTLx
  - EPDIS=1 in USBHS\_DIEPCTLx
6. Hardware triggers the "Endpoint Disable" interrupt in USBHS\_DIEPINTx, indicating the module has disabled this endpoint.
  - At this point, the application must either empty the data in the associated transmit FIFO or overwrite the existing data in the FIFO by enabling a new transmission endpoint in the next frame. To refresh the data, the application must use the USBHS\_GRSTCTL register.

#### Stop the Asynchronous IN Endpoint

This section describes how an application can stop an asynchronous endpoint.

#### Application Programming Sequence:

1. Disable the IN endpoint to be stopped. Set the STALL bit to 1 simultaneously.
2. EPDIS=1 in USBHS\_DIEPCTLx (when the endpoint is already enabled)
  - STALL=1 in USBHS\_DIEPCTLx.
  - The priority of the STALL bit is always higher than that of the NAK bit.
3. A hardware-triggered "Endpoint Disable" interrupt (in USBHS\_DIEPINTx) notifies the application that the module has disabled the specified endpoint.
4. The application must empty the non-periodic or periodic transmit FIFO based on the endpoint type. For non-periodic endpoints, the application must re-enable another non-periodic endpoint without stopping to transmit data.
5. When the application is ready to end the STALL handshake signal for this endpoint, it must clear the STALL bit in USBHS\_DIEPCTLx.
6. If the application sets or clears the STALL state of an endpoint upon receiving a SetFeature.Endpoint Halt or ClearFeature.Endpoint Halt command from the host, the STALL bit must be set to 1 or cleared before the status phase transfer of the control endpoint.

#### Special Case: Stop Controlling the OUT Endpoint

During the data phase of a control transmission, if the number of IN/OUT tokens sent by the host exceeds the value specified in the SETUP packet, the module must respond with STALL to these excess IN/OUT tokens. In this scenario, the application must enable the ITTXFE interrupt of USBHS\_DIEPINTx and the OTEPDIS interrupt of USBHS\_DOEPINTx during the data phase of the control transmission (when the module has completed transmitting the data amount specified by the SETUP packet). Subsequently,

when the application receives this interrupt, it must set the STALL bit in the corresponding endpoint control register to 1 and clear this interrupt.

## 34.6 Register Description

The application controls the USBHS module by performing read and write operations on the control and status registers through the AHB slave interface. All registers in the USBHS module are 32-bit registers with 32-bit aligned addresses, and can only be accessed in 32-bit mode.

Control and status registers are categorized as follows:

- USBHS System Control Register
- Module Global Register
- Host Mode Register
- Device Mode Register
- Power and Clock Gating Control Registers
- Data FIFO(DFIFO) Access Register

The USBHS system control register has a different base address from other registers. This register is independent of the USBHS module and controls the relevant settings of the USBHS module.

Only the module global register, power and clock gating control registers, and data FIFO access registers can be accessed in both host and device modes. When the USBHS module operates in one mode (host or device), the application must not access registers in the alternate role mode, such as accessing device mode registers while in host mode. If an illegal access occurs, a mode mismatch interrupt will be generated and reflected in the module interrupt register USBHS.GINTSTS.NMIS bit. When the module switches from one role mode to another, the registers in the new operating mode must be reprogrammed to the power on reset state.

### Control Status Register Memory Mapping

The host and device mode registers occupy different addresses. All registers are implemented within the AHB clock domain.

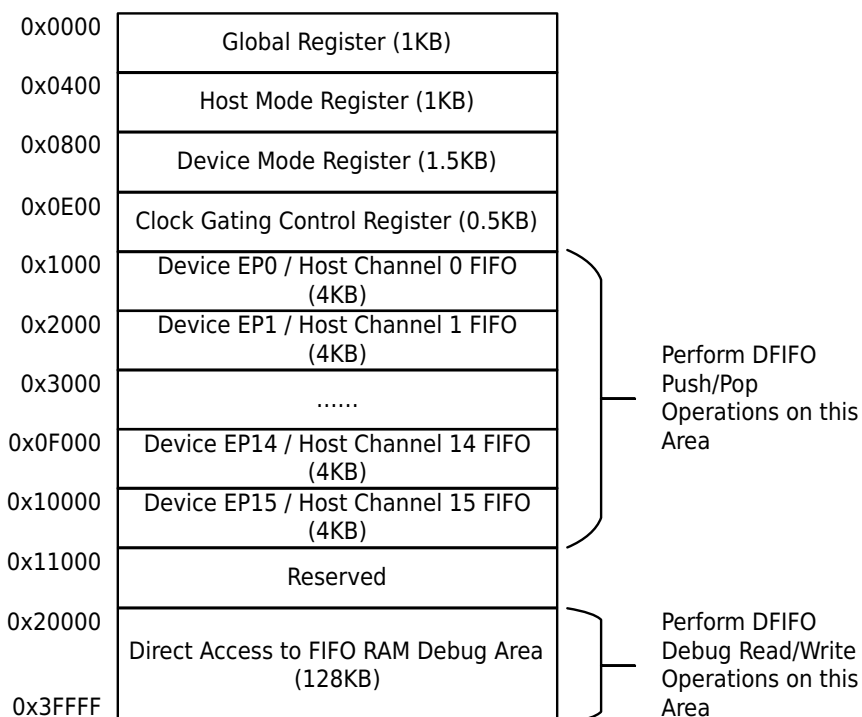


Figure 34-7 USBHS Control and Status Register Memory Mapping

### 34.6.1 Register List

USB system control register base address: 0x40055400

Table 34-3 USB System Control Register Offset

| Offset | Register     | Description                 | Width |
|--------|--------------|-----------------------------|-------|
| 0x00   | USB_SYCTLREG | USB System Control Register | 32    |

USBHS module register base address: 0x400C0000

Table 34-4 USB 2.0 High-Speed Module USBHS Register Offset

| Offset      | Register        | Description                                             | Width |
|-------------|-----------------|---------------------------------------------------------|-------|
| 0x00        | USBHS_GVBUSCFG  | VBUS Control Register                                   | 32    |
| 0x08        | USBHS_GAHBCFG   | AHB Control Register                                    | 32    |
| 0x0C        | USBHS_GUSBCFG   | USB Configuration Register                              | 32    |
| 0x10        | USBHS_GRSTCTL   | Reset Register                                          | 32    |
| 0x14        | USBHS_GINTSTS   | Global Interrupt Status Register                        | 32    |
| 0x18        | USBHS_GINTMSK   | Global Interrupt Mask Register                          | 32    |
| 0x1C        | USBHS_GRXSTSR   | Receive Status Debug Read/Status Read and Pop Register  | 32    |
| 0x20        | USBHS_GRXSTSP   |                                                         | 32    |
| 0x24        | USBHS_GRXFSIZ   | Receive FIFO Size Register                              | 32    |
| 0x28        | USBHS_HNPTXFSIZ | Host Non-periodic Transmit FIFO Size Register           | 32    |
| 0x28        | USBHS_DIEPTXF0  | Device Endpoint 0 Transmit FIFO Size Register           | 32    |
| 0x2C        | USBHS_HNPTXSTS  | Non-periodic Transmit FIFO/ Queue Status Register       | 32    |
| 0x3C        | USBHS_CID       | Module ID Register                                      | 32    |
| 0x54        | USBHS_GLPMCFG   | LPM Configuration Register                              | 32    |
| 0x100       | USBHS_HPTXFSIZ  | Host Periodic Transmit FIFO Size Register               | 32    |
| 0x100+0x4*x | USBHS_DIEPTXFx  | Device IN Endpoint Transmit FIFO Size Register (x=1-15) | 32    |
| 0x400       | USBHS_HCFG      | Host Configuration Register                             | 32    |
| 0x404       | USBHS_HFIR      | Host Frame Time Interval Register                       | 32    |
| 0x408       | USBHS_HFNUM     | Host Frame Number/Frame Remaining Time Register         | 32    |
| 0x410       | USBHS_HPTXSTS   | Host Periodic Transmit FIFO/Queue Status Register       | 32    |
| 0x414       | USBHS_HAINT     | Host All Channels Interrupt Register                    | 32    |

| Offset       | Register           | Description                                           | Width |
|--------------|--------------------|-------------------------------------------------------|-------|
| 0x418        | USBHS_HAINTMSK     | Host All Channels Interrupt Mask Register             | 32    |
| 0x440        | USBHS_HPRT         | Host Port Control and Status Register                 | 32    |
| 0x500+0x20*x | USBHS_HCCHARx      | Host Channel x Characteristic Register                | 32    |
| 0x504+0x20*x | USBHS_HCSPLTx      | Host Channel x Separation Control Register            | 32    |
| 0x508+0x20*x | USBHS_HCINTx       | Host Channel x Interrupt Register                     | 32    |
| 0x50C+0x20*x | USBHS_HCINTMSKx    | Host Channel x Interrupt Mask Register                | 32    |
| 0x510+0x20*x | USBHS_HCTSIZx      | Host Channel x Transfer Size Register                 | 32    |
| 0x514+0x20*x | USBHS_HCDMAX       | Host Channel x DMA Address Register                   | 32    |
| 0x800        | USBHS_DCFG         | Device Configuration Register                         | 32    |
| 0x804        | USBHS_DCTL         | Device Control Register                               | 32    |
| 0x808        | USBHS_DSTS         | Device Status Register                                | 32    |
| 0x810        | USBHS_DIEPMSK      | Device IN Endpoint General Interrupt Mask Register    | 32    |
| 0x814        | USBHS_DOEPMSK      | Device OUT Endpoint General Interrupt Mask Register   | 32    |
| 0x818        | USBHS_DAIN         | Device All Endpoints Interrupt Register               | 32    |
| 0x81C        | USBHS_DAINMSK      | Device All Endpoints Interrupt Mask Register          | 32    |
| 0x830        | USBHS_DTHRCTL      | Device Threshold Control Register                     | 32    |
| 0x834        | USBHS_DIEPEMPMSK   | Device IN Endpoint FIFO Empty Interrupt Mask Register | 32    |
| 0x838        | USBHS_DEACHINT     | Device Single Endpoint Interrupt Register             | 32    |
| 0x83C        | USBHS_DEACHINTMSK  | Device Single Endpoint Interrupt Mask Register        | 32    |
| 0x844        | USBHS_DIEPEACHMSK1 | Device IN Endpoint 1 Interrupt Mask Register          | 32    |
| 0x884        | USBHS_DOEPEACHMSK1 | Device OUT Endpoint 1 Interrupt Mask Register         | 32    |
| 0x900        | USBHS_DIEPCTL0     | Device IN Endpoint 0 Control Register                 | 32    |
| 0x900+0x20*x | USBHS_DIEPCTLx     | Device IN Endpoint x Control Register (x=1-15)        | 32    |
| 0x908+0x20*x | USBHS_DIEPINTx     | Device IN Endpoint x Interrupt Register               | 32    |
| 0x910        | USBHS_DIEPTSIZ0    | Device IN Endpoint 0 Transfer Size Register           | 32    |
| 0x910+0x20*x | USBHS_DIEPTSIZx    | Device IN Endpoint x Transfer Size Register (x=1-15)  | 32    |
| 0x914+0x20*x | USBHS_DIEPDMAx     | Device IN Endpoint x DMA Address Register             | 32    |

| Offset       | Register       | Description                                          | Width |
|--------------|----------------|------------------------------------------------------|-------|
| 0x918+0x20*x | USBHS_DTXFSTSx | Device IN Endpoint x Transmit FIFO Status Register   | 32    |
| 0xB00        | USBHS_DOEPCTL0 | Device OUT Endpoint 0 Control Register               | 32    |
| 0xB00+0x20*x | USBHS_DOEPCTLx | Device OUT Endpoint x Control Register (x=1-15)      | 32    |
| 0xB08+0x20*x | USBHS_DOEPINTx | Device OUT Endpoint x Interrupt Register             | 32    |
| 0xB10        | USBHS_DOEPSIZ0 | Device OUT Endpoint 0 Transfer Size Register         | 32    |
| 0xB10+0x20*x | USBHS_DOEPSIZx | Device OUT Endpoint x Transfer Size Register(x=1-15) | 32    |
| 0xB14+0x20*x | USBHS_DOEPDMAX | Device OUT Endpoint x DMA Address Register           | 32    |
| 0xE00        | USBHS_GCCTL    | Clock Gating Control Register                        | 32    |

**Note**

In the table, x represents there are multiple registers of the same type, with values ranging from 0 to 15. Special identifiers are subject to the description in the register row.

## 34.6.2 USB System Control Register

### 34.6.2.1 USB System Control Register (USB\_SYCTLREG)

| Offset | Bit Position |    |    |    |           |    |           |    |          |    |    |    |           |    |           |    |          |    |    |    |              |    |             |   |           |   |          |   |   |   |             |   |           |  |
|--------|--------------|----|----|----|-----------|----|-----------|----|----------|----|----|----|-----------|----|-----------|----|----------|----|----|----|--------------|----|-------------|---|-----------|---|----------|---|---|---|-------------|---|-----------|--|
| 0x00   | 31           | 30 | 29 | 28 | 27        | 26 | 25        | 24 | 23       | 22 | 21 | 20 | 19        | 18 | 17        | 16 | 15       | 14 | 13 | 12 | 11           | 10 | 9           | 8 | 7         | 6 | 5        | 4 | 3 | 2 | 1           | 0 |           |  |
| Reset  | 0x00000000   |    |    |    |           |    |           |    |          |    |    |    |           |    |           |    |          |    |    |    |              |    |             |   |           |   |          |   |   |   |             |   |           |  |
| Name   | Reserved     |    |    |    | USBHS_NFE |    | USBHS_NFS |    | Reserved |    |    |    | USBFS_NFE |    | USBFS_NFS |    | Reserved |    |    |    | USBHS_FSPHYE |    | USBHS_SOFEN |   | USBHS_DFB |   | Reserved |   |   |   | USBFS_SOFEN |   | USBFS_DFB |  |
| Access |              |    |    |    | RW        |    | RW        |    |          |    |    |    | RW        |    | RW        |    |          |    |    |    | RW           |    | RW          |   | RW        |   |          |   |   |   | RW          |   | RW        |  |

| Bit   | Name      | Function                        | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | Access |
|-------|-----------|---------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:27 | Reserved  | Reserved                        | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | -      |
| 26    | USBHS_NFE | USBHS filter enable register    | Used to control the switch of the USBHS on-chip full-speed PHY DP/DM analog filter in STOP mode <ul style="list-style-type: none"> <li>0b0: Analog filter off</li> <li>0b1: Analog filter on, filtering range please refer to the setting of USBHS_NFS</li> </ul>                                                                                                                                                                                                                   | RW     |
| 25:24 | USBHS_NFS | USBHS filter selection register | Used to control the filtering range of the USBHS on-chip full-speed PHY DP/DM analog filter in STOP mode <ul style="list-style-type: none"> <li>0b00: Filter width gear 1</li> <li>0b01: Filter width gear 2</li> <li>0b10: Filter width gear 3</li> <li>0b11: Filter width gear 4</li> </ul> For specific values of each gear, please refer to the "Electrical Characteristics" chapter in the "Datasheet" under "USB On-Chip Full-Speed PHY STOP Mode Filtering Characteristics." | RW     |
| 23:19 | Reserved  | Reserved                        | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | -      |
| 18    | USBFS_NFE | USBFS filter enable register    | Used to control the switch of the USBHS on-chip full-speed PHY DP/DM analog filter in STOP mode <ul style="list-style-type: none"> <li>0b0: Analog filter off</li> <li>0b1: Analog filter on, filtering range please refer to the setting of USBFS_NFS</li> </ul>                                                                                                                                                                                                                   | RW     |



| Bit   | Name         | Function                                                     | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | Access |
|-------|--------------|--------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 17:16 | USBFS_NFS    | USBFS filter selection register                              | Used to control the filtering range of the USBHS on-chip full-speed PHY DP/DM analog filter in STOP mode <ul style="list-style-type: none"> <li>● 0b00: Filter width gear 1</li> <li>● 0b01: Filter width gear 2</li> <li>● 0b10: Filter width gear 3</li> <li>● 0b11: Filter width gear 4</li> </ul> For specific values of each gear, please refer to the "Electrical Characteristics" chapter in the "Datasheet" under "USB On-Chip Full-Speed PHY STOP Mode Filtering Characteristics." | RW     |
| 15:11 | Reserved     | Reserved                                                     | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | -      |
| 10    | USBHS_FSPHYE | USBHS on-chip full-speed PHY enable bit                      | <ul style="list-style-type: none"> <li>● 0b0: USBHS on-chip full-speed PHY suspended</li> <li>● 0b1: USBHS on-chip full-speed PHY enabled</li> </ul>                                                                                                                                                                                                                                                                                                                                        | RW     |
| 9     | USBHS_SOFEN  | USBHS SOF pulse output enable bit                            | When the USBHS host transmits an SOF or the device successfully receives an SOF, the SOF pulse with a width of 16 HCLK clock periods is output from the PAD enable <ul style="list-style-type: none"> <li>● 0b0: SOF pulse not output</li> <li>● 0b1: SOF pulse output</li> </ul>                                                                                                                                                                                                           | RW     |
| 8     | USBHS_DFB    | USBHS VBUS/ID pin internal debounce filter bypass enable bit | <ul style="list-style-type: none"> <li>● 0b0: Module internal debounce filter active</li> <li>● 0b1: Bypass module internal debounce filter</li> </ul> <div>  <b>Note</b><br/>Active in both device and host modes. </div>                                                                                                                                                                               | RW     |
| 7:2   | Reserved     | Reserved                                                     | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | -      |
| 1     | USBFS_SOFEN  | USBFS SOF pulse output enable bit                            | When the USBFS host transmits an SOF or the device successfully receives an SOF, the SOF pulse with a width of 16 HCLK clock periods is output from the PAD enable <ul style="list-style-type: none"> <li>● 0b0: SOF pulse not output</li> <li>● 0b1: SOF pulse output</li> </ul> <div>  <b>Note</b><br/>Active in both device and host modes. </div>                                                    | RW     |
| 0     | USBFS_DFB    | USBFS VBUS/ID pin internal debounce filter bypass enable bit | <ul style="list-style-type: none"> <li>● 0b0: Module internal debounce filter active</li> <li>● 0b1: Bypass module internal debounce filter</li> </ul> <div>  <b>Note</b><br/>Active in both device and host modes. </div>                                                                                                                                                                               | RW     |

### 34.6.3 USBHS Global Register

These registers are available in both host and device modes and do not need to be reprogrammed when switching between the two modes.

#### 34.6.3.1 VBUS Control Register (USBHS\_GVBUSCFG)

This register can be used to set the VBUS value to ignore the state of the VBUS pin.

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |         |   |          |   |          |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---------|---|----------|---|----------|---|---|---|
| 0x00   | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7       | 6 | 5        | 4 | 3        | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |         |   |          |   |          |   |   |   |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   | VBUSVAL |   | VBUSOVEN |   | Reserved |   |   |   |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   | RW      |   | RW       |   |          |   |   |   |

| Bit  | Name     | Function              | Description                                                                                                                                                                                                                                                                    | Access |
|------|----------|-----------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:8 | Reserved | Reserved              | -                                                                                                                                                                                                                                                                              | -      |
| 7    | VBUSVAL  | VBUS value            | Used to set the VBUS value of USBHS; when set to 1 and VBUSOVEN is set to 1, the USBHS is powered on.<br><br> <b>Note</b><br>Active only in device mode.                                    | RW     |
| 6    | VBUSOVEN | VBUS override enabled | Used to reflect the value set by VBUSVAL to the status of the USBHS CORE. The value of VBUSVAL is enable only when this bit is set to 1.<br><br> <b>Note</b><br>Active only in device mode. | RW     |
| 5:0  | Reserved | Reserved              | -                                                                                                                                                                                                                                                                              | -      |

### 34.6.3.2 AHB Control Register (USBHS\_GAHBCFG)

This register can be used to configure the module after power-on or when changing role modes, mainly containing AHB system-related configuration parameters.

The application must program this register before starting any AHB or USB transactions. Do not change this register after initial programming.

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |          |         |          |       |         |   |   |         |    |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|----------|---------|----------|-------|---------|---|---|---------|----|
| 0x08   | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8        | 7       | 6        | 5     | 4       | 3 | 2 | 1       | 0  |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |          |         |          |       |         |   |   |         |    |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   | PTXFELVL | TXFELVL | Reserved | DMAEN | HBSTLEN |   |   | GINTMSK |    |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   | RW       | RW      |          |       |         |   |   | RW      | RW |

| Bit  | Name     | Function                     | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | Access |
|------|----------|------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:9 | Reserved | Reserved                     | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | -      |
| 8    | PTXFELVL | Periodic Tx FIFO empty level | <p>Indicates when to trigger the periodic Tx FIFO empty interrupt bit (PTXFE bit in USBHS_GINTSTS) in the module interrupt register.</p> <ul style="list-style-type: none"> <li>0b0: PTXFE (in USBHS_GINTSTS) interrupt indicates the periodic Tx FIFO is in half empty status</li> <li>0b1: PTXFE (in USBHS_GINTSTS) interrupt indicates the periodic Tx FIFO is completely empty</li> </ul> <div>  <b>Note</b><br/>           Accessible only in host mode.         </div> | RW     |
| 7    | TXFELVL  | Device Tx FIFO empty level   | <p>In device mode, this bit indicates when to trigger the IN endpoint transmit FIFO empty interrupt (TXFE in USBHS_DIEPINTx).</p> <ul style="list-style-type: none"> <li>0b0: TXFE (in USBHS_DIEPINTx) interrupt indicates the IN endpoint Tx FIFO is in half empty status</li> <li>0b1: TXFE (in USBHS_DIEPINTx) interrupt indicates the IN endpoint Tx FIFO is completely empty</li> </ul> <div>  <b>Note</b><br/>           Active only in device mode.         </div>    | RW     |
| 6    | Reserved | Reserved                     | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | -      |
| 5    | DMAEN    | DMA enable                   | <ul style="list-style-type: none"> <li>0b0: The module operates in slave mode</li> <li>0b1: The module operates in DMA mode</li> </ul> <div>  <b>Note</b><br/>           Active in both device and host modes.         </div>                                                                                                                                                                                                                                                | RW     |

| Bit | Name    | Function                 | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                             | Access |
|-----|---------|--------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 4:1 | HBSTLEN | Burst length/<br>type    | <ul style="list-style-type: none"> <li>● 0b0000: Single</li> <li>● 0b0001: INCR</li> <li>● 0b0011: INCR4</li> <li>● 0b0101: INCR8</li> <li>● 0b0111: INCR16</li> <li>● Others: Setting prohibited</li> </ul> <div>  <b>Note</b><br/>Active in both device and host modes. </div>                                                                                                       | RW     |
| 0   | GINTMSK | Global interrupt<br>mask | <p>This bit is used to mask or unmask global interrupts. The interrupt status register is updated by the module, regardless of the setting of this bit.</p> <ul style="list-style-type: none"> <li>● 0b0: Mask application-triggered interrupts</li> <li>● 0b1: Unmask application-triggered interrupts</li> </ul> <div>  <b>Note</b><br/>Active in both device and host modes. </div> | RW     |

### 34.6.3.3 USB Configuration Register (USBHS\_GUSBCFG)

This register can be used to configure the module after power on or when changing role modes. It contains configuration parameters related to USB and USB-PHY.

The application must program this register before starting any AHB or USB transactions. Do not change this register after initial programming.

| Offset | Bit Position |       |       |          |    |    |        |      |      |          |           |           |         |        |       |          |         |          |      |    |    |          |   |   |        |          |   |   |       |   |   |   |
|--------|--------------|-------|-------|----------|----|----|--------|------|------|----------|-----------|-----------|---------|--------|-------|----------|---------|----------|------|----|----|----------|---|---|--------|----------|---|---|-------|---|---|---|
| 0x0C   | 31           | 30    | 29    | 28       | 27 | 26 | 25     | 24   | 23   | 22       | 21        | 20        | 19      | 18     | 17    | 16       | 15      | 14       | 13   | 12 | 11 | 10       | 9 | 8 | 7      | 6        | 5 | 4 | 3     | 2 | 1 | 0 |
| Reset  | 0x00001410   |       |       |          |    |    |        |      |      |          |           |           |         |        |       |          |         |          |      |    |    |          |   |   |        |          |   |   |       |   |   |   |
| Name   | Reserved     | FDMOD | FHMOD | Reserved |    |    | ULPIPD | PTCI | PCCI | Reserved | ULPIEBUSI | ULPIEBUSD | ULPICSM | ULPIAR | ULFSL | Reserved | PHYLPCS | Reserved | TRDT |    |    | Reserved |   |   | PHYSEL | Reserved |   |   | TOTAL |   |   |   |
| Access |              | RW    | RW    |          |    |    | RW     | RW   | RW   |          | RW        | RW        | RW      | RW     | RW    |          | RW      |          | RW   |    |    |          |   |   | RW     |          |   |   | RW    |   |   |   |

| Bit   | Name     | Function          | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | Access |
|-------|----------|-------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31    | Reserved | Reserved          | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | -      |
| 30    | FDMOD    | Force device mode | <p>Writing 1 to this bit forces the module into device mode, ignoring the state of the USBHS_ID input pin.</p> <ul style="list-style-type: none"> <li>0b0: Normal mode, depends on the input status of the USBHS_ID pin.</li> <li>0b1: Forced device mode</li> </ul> <p>After forcing the bit to 1, the application must wait at least 25ms before the change takes effect.</p> <div>  <b>Note</b><br/>           Active in both device and host modes.         </div>            | RW     |
| 29    | FHMOD    | Force host mode   | <p>When this bit is written to 1, the module is forced into host mode, ignoring the status of the USBHS_ID input pin.</p> <ul style="list-style-type: none"> <li>0b0: Normal mode, determined by the input status of the USBHS_ID pin.</li> <li>0b1: Force host mode</li> </ul> <p>After forcing the bit to 1, the application must wait at least 25ms before the change takes effect.</p> <div>  <b>Note</b><br/>           Active in both device and host modes.         </div> | RW     |
| 28:26 | Reserved | Reserved          | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | -      |

| Bit | Name            | Function                       | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | Access |
|-----|-----------------|--------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 25  | ULPIPD          | ULPI interface protect disable | <p>This bit controls the internal circuitry in the PHY to protect the ULPI interface. Any pull-up or pull-down resistors used by this function can be disabled. For more details, refer to the ULPI specification.</p> <ul style="list-style-type: none"> <li>● 0b0: Interface protection circuitry enabled</li> <li>● 0b1: Interface protection circuitry disabled</li> </ul>                                                                                                                 | RW     |
| 24  | PTCI            | Indicator pass through         | <p>This bit controls whether the complementary output must pass through the internal VBUS active level comparator function before being used for the VBUS state in the RX CMD. For more details, refer to the ULPI specification.</p> <ul style="list-style-type: none"> <li>● 0b0: The complementary output signal is enabled through the internal VBUS active level comparator</li> <li>● 0b1: The complementary output signal bypasses the internal VBUS active level comparator</li> </ul> | RW     |
| 23  | PCCI            | Indicator complement           | <p>This bit controls whether the PHY inverts the ExternalVbusIndicator input signal and generates a complementary output. For more information, refer to the ULPI specification.</p> <ul style="list-style-type: none"> <li>● 0b0: The PHY does not invert the ExternalVbusIndicator signal</li> <li>● 0b1: The PHY inverts the ExternalVbusIndicator signal</li> </ul>                                                                                                                        | RW     |
| 22  | <i>Reserved</i> | <i>Reserved</i>                | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | -      |
| 21  | ULPIVBUSI       | ULPI external VBUS indicator   | <p>This bit indicates that the ULPI PHY uses an external VBUS overcurrent indicator.</p> <ul style="list-style-type: none"> <li>● 0b0: PHY uses the internal VBUS active comparator</li> <li>● 0b1: PHY uses the external VBUS active comparator</li> </ul>                                                                                                                                                                                                                                    | RW     |
| 20  | ULPIVBUSD       | ULPI external VBUS drive       | <p>This bit selects whether the ULPI PHY uses an internal or external power source to drive 5V on the VBUS line.</p> <ul style="list-style-type: none"> <li>● 0b0: PHY drives VBUS using the internal charge pump (default)</li> <li>● 0b1: PHY drives VBUS using an external power source</li> </ul>                                                                                                                                                                                          | RW     |
| 19  | ULPICSM         | ULPI clock SuspendM            | <p>This bit maps to the ClockSuspendM bit in the ULPI PHY interface control register.</p> <ul style="list-style-type: none"> <li>● 0b0: PHY powers down the internal clock during suspend</li> <li>● 0b1: PHY does not power down the internal clock during suspend</li> </ul> <div>  <b>Note</b><br/>           This bit is enable only in serial mode.         </div>                                     | RW     |

| Bit   | Name            | Function                                                                    | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | Access |
|-------|-----------------|-----------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 18    | ULPIAR          | ULPI Auto-re-sume                                                           | This bit maps to the AutoResume bit in the ULPI PHY Interface Control register. <ul style="list-style-type: none"> <li>0b0: PHY does not support auto-resume functionality</li> <li>0b1: PHY uses auto-resume functionality</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                | RW     |
| 17    | ULFSLS          | ULPI FS/LS Select                                                           | This bit allows the application to select FS/LS serial interface mode for the ULPI PHY. This bit is only active when an FS serial transceiver is selected on the ULPI PHY. <ul style="list-style-type: none"> <li>0b0: ULPI Interface</li> <li>0b1: ULPI FS/LS Serial Interface</li> </ul>                                                                                                                                                                                                                                                                                                                                            | RW     |
| 16    | <i>Reserved</i> | <i>Reserved</i>                                                             | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | -      |
| 15    | PHYLPCS         | PHY Low-power clock select                                                  | This bit is used to select between 480 MHz or 48 MHz (low-power) PHY mode. In FS and LS modes, the PHY can typically operate using a 48 MHz clock to save power. <ul style="list-style-type: none"> <li>0b0: 480MHz Internal PLL Clock</li> <li>0b1: With the 48MHz external clock in 480MHz mode, the UTMI interface operates at either 60 MHz or 30 MHz, depending on whether 8-bit or 16-bit data width is selected. In 48 MHz mode, the UTMI interface operates at 48 MHz in FS and LS modes.</li> </ul>                                                                                                                          | RW     |
| 14    | <i>Reserved</i> | <i>Reserved</i>                                                             | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | -      |
| 13:10 | TRDT            | USB turn-around time                                                        | Set the turnaround time in units of PHY clocks. To calculate the value of TRDT, use the following formula:<br>$TRDT = 4 \times AHB \text{ clock} + 1 \text{ PHY clock}$<br>For example: <ul style="list-style-type: none"> <li>If AHB clock frequency=84MHz (PHY clock frequency 48 MHz), then TRDT is set to 9.</li> <li>If AHB clock frequency=48MHz (PHY clock frequency 48 MHz), then TRDT is set to 5.</li> </ul> <div style="border: 1px solid black; padding: 5px; margin-top: 10px;">  <b>Note</b><br/> Active only in device mode. </div> | RW     |
| 9:7   | <i>Reserved</i> | <i>Reserved</i>                                                             | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | -      |
| 6     | PHYSEL          | USB 2.0 high-speed ULPI PHY or USB 1.1 full-speed serial transceiver select | <ul style="list-style-type: none"> <li>0b0: USB 2.0 high-speed ULPI PHY</li> <li>0b1: USB on-chip full-speed serial transceiver</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | RW     |
| 5:3   | <i>Reserved</i> | <i>Reserved</i>                                                             | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | -      |

| Bit | Name  | Function               | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | Access |
|-----|-------|------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 2:0 | TOCAL | FS timeout calibration | <p>The additional latency introduced by the PHY includes the number of PHY clocks set by the application in this field, as well as the module's full-speed inter-packet timeout interval. The impact of delays introduced by different PHYs on the data line state varies. The standard USB timeout value for full-speed operation ranges from 16 to 18 bit times (inclusive). The application must program this field according to the enumeration speed. The number of bit times added per PHY clock is 0.25 bit times.</p> <div>  <b>Note</b><br/>           Active in both device and host modes.         </div> | RW     |



## 34.6.3.4 Reset Register (USBHS\_GRSTCTL)

The application resets various hardware features in the module through this register.

| Offset | Bit Position |        |          |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |        |   |   |         |         |          |       |       |       |   |   |
|--------|--------------|--------|----------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|--------|---|---|---------|---------|----------|-------|-------|-------|---|---|
| 0x10   | 31           | 30     | 29       | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10     | 9 | 8 | 7       | 6       | 5        | 4     | 3     | 2     | 1 | 0 |
| Reset  | 0x80000000   |        |          |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |        |   |   |         |         |          |       |       |       |   |   |
| Name   | AHBIDL       | DMAREQ | Reserved |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | TXFNUM |   |   | TXFFLSH | RXFFLSH | Reserved | FCRST | HSRST | CSRST |   |   |
| Access | RO           | RO     |          |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RW     |   |   | RW      | RW      |          | RW    | RW    | RW    |   |   |

| Bit   | Name     | Function           | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | Access |
|-------|----------|--------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31    | AHBIDL   | AHB master idle    | Indicates the AHB master state machine is in idle condition.<br><br> <b>Note</b><br>Active in both device and host modes.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | RO     |
| 30    | DMAREQ   | DMA request signal | This bit indicates a DMA request is in progress. For debugging purposes.<br><br> <b>Note</b><br>Active in both device and host modes.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | RO     |
| 29:11 | Reserved | Reserved           | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | -      |
| 10:6  | TXFNUM   | TxFIFO number      | FIFO number for FIFO refresh using the TxFIFO refresh bit. This field can only be modified after the module clears the TxFIFO refresh bit.<br><ul style="list-style-type: none"> <li>0b00000: Refresh non-periodic TxFIFO in host mode, refresh TxFIFO0 in device mode</li> <li>0b00001: Refresh periodic TxFIFO in host mode, refresh TxFIFO1 in device mode</li> <li>0b00010: Refresh TxFIFO2 in device mode</li> <li>0b00011: Refresh TxFIFO3 in device mode</li> <li>.....</li> <li>0b00101: Refresh TxFIFO15 in device mode</li> <li>0b10000: Refresh all transmit FIFOs in device mode or host mode</li> <li>Others: Setting prohibited</li> </ul>  <b>Note</b><br>Active in both device and host modes. | RW     |

| Bit | Name            | Function                 | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | Access |
|-----|-----------------|--------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 5   | TXFFLSH         | TxFIFO flush             | <p>This bit selectively refreshes one or all transmit FIFOs, but the operation cannot be performed when the module is handling communication transactions.</p> <p>The application can only perform a write operation to this bit after confirming that the module is not currently executing read and write operations on the TxFIFO. Use the following register for confirmation:</p> <ul style="list-style-type: none"> <li>● Read: NAK active interrupt ensures the module is not currently performing a read operation on the FIFO.</li> <li>● Write: The AHBIDL bit in USBHS_GRSTCTL ensures that the module is not currently performing any write operations to the FIFO.</li> </ul> <div>  <b>Note</b><br/> Active in both device and host modes. </div> | RW     |
| 4   | RXFFLSH         | RxFIFO flush             | <p>Applications can use this bit to refresh the entire RxFIFO, but must first ensure the module is not currently processing communication transactions. The application can only perform a write operation to this bit after confirming that the module is not currently executing read and write operations on the RxFIFO.</p> <p>The application must wait until this bit is cleared before performing other operations. Typically, an 8-clock period wait is required (based on the slowest of the PHY or AHB clock).</p> <div>  <b>Note</b><br/> Active in both device and host modes. </div>                                                                                                                                                             | RW     |
| 3   | <i>Reserved</i> | <i>Reserved</i>          | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | -      |
| 2   | FCRST           | Host frame counter reset | <p>When the application writes to this bit, the frame counter in the module is reset. After the frame counter is reset, the frame number of the next SOF sent by the module will be 0.</p> <div>  <b>Note</b><br/> Active in both device and host modes. </div>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | RW     |
| 1   | HSRST           | HCLK soft reset          | <p>The application uses this bit to refresh the control logic in the AHB clock domain. Reset only the AHB clock domain pipeline.</p> <p>FIFO is not refreshed via this bit.</p> <p>After terminating transactions on the AHB in accordance with the protocol, all state machines in the AHB clock domain are reset to the idle state. Clear the CSR control bits used by the AHB clock domain state machine.</p> <p>To clear this interrupt, the status mask bit generated by the AHB clock domain state machine</p>                                                                                                                                                                                                                                                                                                                             | RW     |

| Bit | Name  | Function        | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | Access |
|-----|-------|-----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
|     |       |                 | <p>and used to control the interrupt status must be cleared.</p> <p>Since the interrupt status bit is not cleared, the application can obtain the status of all module events that occurred after this bit was set to 1. This bit is self-clearing; the module will clear this bit after all necessary logic is reset. This process requires several clock periods, depending on the current state of the module.</p> <div>  <b>Note</b><br/>           Active in both device and host modes.         </div>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |        |
| 0   | CSRST | Core soft reset | <p>Reset the HCLK and PCLK domains as follows:<br/>Clear individual interrupt and all CSR register bits except:</p> <ul style="list-style-type: none"> <li>● GATEHCLK bit in USBHS_GCCTL</li> <li>● STPPCLK bit in USBHS_GCCTL</li> <li>● FSLSPCS bit in USBHS_HCFG</li> <li>● DSPD bit in USBHS_DCFG</li> </ul> <p>Reset all module state machines (excluding AHB slaves) to idle state and flush all transmit FIFOs and receive FIFOs.</p> <p>At the end of the last data phase in an AHB transmission, terminate all transactions on the AHB master as soon as possible. Terminate all USB transactions immediately.</p> <p>The application can perform a write operation to this bit at any time when a reset of the module is required. This bit is self-clearing. The block will clear this bit after all necessary logic is reset, which requires several clock periods depending on the current state of the module. Once this bit is cleared, software must wait for at least 3 PHY clock periods before accessing the PHY domain (synchronization delay). Additionally, the software must wait until bit 31 in this register is set to 1 (indicating the AHB master is idle) before it can start operation.</p> <p>Software reset is typically used in two scenarios: first, during software development, and second, after the user dynamically modifies the PHY selection bit in the aforementioned USB configuration register. When the user modifies the PHY, the corresponding clock will be selected for the PHY and used in the PHY domain. Once a new clock is selected, the PHY domain must be reset to ensure normal operation.</p> <div>  <b>Note</b><br/>           Active in both device and host modes.         </div> | RW     |

### 34.6.3.5 Global Interrupt Status Register (USBHS\_GINTSTS)

This register is used to interrupt the application via system-level events in the current mode (Device Mode or Host Mode). This register indicates the current mode.

The FIFO status interrupt is read-only; if software performs read and write operations on the FIFO while processing these interrupts, the FIFO interrupt flag will be automatically cleared.

Before enabling the interrupt bit, the application must clear the USBHS\_GINTSTS register during initialization to avoid any interrupts before initialization.

| Offset | Bit Position |          |         |         |        |       |       |         |          |           |               |         |      |      |          |      |         |         |        |         |       |          |          |          |        |       |     |          |      |      |    |   |
|--------|--------------|----------|---------|---------|--------|-------|-------|---------|----------|-----------|---------------|---------|------|------|----------|------|---------|---------|--------|---------|-------|----------|----------|----------|--------|-------|-----|----------|------|------|----|---|
| 0x14   | 31           | 30       | 29      | 28      | 27     | 26    | 25    | 24      | 23       | 22        | 21            | 20      | 19   | 18   | 17       | 16   | 15      | 14      | 13     | 12      | 11    | 10       | 9        | 8        | 7      | 6     | 5   | 4        | 3    | 2    | 1  | 0 |
| Reset  | 0x14000020   |          |         |         |        |       |       |         |          |           |               |         |      |      |          |      |         |         |        |         |       |          |          |          |        |       |     |          |      |      |    |   |
| Name   | WKUINT       | VBUSVINT | DISCINT | CIDSCHG | LPMINT | PTXFE | HCINT | HPRTINT | Reserved | DATAFSUSP | IPXFR/INCOMPI | ISOIXFR | OEPI | IEPI | Reserved | EOPF | ISOODRP | ENUMDNE | USBRST | USBSUSP | ESUSP | Reserved | GONAKEFF | GINAKEFF | NPTXFE | RXFNE | SOF | Reserved | MMIS | CMOD |    |   |
| Access | RW           | RW       | RW      | RW      | RW     | RO    | RO    | RO      |          | RW        | RW            | RW      | RO   | RO   |          | RW   | RW      | RW      | RW     | RW      | RW    |          |          | RO       | RO     | RO    | RO  | RW       |      | RW   | RO |   |

| Bit | Name     | Function                                            | Description                                                                                                                                                                                                                                                                                                                                                                                                                     | Access |
|-----|----------|-----------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31  | WKUINT   | Resume/<br>remote wakeup<br>detected inter-<br>rupt | <p>In device mode, this interrupt is triggered when a resume signal is detected on the USB bus.<br/>In host mode, this interrupt will be triggered when a remote wakeup is detected on the USB.<br/>Writing 1 to this bit clears it via software.</p> <div>  <b>Note</b><br/>           Active in both device and host modes.         </div> | RW     |
| 30  | VBUSVINT | VBUS valid in-<br>terrupt                           | <p>In device mode, this interrupt is triggered when a low-to-high transition is detected on the USBHS_VBUS pin.<br/>Writing 1 to this bit clears it via software.</p> <div>  <b>Note</b><br/>           Active only in device mode.         </div>                                                                                           | RW     |
| 29  | DISCINT  | Disconnect de-<br>tected interrupt                  | <p>Trigger this interrupt when a device disconnection is detected.<br/>Writing 1 to this bit clears it via software.</p> <div>  <b>Note</b><br/>           Active only in host mode.         </div>                                                                                                                                          | RW     |

| Bit | Name     | Function                         | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | Access |
|-----|----------|----------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 28  | CIDSCHG  | Connector ID status change       | <p>When the connector ID line status changes, the module sets this bit to 1.<br/>Writing 1 to this bit clears it via software.</p> <div>  <b>Note</b><br/>           Active in both device and host modes.         </div>                                                                                                                                                                                                                                                                                                                                | RW     |
| 27  | LPMINT   | LPM interrupt                    | <ul style="list-style-type: none"> <li>● Device mode: This bit is set to 1 when the device receives a correct LPM transmission and active response.</li> <li>● Host mode: This bit is set to 1 when the host receives a active response after sending LPM or when the preset retry count is exceeded.</li> </ul>                                                                                                                                                                                                                                                                                                                          | RW     |
| 26  | PTXFE    | Periodic Tx FIFO empty interrupt | <p>When the periodic transmit FIFO is half-empty or in full empty status, and there is space in the periodic request queue to write at least one entry, this interrupt will be triggered. Whether the FIFO is in a half empty state or full empty status is determined by the periodic Tx FIFO empty level bit (PTXFELVL bit in USBHS_GAHBCFG register) in the USBHS_GAHBCFG register.</p> <div>  <b>Note</b><br/>           Active only in host mode.         </div>                                                                                   | RO     |
| 25  | HCINT    | Host channels interrupt          | <p>When the module sets this bit to 1, it indicates a pending interrupt in one channel of the block (in Host Mode). The application must read the host USBHS_HAINT register to determine the exact channel number where the interrupt occurred, then read the corresponding USBHS_HCINTx register to identify the specific cause of the interrupt. The application must first clear the corresponding status bit in the USBHS_HCINTx register before clearing that bit.</p> <div>  <b>Note</b><br/>           Active only in host mode.         </div> | RO     |
| 24  | HPRTINT  | Host port interrupt              | <p>When the module sets this bit to 1, it indicates that the state of the USBHS controller port in host mode changes. The application must read USBHS_HPRT register to determine the exact event that raised this interrupt. The application must first clear the corresponding status bit of USBHS_HPRT register before clearing this bit.</p> <div>  <b>Note</b><br/>           Active only in host mode.         </div>                                                                                                                             | RO     |
| 23  | Reserved | Reserved                         | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | -      |

| Bit | Name      | Function             | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | Access |
|-----|-----------|----------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 22  | DATAFSUSP | Data fetch suspended | <p>This interrupt is only active in DMA mode. This interrupt indicates that the module has stopped fetching data for the IN endpoint due to unavailable TxFIFO space or request queue space. The application uses this interrupt in the endpoint mismatch algorithm. For example, after detecting an endpoint mismatch, the application will perform the following operations:</p> <ul style="list-style-type: none"> <li>● Set the global non-periodic IN NAK handshake signal to 1</li> <li>● Disable IN endpoint</li> <li>● Clear FIFO</li> <li>● Determine the token sequence based on the IN token sequence learning queue</li> <li>● Re-enable endpoint</li> <li>● Clear global non-periodic IN NAK handshake signal</li> </ul> <p>If the global non-periodic IN NAK has been cleared but the module has not yet retrieved data from the IN endpoint, and an IN token is received simultaneously, the module will generate an "IN token received while FIFO is empty" interrupt. Then USBHS sends a NAK response to the host. To avoid the above situation, the application can check the DATAFSUSP interrupt in USBHS_GINTSTS, which ensures the global NAK handshake signal is cleared only after the FIFO is full. Additionally, the application can mask the "Interrupt upon receiving IN token when FIFO is empty" while clearing the global IN NAK handshake signal.</p> <p>Writing 1 to this bit clears it via software.</p> <div>  <b>Note</b><br/>           Active only in device mode.         </div> | RW     |

| Bit | Name                        | Function                                                                                  | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | Access |
|-----|-----------------------------|-------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 21  | IPXFR/<br>INCOMPISO-<br>OUT | IPXFR: Incomplete periodic transfer<br>INCOMPISO-OUT: Incomplete isochronous OUT transfer | <ul style="list-style-type: none"> <li>● <b>IPXFR:</b><br/>In host mode, if there are pending periodic transactions that are scheduled to complete during the current frame, the module will set this interrupt bit to 1.<br/>Writing 1 to this bit clears it via software.</li> </ul> <div>  <b>Note</b><br/>Active only in host mode.         </div> <ul style="list-style-type: none"> <li>● <b>INCOMPISOOUT:</b><br/>In device mode, when the module sets this interrupt to 1, it indicates that at least one transfer on a synchronous OUT endpoint in the current frame remains incomplete. This interrupt is triggered along with the periodic frame end interrupt (EOPF) bit in this register.<br/>Writing 1 to this bit clears it via software.</li> </ul> <div>  <b>Note</b><br/>Active only in device mode.         </div> | RW     |
| 20  | IISOIXFR                    | Incomplete isochronous IN transfer                                                        | <p>When the module sets this interrupt to 1, it indicates that at least one transfer on a synchronous IN endpoint in the current frame remains incomplete. This interrupt is triggered along with the periodic frame end interrupt (EOPF) bit in this register.<br/>Writing 1 to this bit clears it via software.</p> <div>  <b>Note</b><br/>Active only in device mode.         </div>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | RW     |
| 19  | OEPINT                      | OUT endpoint interrupt                                                                    | <p>When the module sets this bit to 1, it indicates a pending interrupt on an OUT endpoint in the module (in device mode). The application must read the host USBHS_DAIN register to determine the exact OUT endpoint number where the interrupt occurred, then read the corresponding USBHS_DOEPINTx register to identify the specific cause of the interrupt. The application must first clear the corresponding status bit in the USBHS_DOEPINTx register before clearing this bit.</p> <div>  <b>Note</b><br/>Active only in device mode.         </div>                                                                                                                                                                                                                                                                                                                                                         | RO     |

| Bit   | Name     | Function                                 | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | Access |
|-------|----------|------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 18    | IEPINT   | IN endpoint interrupt                    | <p>When the module sets this bit to 1, it indicates a pending interrupt on an IN endpoint within the module (in device mode). The application must read the host USBHS_DAIN register to determine the exact IN endpoint number where the interrupt occurred, then read the corresponding USBHS_DIEPINTx register to identify the specific cause of the interrupt. The application must first clear the corresponding status bit in the USBHS_DIEPINTx register before clearing this bit.</p> <div>  <b>Note</b><br/>           Active only in device mode.         </div> | RO     |
| 17:16 | Reserved | Reserved                                 | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | -      |
| 15    | EOPF     | End of periodic frame interrupt          | <p>Indicates that the current frame has reached the period specified by the periodic frame interval field (PFIVL bit in USBHS_DCFG) in the USBHS_DCFG register.</p> <p>Writing 1 to this bit clears it via software.</p> <div>  <b>Note</b><br/>           Active only in device mode.         </div>                                                                                                                                                                                                                                                                     | RW     |
| 14    | ISOODRP  | Isochronous OUT packet dropped interrupt | <p>If the module cannot write the OUT data packet to the RxFIFO due to insufficient space to accommodate the maximum data packet of the OUT endpoint, the module will set this bit to 1.</p> <p>Writing 1 to this bit clears it via software.</p> <div>  <b>Note</b><br/>           Active only in device mode.         </div>                                                                                                                                                                                                                                          | RW     |
| 13    | ENUMDNE  | Enumeration done interrupt               | <p>When the module sets this bit to 1, it indicates the speed enumeration is complete. The application must read the USBHS_DSTS register to obtain the enumeration speed.</p> <p>Writing 1 to this bit clears it via software.</p> <div>  <b>Note</b><br/>           Active only in device mode.         </div>                                                                                                                                                                                                                                                         | RW     |
| 12    | USBRST   | USB reset interrupt                      | <p>When the module sets this bit to 1, it indicates a reset signal is detected on the USB.</p> <p>Writing 1 to this bit clears it via software.</p> <div>  <b>Note</b><br/>           Active only in device mode.         </div>                                                                                                                                                                                                                                                                                                                                        | RW     |



| Bit | Name            | Function                                       | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | Access |
|-----|-----------------|------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 11  | USBSUSP         | USB suspend interrupt                          | <p>When the module sets this bit to 1, it indicates that a suspend status has been detected on the USB. When the idle state on the USB bus persists for 3ms, the module enters a suspend state. Writing 1 to this bit clears it via software.</p> <div>  <b>Note</b><br/>           Active only in device mode.         </div>                                                                                                                                                                                                                                                                                                                                               | RW     |
| 10  | ESUSP           | Early suspend interrupt                        | <p>When the module sets this bit, it indicates that the USB idle state has been detected for 3ms.</p> <div>  <b>Note</b><br/>           Active only in device mode.         </div>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | RW     |
| 9:8 | <i>Reserved</i> | <i>Reserved</i>                                | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | -      |
| 7   | GONAKEFF        | Global OUT NAK effective interrupt             | <p>Indicates that the "Set global OUT NAK to 1" bit (SGONAK bit in USBHS_DCTL register) set by the application has taken effect in the module. By writing to the "Clear Global OUT NAK" bit (CGONAK bit in USBHS_DCTL register), this bit can be cleared.</p> <div>  <b>Note</b><br/>           Active only in device mode.         </div>                                                                                                                                                                                                                                                                                                                                  | RO     |
| 6   | GINAKEFF        | Global IN non-periodic NAK effective interrupt | <p>Indicates that the "Set global non-periodic IN NAK to 1" bit (SGINAK bit in USBHS_DCTL) set by the application in the USBHS_DCTL register has taken effect in the module. That is, the module has sampled the global IN NAK bit set by the application, and the result has taken effect. This bit is cleared by clearing the Clear Global Non-periodic IN NAK bit in the USBHS_DCTL register (CGINAK bit in UBSFS_DCTL). This interrupt does not necessarily indicate that a NAK handshake has been sent on the USB. The priority of the STALL bit is higher than that of the NAK bit.</p> <div>  <b>Note</b><br/>           Active only in device mode.         </div> | RO     |

| Bit | Name            | Function                            | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | Access |
|-----|-----------------|-------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 5   | NPTXFE          | Non-periodic TxFIFO empty interrupt | <p>When the non-periodic TxFIFO is half-empty or in full empty status, and there is at least space for one writable entry in the non-periodic transmission request queue, this interrupt will be triggered. Whether the FIFO is in a half empty state or full empty status is determined by the non-periodic TxFIFO empty level bit (TXFELVL bit in USBHS_GAHBCFG register) in the USBHS_GAHBCFG register.</p> <div>  <b>Note</b><br/>           Active only in host mode.         </div>                                                                                                                                                         | RO     |
| 4   | RXFNE           | RxFIFO non-empty interrupt          | <p>Indicates that at least one data packet is waiting to be read in the RxFIFO.</p> <div>  <b>Note</b><br/>           Active in both device and host modes.         </div>                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | RO     |
| 3   | SOF             | Start of frame interrupt            | <p>In host mode, when the module sets this bit to 1, it indicates that an SOF (FS) or Keep-Alive (LS) signal has been transmitted on the USB. The application must set this bit to 1 to clear the interrupt.</p> <p>In device mode, when the module sets this bit to 1, it indicates that an SOF token has been received on the USB. The application can obtain the current frame number by reading the device status register. This interrupt occurs only when the module operates in FS mode.</p> <p>Writing 1 to this bit clears it via software.</p> <div>  <b>Note</b><br/>           Active in both device and host modes.         </div> | RW     |
| 2   | <i>Reserved</i> | <i>Reserved</i>                     | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | -      |
| 1   | MMIS            | Mode mismatch interrupt             | <p>When the application attempts to access the following register, the module sets this bit to 1:</p> <ul style="list-style-type: none"> <li>● The module operates in device mode to access the host mode register.</li> <li>● The module operates in host mode to access the device mode register.</li> </ul> <p>Register access ends with an OKAY response on the AHB, but the access is internally ignored by the module and does not affect module operation.</p> <p>Writing 1 to this bit clears it via software.</p> <div>  <b>Note</b><br/>           Active in both device and host modes.         </div>                               | RW     |

| Bit | Name | Function                  | Description                                                                                                                                                                                                                                                                       | Access |
|-----|------|---------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 0   | CMOD | Current mode of operation | <p>Indicates the current mode.</p> <ul style="list-style-type: none"><li>● 0b0: Device mode</li><li>● 0b1: Host mode</li></ul> <div> <b>Note</b><br/>Active in both device and host modes.</div> | RO     |

### 34.6.3.6 Global Interrupt Mask Register (USBHS\_GINTMSK)

This register is used in conjunction with the global interrupt status register (USBHS\_GINTSTS) to interrupt the application. If an interrupt bit is masked, the corresponding interrupt will not be generated, but the associated register bit will still be set to 1.

| Offset | Bit Position |         |        |          |         |        |      |        |          |            |                      |           |       |       |          |       |          |          |         |          |        |          |           |           |         |        |      |          |       |          |   |   |
|--------|--------------|---------|--------|----------|---------|--------|------|--------|----------|------------|----------------------|-----------|-------|-------|----------|-------|----------|----------|---------|----------|--------|----------|-----------|-----------|---------|--------|------|----------|-------|----------|---|---|
| 0x18   | 31           | 30      | 29     | 28       | 27      | 26     | 25   | 24     | 23       | 22         | 21                   | 20        | 19    | 18    | 17       | 16    | 15       | 14       | 13      | 12       | 11     | 10       | 9         | 8         | 7       | 6      | 5    | 4        | 3     | 2        | 1 | 0 |
| Reset  | 0x00000000   |         |        |          |         |        |      |        |          |            |                      |           |       |       |          |       |          |          |         |          |        |          |           |           |         |        |      |          |       |          |   |   |
| Name   | WKUIM        | VBUSVIM | DISCIM | CIDSCHGM | LPMINTM | PTXFEM | HCIM | HPRTIM | Reserved | DATAFSUSPM | IPXFRM/INCOMPISOOUTM | IISOIXFRM | OEPIM | IEPIM | Reserved | EOPFM | ISOODRPM | ENUMDNEM | USBRSTM | USBSUSPM | ESUSPM | Reserved | GONAKEFFM | GINAKEFFM | NPTXFEM | RXFNEM | SOFM | Reserved | MMISM | Reserved |   |   |
| Access | RW           | RW      | RW     | RW       | RW      | RW     | RW   | RW     |          | RW         | RW                   | RW        | RW    | RW    |          | RW    | RW       | RW       | RW      | RW       | RW     |          |           | RW        | RW      | RW     | RW   | RW       |       | RW       |   |   |

| Bit | Name     | Function                                                 | Description                                                                                                                                                                                                                                                             | Access |
|-----|----------|----------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31  | WKUIM    | Resume/<br>remote wakeup<br>detected inter-<br>rupt mask | <ul style="list-style-type: none"> <li>0b0: Maskable interrupt</li> <li>0b1: Enable interrupt</li> </ul> <div>  <b>Note</b><br/>Active in both device and host modes.         </div> | RW     |
| 30  | VBUSVIM  | VBUS valid in-<br>terrupt mask                           | <ul style="list-style-type: none"> <li>0b0: Maskable interrupt</li> <li>0b1: Enable interrupt</li> </ul> <div>  <b>Note</b><br/>Active only in device mode.         </div>           | RW     |
| 29  | DISCIM   | Disconnect de-<br>tected interrupt<br>mask               | <ul style="list-style-type: none"> <li>0b0: Maskable interrupt</li> <li>0b1: Enable interrupt</li> </ul> <div>  <b>Note</b><br/>Active only in host mode.         </div>             | RW     |
| 28  | CIDSCHGM | Connector ID<br>status change<br>interrupt mask          | <ul style="list-style-type: none"> <li>0b0: Maskable interrupt</li> <li>0b1: Enable interrupt</li> </ul> <div>  <b>Note</b><br/>Active in both device and host modes.         </div> | RW     |
| 27  | LPMINTM  | LPM interrupt<br>mask                                    | <ul style="list-style-type: none"> <li>0b0: Maskable interrupt</li> <li>0b1: Enable interrupt</li> </ul> <div>  <b>Note</b><br/>Active in both device and host modes.         </div> | RW     |

| Bit | Name                      | Function                                                                                                                  | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | Access |
|-----|---------------------------|---------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 26  | PTXFEM                    | Periodic TxFIFO empty interrupt mask                                                                                      | <ul style="list-style-type: none"> <li>● 0b0: Maskable interrupt</li> <li>● 0b1: Enable interrupt</li> </ul> <div>  <b>Note</b><br/>Active only in host mode.         </div>                                                                                                                                                                                                                                                                                                                                                                                                                                  | RW     |
| 25  | HCIM                      | Host channels interrupt mask                                                                                              | <ul style="list-style-type: none"> <li>● 0b0: Maskable interrupt</li> <li>● 0b1: Enable interrupt</li> </ul> <div>  <b>Note</b><br/>Active only in host mode.         </div>                                                                                                                                                                                                                                                                                                                                                                                                                                  | RW     |
| 24  | HPRTIM                    | Host port interrupt mask                                                                                                  | <ul style="list-style-type: none"> <li>● 0b0: Maskable interrupt</li> <li>● 0b1: Enable interrupt</li> </ul> <div>  <b>Note</b><br/>Active only in host mode.         </div>                                                                                                                                                                                                                                                                                                                                                                                                                                  | RW     |
| 23  | <i>Reserved</i>           | <i>Reserved</i>                                                                                                           | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | -      |
| 22  | DATA-FSUSPM               | Data fetch suspended interrupt mask                                                                                       | <ul style="list-style-type: none"> <li>● 0b0: Maskable interrupt</li> <li>● 0b1: Enable interrupt</li> </ul> <div>  <b>Note</b><br/>Active only in device mode.         </div>                                                                                                                                                                                                                                                                                                                                                                                                                              | RW     |
| 21  | IPXFRM/<br>INCOMPI-SOOUTM | IPXFRM: Incomplete periodic transfer interrupt mask<br>INCOMPI-SOOUTM: Incomplete isochronous OUT transfer interrupt mask | <ul style="list-style-type: none"> <li>● IPXFRM:               <ul style="list-style-type: none"> <li>▶ 0b0: Maskable interrupt</li> <li>▶ 0b1: Enable interrupt</li> </ul> </li> </ul> <div>  <b>Note</b><br/>Active only in host mode.         </div> <ul style="list-style-type: none"> <li>● INCOMPISOOUTM:               <ul style="list-style-type: none"> <li>▶ 0b0: Maskable interrupt</li> <li>▶ 0b1: Enable interrupt</li> </ul> </li> </ul> <div>  <b>Note</b><br/>Active only in device mode.         </div> | RW     |
| 20  | IISOIXFRM                 | Incomplete isochronous IN transfer interrupt mask                                                                         | <ul style="list-style-type: none"> <li>● 0b0: Maskable interrupt</li> <li>● 0b1: Enable interrupt</li> </ul> <div>  <b>Note</b><br/>Active only in device mode.         </div>                                                                                                                                                                                                                                                                                                                                                                                                                              | RW     |
| 19  | OEPIM                     | OUT endpoint interrupt mask                                                                                               | <ul style="list-style-type: none"> <li>● 0b0: Maskable interrupt</li> <li>● 0b1: Enable interrupt</li> </ul> <div>  <b>Note</b><br/>Active only in device mode.         </div>                                                                                                                                                                                                                                                                                                                                                                                                                              | RW     |

| Bit   | Name      | Function                                      | Description                                                                                                                                                                                                                                                       | Access |
|-------|-----------|-----------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 18    | IEPIM     | IN endpoint interrupt mask                    | <ul style="list-style-type: none"> <li>● 0b0: Maskable interrupt</li> <li>● 0b1: Enable interrupt</li> </ul> <div>  <b>Note</b><br/>Active only in device mode.         </div>   | RW     |
| 17:16 | Reserved  | Reserved                                      | -                                                                                                                                                                                                                                                                 | -      |
| 15    | EOPFM     | End of periodic frame interrupt mask          | <ul style="list-style-type: none"> <li>● 0b0: Maskable interrupt</li> <li>● 0b1: Enable interrupt</li> </ul> <div>  <b>Note</b><br/>Active only in device mode.         </div>   | RW     |
| 14    | ISOODRPM  | Isochronous OUT packet dropped interrupt mask | <ul style="list-style-type: none"> <li>● 0b0: Maskable interrupt</li> <li>● 0b1: Enable interrupt</li> </ul> <div>  <b>Note</b><br/>Active only in device mode.         </div>   | RW     |
| 13    | ENUMDNEM  | Enumeration done interrupt mask               | <ul style="list-style-type: none"> <li>● 0b0: Maskable interrupt</li> <li>● 0b1: Enable interrupt</li> </ul> <div>  <b>Note</b><br/>Active only in device mode.         </div> | RW     |
| 12    | USBRSTM   | USB reset interrupt mask                      | <ul style="list-style-type: none"> <li>● 0b0: Maskable interrupt</li> <li>● 0b1: Enable interrupt</li> </ul> <div>  <b>Note</b><br/>Active only in device mode.         </div> | RW     |
| 11    | USBSUSPM  | USB suspend interrupt mask                    | <ul style="list-style-type: none"> <li>● 0b0: Maskable interrupt</li> <li>● 0b1: Enable interrupt</li> </ul> <div>  <b>Note</b><br/>Active only in device mode.         </div> | RW     |
| 10    | ESUSPM    | Early suspend interrupt mask                  | <ul style="list-style-type: none"> <li>● 0b0: Maskable interrupt</li> <li>● 0b1: Enable interrupt</li> </ul> <div>  <b>Note</b><br/>Active only in device mode.         </div> | RW     |
| 9:8   | Reserved  | Reserved                                      | -                                                                                                                                                                                                                                                                 | -      |
| 7     | GONAKEFFM | Global OUT NAK effective interrupt mask       | <ul style="list-style-type: none"> <li>● 0b0: Maskable interrupt</li> <li>● 0b1: Enable interrupt</li> </ul> <div>  <b>Note</b><br/>Active only in device mode.         </div> | RW     |

| Bit | Name            | Function                                            | Description                                                                                                                                                                                                                                                                 | Access |
|-----|-----------------|-----------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 6   | GINAKEFFM       | Global IN non-periodic NAK effective interrupt mask | <ul style="list-style-type: none"> <li>● 0b0: Maskable interrupt</li> <li>● 0b1: Enable interrupt</li> </ul> <div>  <b>Note</b><br/>Active only in device mode.         </div>             | RW     |
| 5   | NPTXFEM         | Non-periodic Tx FIFO empty interrupt mask           | <ul style="list-style-type: none"> <li>● 0b0: Maskable interrupt</li> <li>● 0b1: Enable interrupt</li> </ul> <div>  <b>Note</b><br/>Active only in host mode.         </div>               | RW     |
| 4   | RXFNEM          | Rx FIFO non-empty interrupt mask                    | <ul style="list-style-type: none"> <li>● 0b0: Maskable interrupt</li> <li>● 0b1: Enable interrupt</li> </ul> <div>  <b>Note</b><br/>Active in both device and host modes.         </div>   | RW     |
| 3   | SOFM            | Start of frame interrupt mask                       | <ul style="list-style-type: none"> <li>● 0b0: Maskable interrupt</li> <li>● 0b1: Enable interrupt</li> </ul> <div>  <b>Note</b><br/>Active in both device and host modes.         </div>  | RW     |
| 2   | <i>Reserved</i> | <i>Reserved</i>                                     | -                                                                                                                                                                                                                                                                           | -      |
| 1   | MMISM           | Mode mismatch interrupt mask                        | <ul style="list-style-type: none"> <li>● 0b0: Maskable interrupt</li> <li>● 0b1: Enable interrupt</li> </ul> <div>  <b>Note</b><br/>Active in both device and host modes.         </div> | RW     |
| 0   | <i>Reserved</i> | <i>Reserved</i>                                     | -                                                                                                                                                                                                                                                                           | -      |

### 34.6.3.7 Receive Status Debug Read/Status Read and Pop Register (USBHS\_GRXSTSR/USBHS\_GRXSTSP)

Reading the receive status debug read register returns the content at the top of the receive FIFO.

Reading the receive status read and pop register can additionally pop the data entry at the top of the RxFIFO. The interpretation of the receive status content differs between host mode and device mode.

When the receive FIFO is empty, the module ignores read or pop operations on this register and return the value 0x0000 0000. When the receive FIFO non-empty bit (RXFNE bit in USBHS\_GINTSTS) of the module interrupt register is set, the application must only pop the receive status FIFO.

The read offset address (Offset) : 0x1C

The pop offset address (Offset) : 0x20

#### Host Mode

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |        |    |      |    |      |    |    |    |    |    |   |   |   |   |       |   |   |   |   |   |  |  |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|--------|----|------|----|------|----|----|----|----|----|---|---|---|---|-------|---|---|---|---|---|--|--|
| -      | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19     | 18 | 17   | 16 | 15   | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5     | 4 | 3 | 2 | 1 | 0 |  |  |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |        |    |      |    |      |    |    |    |    |    |   |   |   |   |       |   |   |   |   |   |  |  |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    | PKTSTS |    | DPID |    | BCNT |    |    |    |    |    |   |   |   |   | CHNUM |   |   |   |   |   |  |  |
| Access |              |    |    |    |    |    |    |    |    |    |    |    | RO     |    | RO   |    | RO   |    |    |    |    |    |   |   |   |   | RO    |   |   |   |   |   |  |  |

| Bit   | Name     | Function       | Description                                                                                                                                                                                                                                                                                                                                           | Access |
|-------|----------|----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:21 | Reserved | Reserved       | -                                                                                                                                                                                                                                                                                                                                                     | -      |
| 20:17 | PKTSTS   | Packet status  | Indicates the status of the received data packet. <ul style="list-style-type: none"> <li>● 0b0010: IN packet received</li> <li>● 0b0011: IN transfer complete (triggers interrupt)</li> <li>● 0b0101: Data synchronization error (triggers interrupt)</li> <li>● 0b0111: Channel suspended (triggers interrupt)</li> <li>● Other: Reserved</li> </ul> | RO     |
| 16:15 | DPID     | Data PID       | Indicates the data PID of the received packet <ul style="list-style-type: none"> <li>● 0b00: DATA0</li> <li>● 0b01: DATA2</li> <li>● 0b10: DATA1</li> <li>● 0b11: MDATA</li> </ul>                                                                                                                                                                    | RO     |
| 14:4  | BCNT     | Byte count     | Indicates the number of bytes in the received IN data packet.                                                                                                                                                                                                                                                                                         | RO     |
| 3:0   | CHNUM    | Channel number | Indicates the channel number to which the currently received data packet belongs.                                                                                                                                                                                                                                                                     | RO     |



## Device Mode:

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |        |    |    |      |    |      |    |    |    |    |    |   |   |   |   |       |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|--------|----|----|------|----|------|----|----|----|----|----|---|---|---|---|-------|---|---|---|---|---|
| -      | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20     | 19 | 18 | 17   | 16 | 15   | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5     | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |        |    |    |      |    |      |    |    |    |    |    |   |   |   |   |       |   |   |   |   |   |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    | PKTSTS |    |    | DPID |    | BCNT |    |    |    |    |    |   |   |   |   | EPNUM |   |   |   |   |   |
| Access |              |    |    |    |    |    |    |    |    |    |    | RO     |    |    | RO   |    | RO   |    |    |    |    |    |   |   |   |   | RO    |   |   |   |   |   |

| Bit   | Name     | Function       | Description                                                                                                                                                                                                                                                                                                                                                                                      | Access |
|-------|----------|----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:21 | Reserved | Reserved       | -                                                                                                                                                                                                                                                                                                                                                                                                | -      |
| 20:17 | PKTSTS   | Packet status  | Indicates the status of the received data packet. <ul style="list-style-type: none"> <li>● 0b0001: Global OUT NAK (triggers interrupt).</li> <li>● 0b0010: OUT packet received</li> <li>● 0b0011: OUT transfer complete (triggers interrupt).</li> <li>● 0b0100: SETUP transaction complete (triggers interrupt).</li> <li>● 0b0110: SETUP packet received</li> <li>● Other: Reserved</li> </ul> | RO     |
| 16:15 | DPID     | Data PID       | Indicates the data PID of the received OUT data packet. <ul style="list-style-type: none"> <li>● 0b00: DATA0</li> <li>● 0b01: DATA2</li> <li>● 0b10: DATA1</li> <li>● 0b11: MDATA</li> </ul>                                                                                                                                                                                                     | RO     |
| 14:4  | BCNT     | Byte count     | Indicates the number of bytes in the received data packet.                                                                                                                                                                                                                                                                                                                                       | RO     |
| 3:0   | EPNUM    | Channel number | Indicates the endpoint number to which the currently received data packet belongs.                                                                                                                                                                                                                                                                                                               | RO     |

### 34.6.3.8 Receive FIFO Size Register (USBHS\_GRXFSIZ)

This application can program the RAM size that must be allocated to the RxFIFO.

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |      |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0x24   | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15   | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000800   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |      |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RXFD |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |      |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit   | Name     | Function     | Description                                                                                                                     | Access |
|-------|----------|--------------|---------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:12 | Reserved | Reserved     | -                                                                                                                               | -      |
| 11:0  | RXFD     | RxFIFO depth | In units of 32-bit words. The value range is from 16 to 2048.<br>The power-on reset value is the maximum FIFO depth of Rx data. | RW     |

### 34.6.3.9 Host Non-periodic Transmit FIFO Size Register (USBHS\_HNPTXFSIZ) /Device Endpoint 0 Transmit FIFO Size Register (USBHS\_DIEPTXF0)

This application can program the size of RAM that must be allocated to the TxFIFO.

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |                |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----------------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0x28   | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15             | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x08000800   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |                |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | NPTXFD/TX0FD |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | NPTXFSA/TX0FSA |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access | RW           |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RW             |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit   | Name               | Function                                                                                           | Description                                                                                                                                                                                                                                                                           | Access |
|-------|--------------------|----------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:16 | NPTXFD/<br>TX0FD   | NPTXFD: Non-periodic TxFIFO depth<br>TX0FD: End-point 0 TxFIFO depth                               | <ul style="list-style-type: none"> <li>Host mode: NPTXFD, in units of 32-bit words. The value range is from 16 to 2048.</li> <li>Device mode: TX0FD, in units of 32-bit words. The value range is from 16 to 2048.</li> </ul>                                                         | RW     |
| 15:0  | NPTXFSA/<br>TX0FSA | NPTXFSA: Non-periodic transmit RAM start address<br>TX0FSA: End-point 0 transmit RAM start address | <ul style="list-style-type: none"> <li>Host mode: NPTXFSA, this field contains the starting address of the memory of the non-periodic send FIFO RAM.</li> <li>Device mode: TX0FSA, this field contains the starting address of the memory of the endpoint 0 send FIFO RAM.</li> </ul> | RW     |

### 34.6.3.10 Non-periodic Transmit FIFO/Queue Status Register (USBHS\_HNPTXSTS)

This read-only register contains free space information for the non-periodic TxFIFO and non-periodic transmit request queue.

This register is enable only in host mode, not in device mode.

| Offset | Bit Position |          |    |    |    |    |    |    |    |          |    |    |    |    |    |    |    |          |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |
|--------|--------------|----------|----|----|----|----|----|----|----|----------|----|----|----|----|----|----|----|----------|----|----|----|----|---|---|---|---|---|---|---|---|---|---|--|
| 0x2C   | 31           | 30       | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22       | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14       | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |  |
| Reset  | 0x00080800   |          |    |    |    |    |    |    |    |          |    |    |    |    |    |    |    |          |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |
| Name   | Reserved     | NPTXQTOP |    |    |    |    |    |    |    | NPTQXSAV |    |    |    |    |    |    |    | NPTXFSAV |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |
| Access |              | RO       |    |    |    |    |    |    |    | RO       |    |    |    |    |    |    |    | RO       |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |

| Bit   | Name     | Function                                            | Description                                                                                                                                                                                                                                                                                                                                                                              | Access |
|-------|----------|-----------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31    | Reserved | Reserved                                            | -                                                                                                                                                                                                                                                                                                                                                                                        | -      |
| 30:24 | NPTXQTOP | Top of the non-periodic transmit request queue      | The entry currently being processed by the MAC in the non-periodic transmit request queue.<br>[30:27]: Channel/endpoint number<br>[26:25]:<br><ul style="list-style-type: none"> <li>0b00: IN/OUT token</li> <li>0b01: Zero-length transmit data packet</li> <li>0b10: Reserved</li> <li>0b11: Channel stop command</li> </ul> [24]: End (last entry of the selected channel/endpoint)   | RO     |
| 23:16 | NPTQXSAV | Non-periodic transmit request queue space available | Indicates the available free space size in the non-periodic transmit request queue, ranging from 0 to 8.<br>In host mode, this queue holds IN and OUT requests.<br><ul style="list-style-type: none"> <li>0: Non-periodic transmit request queue is full</li> <li>1: 1 positions available</li> <li>2: 2 positions available</li> <li>.....</li> <li>8: 8 positions available</li> </ul> | RO     |
| 15:0  | NPTXFSAV | Non-periodic TxFIFO space available                 | Indicates the available free space size in the non-periodic TxFIFO, in 32-bit word units, ranging from 0 to 2048.<br><ul style="list-style-type: none"> <li>0: Non-periodic TxFIFO is full</li> <li>1: 1 word available</li> <li>2: 2 words available</li> <li>.....</li> <li>2048: 2048 words available</li> </ul>                                                                      | RO     |

34.6.3.11 Module ID Register (USBHS\_CID)

This register is a programmable user configuration ID register.

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0x3C   | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x12345678   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | PRODUCT_ID   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access | RW           |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit  | Name            | Function         | Description                        | Access |
|------|-----------------|------------------|------------------------------------|--------|
| 31:0 | PROD-<br>UCT_ID | Product ID field | Application-programmable ID field. | RW     |

## 34.6.3.12 LPM Configuration Register (USBHS\_GLPMCFG)

This register is a programmable LPM configuration register.

| Offset | Bit Position |    |        |            |    |         |         |    |          |    |         |        |        |        |          |    |        |         |      |    |        |       |   |   |   |   |   |   |   |   |   |   |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|--------|--------------|----|--------|------------|----|---------|---------|----|----------|----|---------|--------|--------|--------|----------|----|--------|---------|------|----|--------|-------|---|---|---|---|---|---|---|---|---|---|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
| 0x54   | 31           | 30 | 29     | 28         | 27 | 26      | 25      | 24 | 23       | 22 | 21      | 20     | 19     | 18     | 17       | 16 | 15     | 14      | 13   | 12 | 11     | 10    | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Reset  | 0x00000000   |    |        |            |    |         |         |    |          |    |         |        |        |        |          |    |        |         |      |    |        |       |   |   |   |   |   |   |   |   |   |   |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Name   | Reserved     |    | ENBSEL | LPMRCNTSTS |    | SENDLPM | LPMRCNT |    | LPMCHIDX |    | L1RSMOK | SLPSTS | LPMRSP | L1DSEN | BSELTHRS |    | L1SSEN | REMWAKE | BSEL |    | LPMACK | LPMEN |   |   |   |   |   |   |   |   |   |   |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|        |              |    |        |            |    |         |         |    |          |    |         |        |        |        |          |    |        |         |      |    |        |       |   |   |   |   |   |   |   |   |   |   |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

| Bit   | Name       | Function                   | Description                                                                                                                                                                                                                                                                                                                                                                                               | Access |
|-------|------------|----------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:29 | Reserved   | Reserved                   | -                                                                                                                                                                                                                                                                                                                                                                                                         | -      |
| 28    | ENBSEL     | BSEL enable bit            | <ul style="list-style-type: none"> <li>0b0: Complies with the USB 2.0 Link Power Management Addendum Engineering Change Notice to the USB 2.0 specification, July 16, 2007.</li> <li>0b1: Follow the Errata for USB 2.0 ECN: Link Power Management (LPM) -7/2007</li> </ul>                                                                                                                               | RW     |
| 27:25 | LPMRCNTSTS | LPM retry status           | Records the remaining number of LPM retries.<br><div>  <b>Note</b><br/>           Active only in host mode.         </div>                                                                                                                                                                                             | RO     |
| 24    | SENDLPM    | Send LPM transaction       | <ul style="list-style-type: none"> <li>0b0: Do not send LPM transaction.</li> <li>0b1: Send LPM transaction (EXT and LPM tokens).</li> </ul> This bit is automatically cleared when the device returns a response (ACK, STALL, or NYET).<br><div>  <b>Note</b><br/>           Active only in host mode.         </div> | RW     |
| 23:21 | LPMRCNT    | LPM retry count            | When an ERROR response is received, the host retries the LPM transaction until a active response (ACK, STALL, or NYET) is received.<br><div>  <b>Note</b><br/>           Active only in host mode.         </div>                                                                                                      | RW     |
| 20:17 | LPMCHIDX   | LPM transmit channel index | The host channel index used for LPM transactions; the hardware automatically inserts the device address and endpoint information set for this channel.<br><div>  <b>Note</b><br/>           Active only in host mode.         </div>                                                                                   | RW     |

| Bit   | Name     | Function                    | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | Access |
|-------|----------|-----------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 16    | L1RSMOK  | L1 status bit               | <ul style="list-style-type: none"> <li>● 0b0: Current state cannot recover from L1.</li> <li>● 0b1: Current state can recover from L1.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | RO     |
| 15    | SLPSTS   | Sleep status bit            | <b>Device Mode:</b><br>When the device sends an ACK in response to an LPM transaction and the protocol-specified time has elapsed, this bit is set to 1, indicating entry into SLEEP mode.<br>When the bus state changes, disconnects, or a remote wakeup signal is sent, this bit is automatically cleared to 0, exiting SLEEP mode.<br><b>Host Mode</b><br>The host sends an LPM transaction; after receiving an ACK response from the device, this bit is set to 1, indicating entry into SLEEP mode.<br>When the host receives a remote wakeup signal, initiates a wakeup, or initiates a reset, this bit is automatically cleared to 0, exiting SLEEP mode. <ul style="list-style-type: none"> <li>● 0b0: Not in L1</li> <li>● 0b1: In L1</li> </ul> | RO     |
| 14:13 | LPMRSP   | LPM response                | LPM response received by the host or sent by the device. <ul style="list-style-type: none"> <li>● 0b00: ERROR (no response).</li> <li>● 0b01: STALL</li> <li>● 0b10: NYET</li> <li>● 0b11: ACK</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | RO     |
| 12    | L1DSEN   | L1 deep sleep enable bit    | To minimize power consumption, the application should set this bit to 1.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | RW     |
| 11:8  | BSELTHRS | BSEL threshold setting      | <b>Device Mode:</b> When the device receives a BSEL value greater than or equal to the register setting, it enters L1.<br><b>Host Mode:</b> The time for the host to send a resume signal.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | RW     |
| 7     | L1SSEN   | L1 shallow sleep enable bit | To minimize power consumption, the application should set this bit to 1.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | RW     |
| 6     | REMWAKE  | bRemoteWake value           | bRemoteWake value sent by the host or received by the device.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | RW     |
| 5:2   | BSEL     | BSEL setting                | <b>Device Mode:</b> BSEL bmAttribute value, automatically updated after a active LPM transaction response.<br><b>Host Mode:</b> BSEL value sent by the host for LPM transactions, or resume signal length value. <ul style="list-style-type: none"> <li>● 0b0000: 125us</li> <li>● 0b0001: 150us</li> <li>● 0b0010: 200us</li> <li>● 0b0100: 400us</li> <li>● 0b0101: 500us</li> <li>● 0b0110: 1000us</li> </ul>                                                                                                                                                                                                                                                                                                                                          | RW     |

| Bit | Name   | Function       | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | Access |
|-----|--------|----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 5:2 | BSEL   | BSEL setting   | <ul style="list-style-type: none"> <li>● 0b0111: 2000us</li> <li>● 0b1000: 3000us</li> <li>● 0b1001: 4000us</li> <li>● 0b1010: 5000us</li> <li>● 0b1011: 6000us</li> <li>● 0b1100: 7000us</li> <li>● 0b1101: 8000us</li> <li>● 0b1110: 9000us</li> <li>● 0b1111: 10000us</li> <li>● Others: Setting prohibited</li> </ul>                                                                                                                                                                                                                                                                                                                                      | RW     |
| 1   | LPMACK | LPM response   | <ul style="list-style-type: none"> <li>● 0b0: NYET</li> <li>● 0b1: ACK</li> </ul> <p>Even if this bit is written as 1, an ACK response is not sent in the following scenarios (LPM transaction errors) :</p> <ul style="list-style-type: none"> <li>● PID/CRC error exists; in this scenario, an ERROR response is sent.</li> <li>● bLinkState is not equal to 0001b; in this scenario, a STALL response is sent.</li> <li>● Pending data transfer exists; in this scenario, a NYET response is sent.</li> </ul> <div>  <b>Note</b><br/> Active only in device mode. </div> | RW     |
| 0   | LPMEN  | LPM enable bit | <ul style="list-style-type: none"> <li>● 0b0: Disable</li> <li>● 0b1: Enable</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | RW     |

#### 34.6.3.13 Host Periodic Transmit FIFO Size Register (USBHS\_HPTXFSIZ)

This application can program the RAM size that must be allocated to the periodic TxFIFO.

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |       |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0x100  | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15    | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x08001000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |       |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | PTXFD        |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | PTXSA |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access | RW           |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RW    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit   | Name  | Function                           | Description                                                                                                  | Access |
|-------|-------|------------------------------------|--------------------------------------------------------------------------------------------------------------|--------|
| 31:16 | PTXFD | Host periodic TxFIFO depth         | In units of 32-bit words, the minimum value is 16.                                                           | RW     |
| 15:0  | PTXSA | Host periodic TxFIFO start address | The power-on reset value is the sum of the maximum Rx FIFO depth and the maximum non-periodic Tx FIFO depth. | RW     |



### 34.6.3.14 Device IN Endpoint Transmit FIFO Size Register (USBHS\_DIEPTXFx) (x=1-15)

This application can program the size that must be allocated to the device TxFIFO.

Offset:  $0x100 + 0x4 \times x$

| Offset | Bit Position           |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |          |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|------------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| '      | 31                     | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15       | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x08001000+0x800*(x-1) |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |          |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | INEPTXFD               |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | INEPTXSA |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access | RW                     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RW       |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit   | Name     | Function                                     | Description                                                                                                                                     | Access |
|-------|----------|----------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:16 | INEPTXFD | The TxFIFO depth of the device IN end-point. | In units of 32-bit words, the minimum value is 16.                                                                                              | RW     |
| 15:0  | INEPTXSA | IN endpoint TxFIFOx RAM start address        | This field contains the starting address of the memory of the IN endpoint send FIFOx. This address must be aligned to a 32-bit memory location. | RW     |

### 34.6.4 USBHS Host Mode Register

The host mode register affects the operation of the module in host mode. The host mode registers must not be accessed in device mode, as the results produced are ambiguous.

#### 34.6.4.1 Host Configuration Register (USBHS\_HCFG)

This register configures the module after power-on. Do not modify this register after initializing the host.

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |       |   |  |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|-------|---|--|
| 0x400  | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1     | 0 |  |
| Reset  | 0x00000200   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |       |   |  |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   | FSLSS |   |  |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |       |   |  |

| Bit  | Name     | Function                                       | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | Access |
|------|----------|------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:3 | Reserved | Reserved                                       | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | -      |
| 2    | FSLSS    | Only supports FS and LS function control bits. | <p>The application uses this bit to control the enumeration speed of the module. Using this bit, the application can make the module work as FS host even if the connected device supports HS communication. Do not change this field after initial programming.</p> <ul style="list-style-type: none"> <li>0b0: HS/FS/LS, depending on the maximum speed supported by the connected device.</li> <li>0b1: Only FS/LS, even if the connected device can support HS.</li> </ul>                                                                                                                                                                                                  | RW     |
| 1:0  | FSLSPCS  | FS/LS PHY clock select                         | <p>When the module is in FS host mode:</p> <ul style="list-style-type: none"> <li>0b01: PHY clock runs at 48MHz.</li> <li>Others: Setting prohibited</li> </ul> <p>When the module is in LS host mode:</p> <ul style="list-style-type: none"> <li>0b01: PHY clock runs at 48MHz.</li> <li>0b10: PHY clock runs at 6MHz.</li> <li>Others: Setting prohibited</li> </ul> <div>  <b>Notice</b><br/>           When a device is connected to the host, FSLSPCS must be set according to the speed of the connected device (a software reset is required after changing this bit).         </div> | RW     |

### 34.6.4.2 Host Frame Time Interval Register (USBHS\_HFIR)

This register stores the frame interval information set by the USBHS controller for the current speed of the connected device.

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |       |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0x404  | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15    | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x0000EA60   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |       |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | FRIVL |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RW    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit   | Name     | Function       | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | Access |
|-------|----------|----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:16 | Reserved | Reserved       | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | -      |
| 15:0  | FRIVL    | Frame interval | <p>The value programmed by the application in this field is used to specify the time interval between two consecutive SOF(FS) or Keep-Alive tokens(LS). This field contains the number of PHY clocks that make up the desired frame interval. The application can only write values to this register after setting the port enable bit (PENA bit in USBHS_HPRT) of the host port control and status register to 1. If no value is programmed, the module calculates based on the PHY clock specified in the FS/LS PHY clock selection field (FSLSPCS in USBHS_HCFG) of the host configuration register. Do not change the value of this field after initial configuration. Setting value=frame interval (ms) ×(PHY clock frequency) -1</p> <div>  <b>Notice</b><br/>           When the application needs to change the frame interval time, it can modify the FRIVL bit.         </div> | RW     |

## 34.6.4.3 Host Frame Number/Frame Remaining Time Register (USBHS\_HFNUM)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |       |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0x408  | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15    | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00003FFF   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |       |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | FTREM        |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | FRNUM |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access | RO           |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RO    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit   | Name  | Function             | Description                                                                                                                                                                                                                                                                        | Access |
|-------|-------|----------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:16 | FTREM | Frame time remaining | Indicates the remaining time of the current frame (in units of PHY clock counts). This field decrements by 1 for each PHY clock that passes. When the value reaches zero, this field reloads the value from the frame interval register and the module sends a new SOF on the USB. | RO     |
| 15:0  | FRNUM | Frame number         | When a new SOF is sent on the USB, the value of this field increments by 1, and it clears to zero when it reaches 0x3FFF.                                                                                                                                                          | RO     |

#### 34.6.4.4 Host Periodic Transmit FIFO/Queue Status Register (USBHS\_HPTXSTS)

This read-only register contains information about the free space in the periodic Tx FIFO and periodic transmit request queue.

| Offset | Bit Position |    |    |    |    |    |    |    |         |    |    |    |    |    |    |    |          |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |  |  |  |  |  |
|--------|--------------|----|----|----|----|----|----|----|---------|----|----|----|----|----|----|----|----------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|--|--|--|--|--|--|
| 0x410  | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23      | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15       | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |  |  |  |  |  |  |
| Reset  | 0x00080800   |    |    |    |    |    |    |    |         |    |    |    |    |    |    |    |          |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |  |  |  |  |  |
| Name   | PTXQTOP      |    |    |    |    |    |    |    | PTXQSAV |    |    |    |    |    |    |    | PTXFSAVL |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |  |  |  |  |  |
| Access | RO           |    |    |    |    |    |    |    | RO      |    |    |    |    |    |    |    | RO       |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |  |  |  |  |  |

| Bit   | Name     | Function                                        | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | Access |
|-------|----------|-------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:24 | PTXQTOP  | Top of the periodic transmit request queue      | Indicates the item currently being processed by the MAC in the periodic Tx request queue. This register is used for debugging.<br>[31]: Even/odd frame<br><ul style="list-style-type: none"> <li>0b0: Transmit in even frames</li> <li>0b1: Transmit in odd frames</li> </ul> [30:27]: Channel/endpoint number<br>[26:25]: Type<br><ul style="list-style-type: none"> <li>0b00: Input/Output</li> <li>0b01: Zero-length data packet</li> <li>0b10: Reserved</li> <li>0b11: Disable channel command</li> </ul> [24]: Terminate (Last entry for the selected channel) | RO     |
| 23:16 | PTXQSAV  | Periodic transmit request queue space available | Indicates the number of free positions available for writing in the periodic transmit request queue, ranging from 0 to 8. This queue contains both IN and OUT requests.<br><ul style="list-style-type: none"> <li>0: The periodic transmit request queue is full</li> <li>1: 1 position available</li> <li>.....</li> <li>8: 8 positions available</li> </ul>                                                                                                                                                                                                       | RO     |
| 15:0  | PTXFSAVL | Periodic transmit data FIFO space available     | Indicates the number of free positions available for writing in the periodic Tx FIFO. In units of 32-bit words.<br><ul style="list-style-type: none"> <li>0: The periodic Tx FIFO is full</li> <li>1: 1 word available</li> <li>.....</li> <li>n: n words are available (n is the value of the register USBHS_HPTXFSIZ.PTXFD).</li> </ul>                                                                                                                                                                                                                           | RO     |

### 34.6.4.5 Host All Channels Interrupt Register (USBHS\_HAINT)

When an event occurs on the channel, the host all-channel interrupt register interrupts the application using the host channel interrupt bit in the module interrupt register (the HCINT bit in USBHS\_GINTSTS). Each channel corresponds to one interrupt bit, with a maximum of 16 bits.

Bits in this register are also cleared when the application clears an interrupt via the corresponding host channel x interrupt register.

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |       |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0x414  | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15    | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |       |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | HAINT |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RO    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit   | Name     | Function          | Description                                                                                                                                                                                                                                                                                                                                                                   | Access |
|-------|----------|-------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:16 | Reserved | Reserved          | -                                                                                                                                                                                                                                                                                                                                                                             | -      |
| 15:0  | HAINT    | Channel interrupt | Corresponding to Channel 15 to Channel 0: <ul style="list-style-type: none"> <li>0b0: No interrupt was generated on this channel</li> <li>0b1: An interrupt was generated on this channel</li> </ul> <div>  <b>Note</b><br/>           Each bit corresponds to one channel.         </div> | RO     |

### 34.6.4.6 Host All Channels Interrupt Mask Register (USBHS\_HAINTMSK)

The host all-channel interrupt mask register is used in conjunction with the host all-channel interrupt register to interrupt the application when an event occurs on the channel.

Each channel corresponds to 1 interrupt mask bit, up to 16 bits.

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |        |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|--------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0x418  | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15     | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |        |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | HAINTM |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RW     |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit   | Name     | Function               | Description                                                                                                                                                                                                                                                                  | Access |
|-------|----------|------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:16 | Reserved | Reserved               | -                                                                                                                                                                                                                                                                            | -      |
| 15:0  | HAINTM   | Channel interrupt mask | Corresponding to Channel 15 to Channel 0:<br>● 0b0: Maskable interrupt<br>● 0b1: Enable interrupt<br><div>  <b>Note</b><br/>           Each bit corresponds to one channel.         </div> | RW     |

### 34.6.4.7 Host Port Control and Status Register (USBHS\_HPRT)

This register is only available in host mode. Currently, the USBHS host only supports one port.

This register contains information related to the USB port, such as USB reset, enable, suspend, resume, connection status. The PENCHNG/PCDET bit in this register can trigger an application interrupt through the host port interrupt bit in the module interrupt register (HPRTINT bit in USBHS\_GINTST). When a port interrupt occurs, the application must read this register and clear the bit that caused the interrupt. The application must write a 1 to this bit to clear the interrupt.

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |      |       |    |    |    |      |       |    |          |      |       |      |          |         |      |       |       |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|------|-------|----|----|----|------|-------|----|----------|------|-------|------|----------|---------|------|-------|-------|---|---|
| 0x440  | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18   | 17    | 16 | 15 | 14 | 13   | 12    | 11 | 10       | 9    | 8     | 7    | 6        | 5       | 4    | 3     | 2     | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |      |       |    |    |    |      |       |    |          |      |       |      |          |         |      |       |       |   |   |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    | PSPD | PTCTL |    |    |    | PWPR | PLSTS |    | Reserved | PRST | PSUSP | PRES | Reserved | PENCHNG | PENA | PCDET | PCSTS |   |   |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    | RO   | RW    |    |    |    | RW   | RO    |    |          | RW   | RW    | RW   |          | RW      | RW   | RW    | RO    |   |   |

| Bit   | Name     | Function          | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                 | Access |
|-------|----------|-------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:19 | Reserved | Reserved          | -                                                                                                                                                                                                                                                                                                                                                                                                                                                           | -      |
| 18:17 | PSPD     | Port speed        | Indicates the speed of the device connected to the port. <ul style="list-style-type: none"> <li>● 0b00: High speed</li> <li>● 0b01: Full speed</li> <li>● 0b10: Low speed</li> <li>● 0b11: Reserved</li> </ul>                                                                                                                                                                                                                                              | RO     |
| 16:13 | PTCTL    | Port test control | The application writes a non-zero value to this field to place the port in test mode, during which the corresponding mode signal is generated on the port. <ul style="list-style-type: none"> <li>● 0b0000: Test mode disabled</li> <li>● 0b0001: Test_J mode</li> <li>● 0b0010: Test_K mode</li> <li>● 0b0011: Test_SE0_NAK Mode</li> <li>● 0b0100: Test_Packet mode</li> <li>● 0b0101: Test_Force_Enable</li> <li>● Others: Setting prohibited</li> </ul> | RW     |
| 12    | PWPR     | Port power        | Applications use this field to control power to this port. Since the internal PHY of this USBHS does not have power-supply capability, setting this bit to 1 enables the external USB power chip via USBHS_DRVVBUS. <ul style="list-style-type: none"> <li>● 0b0: Power off</li> <li>● 0b1: Power on</li> </ul>                                                                                                                                             | RW     |



| Bit   | Name            | Function                   | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | Access |
|-------|-----------------|----------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 11:10 | PLSTS           | Port line status           | Indicates the current logic level of the USB data line<br>[11]: Logic level of USBHS_DM<br>[10]: Logic level of USBHS_DP                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | RO     |
| 9     | <i>Reserved</i> | <i>Reserved</i>            | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | -      |
| 8     | PRST            | Port reset                 | When an application sets this bit, it initiates a reset sequence on that port. The application must time the reset period and clear this bit after the reset sequence is complete.<br><ul style="list-style-type: none"> <li>● 0b0: Port not in reset state</li> <li>● 0b1: Port in reset state</li> </ul> The application must set this bit for a minimum of 10ms to initiate a reset on the port.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | RW     |
| 7     | PSUSP           | Port suspend               | Applications set this bit to put this port in suspend mode. Only when this bit is set will the module stop sending SOF. To stop the PHY clock, the application must set the port clock stop bit, which enables the PHY's suspend input pin.<br>A read value of this bit reflects the current pending state of the port. After the remote wake-up signal is detected, or the application program sets the port reset bit or port recovery bit in this register to 1, the module can clear this bit; or the application program sets the recovery/remote wake-up detection interrupt bit or interrupt bit in the module interrupt register. The open connection detection interrupt bit (respectively WKUINT or DISCINT in USBHS_GINTSTS) is set to 1, and the module can also clear this bit.<br><ul style="list-style-type: none"> <li>● 0b0: The port is not in suspend mode</li> <li>● 0b1: Port in suspend mode</li> </ul> | RW     |
| 6     | PRES            | Port resume                | The application sets this bit to drive a resume signal on this port. The module will continue to drive the resume signal until the application program clears this bit.<br>As indicated by the port resume/remote wakeup detect interrupt bit in the module interrupt register (WKUINT bit in USBHS_GINTSTS), if the module detects a USB remote wakeup sequence, it starts driving the resume signal without intervention from the application; if the module detects a disconnect If connected, clear this bit to 0.<br>The read value of this bit indicates whether the module is currently driving a resume signal.<br><ul style="list-style-type: none"> <li>● 0b0: Resume signal not driven</li> <li>● 0b1: Resume signal driven</li> </ul>                                                                                                                                                                             | RW     |
| 5:4   | <i>Reserved</i> | <i>Reserved</i>            | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | -      |
| 3     | PENCHNG         | Port enable/disable change | The module sets this bit to 1 when the port enable bit 2 in this register changes state.<br>Write 1 to this bit to clear it to 0 by software.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | RW     |

| Bit | Name  | Function              | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | Access |
|-----|-------|-----------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 2   | PENA  | Port enable           | <p>After the port performs a reset sequence, it can only be enabled by the module and can be disabled by an overcurrent condition, a disconnect condition, or by clearing this bit by the application. The application cannot set this bit by writing to the register. The port can only be disabled by clearing this bit. Manipulation of this bit will not trigger any interrupt to the application.</p> <ul style="list-style-type: none"> <li>● 0b0: Port disabled</li> <li>● 0b1: Port enabled</li> </ul> | RW     |
| 1   | PCDET | Port connect detected | <p>When a device connection is detected, the module sets this bit to 1 to trigger an interrupt to the application using the host port interrupt bit in the module interrupt register (HPRTINT bit in USBHS_GINTSTS).</p> <p>The application must set this bit to 1 to clear the interrupt.</p>                                                                                                                                                                                                                 | RW     |
| 0   | PCSTS | Port connect status   | <ul style="list-style-type: none"> <li>● 0b0: No device connected to port</li> <li>● 0b1: Device connected to port</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                  | RO     |

#### 34.6.4.8 Host Channel x Characteristics Register (USBHS\_HCCHARx) (x=0-15)

This register is used to set the host channel characteristics.

Offset: 0x500+0x20\*x

| Offset | Bit Position |       |        |     |    |    |    |    |    |    |    |    |       |       |          |       |       |    |    |    |    |    |   |   |       |   |   |   |   |   |   |   |  |  |  |  |
|--------|--------------|-------|--------|-----|----|----|----|----|----|----|----|----|-------|-------|----------|-------|-------|----|----|----|----|----|---|---|-------|---|---|---|---|---|---|---|--|--|--|--|
| -      | 31           | 30    | 29     | 28  | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19    | 18    | 17       | 16    | 15    | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7     | 6 | 5 | 4 | 3 | 2 | 1 | 0 |  |  |  |  |
| Reset  | 0x00000000   |       |        |     |    |    |    |    |    |    |    |    |       |       |          |       |       |    |    |    |    |    |   |   |       |   |   |   |   |   |   |   |  |  |  |  |
| Name   | CHENA        | CHDIS | ODDFRM | DAD |    |    |    |    |    |    |    | MC | EPTYP | LSDEV | Reserved | EPDIR | EPNUM |    |    |    |    |    |   |   | MPSIZ |   |   |   |   |   |   |   |  |  |  |  |
| Access | RW           | RW    | RW     | RW  |    |    |    |    |    |    |    | RW | RW    | RW    |          | RW    | RW    |    |    |    |    |    |   |   | RW    |   |   |   |   |   |   |   |  |  |  |  |

| Bit   | Name   | Function                  | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | Access |
|-------|--------|---------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31    | CHENA  | Channel enable            | This field is set by application software and cleared by USBHS host hardware.<br><ul style="list-style-type: none"> <li>0b0: Disable channel</li> <li>0b1: Enable channel</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                             | RW     |
| 30    | CHDIS  | Channel disable           | The application sets this bit to 1 to stop sending/receiving data through the channel, even if the transmission through the channel is not completed, the stop operation still takes effect. The application must wait for the disabled channel interrupt to confirm that the channel has been disabled.                                                                                                                                                                                                                                                                                                                                         | RW     |
| 29    | ODDFRM | Odd frame                 | This field is set or reset by the application to indicate that the USBHS host must transmit odd or even frames, respectively. This field applies only to periodic (synchronous and interrupt) transactions.<br><ul style="list-style-type: none"> <li>0b0: Even (micro) frame</li> <li>0b1: Odd (micro) frame</li> </ul>                                                                                                                                                                                                                                                                                                                         | RW     |
| 28:22 | DAD    | Device address            | This field is used to specify a specific device to communicate with this host.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | RW     |
| 21:20 | MC     | (Multi Count/Error Count) | <b>When the SPLITEN (split enable) bit in USBHS_HCSPLTx is cleared to 0:</b> This field informs the host of the number of transactions that must be performed per microframe for this periodic endpoint. For non-periodic transfers, this field specifies the number of packets this channel should acquire before the internal DMA engine re-arbitrates.<br><ul style="list-style-type: none"> <li>0b00: Setting prohibited</li> <li>0b01: This endpoint requires 1 transaction per microframe</li> <li>0b10: This endpoint requires 2 transaction per microframe</li> <li>0b11: This endpoint requires 3 transaction per microframe</li> </ul> | RW     |

| Bit   | Name     | Function            | Description                                                                                                                                                                                                      | Access |
|-------|----------|---------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
|       |          |                     | <b>When the SPLITEN bit in USBHS_HCSPLTx is set to 1:</b> This field indicates the immediate retry count to be attempted upon error during periodic split transactions. This field must be set to at least 0b01. |        |
| 19:18 | EPTYP    | Endpoint type       | Indicates the selected transport type. <ul style="list-style-type: none"> <li>● 0b00: Control</li> <li>● 0b01: Isochronous</li> <li>● 0b10: Bulk</li> <li>● 0b11: Interrupt</li> </ul>                           | RW     |
| 17    | LSDEV    | Low-speed device    | This field is set by the application to indicate that this channel is communicating with a low-speed device.                                                                                                     | RW     |
| 16    | Reserved | Reserved            | -                                                                                                                                                                                                                | -      |
| 15    | EPDIR    | Endpoint direction  | Indicates whether the direction of the communication transaction is input or output. <ul style="list-style-type: none"> <li>● 0b0: Output</li> <li>● 0b1: Input</li> </ul>                                       | RW     |
| 14:11 | EPNUM    | Endpoint number     | Indicates the endpoint number of the USB device to communicate with this host channel.                                                                                                                           | RW     |
| 10:0  | MPSIZ    | Maximum packet size | Indicates the maximum packet size for device endpoints communicating with this host channel.                                                                                                                     | RW     |

#### 34.6.4.9 Host Channel x Separation Control Register (USBHS\_HCSPLTx) (x=0-15)

Offset: 0x504+0x20\*x

| Offset | Bit Position |          |    |    |    |    |    |    |    |    |    |    |    |    |    |           |         |         |    |    |    |         |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----------|----|----|----|----|----|----|----|----|----|----|----|----|----|-----------|---------|---------|----|----|----|---------|---|---|---|---|---|---|---|---|---|---|
| -      | 31           | 30       | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16        | 15      | 14      | 13 | 12 | 11 | 10      | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |          |    |    |    |    |    |    |    |    |    |    |    |    |    |           |         |         |    |    |    |         |   |   |   |   |   |   |   |   |   |   |
| Name   | SPLITEN      | Reserved |    |    |    |    |    |    |    |    |    |    |    |    |    | COMPLSPLT | XACTPOS | HUBADDR |    |    |    | PRTADDR |   |   |   |   |   |   |   |   |   |   |
| Access | RW           |          |    |    |    |    |    |    |    |    |    |    |    |    |    | RW        | RW      | RW      |    |    |    | RW      |   |   |   |   |   |   |   |   |   |   |

| Bit   | Name      | Function          | Description                                                                                                                    | Access |
|-------|-----------|-------------------|--------------------------------------------------------------------------------------------------------------------------------|--------|
| 31    | SPLITEN   | Split enable      | When the application sets this bit to 1, it indicates that the channel is enabled to perform split communication transactions. | RW     |
| 30:17 | Reserved  | Reserved          | -                                                                                                                              | -      |
| 16    | COMPLSPLT | Do complete split | When the application sets this bit to 1, it can request the host to perform a complete split communication transaction.        | RW     |

| Bit   | Name    | Function             | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | Access |
|-------|---------|----------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 15:14 | XACTPOS | Transaction position | <p>This field is used to determine whether to send the entire, first, middle, or last payload with each OUT transaction.</p> <ul style="list-style-type: none"> <li>● 0b00: Middle. Refers to the middle payload of this transaction (greater than 188 bytes).</li> <li>● 0b01: End. Refers to the last payload of this transaction (greater than 188 bytes).</li> <li>● 0b10: Start. Refers to the first data payload of this transaction (greater than 188 bytes).</li> <li>● 0b11: All. Refers to the entire data payload of this transaction (less than or equal to 188 bytes).</li> </ul> | RW     |
| 13:7  | HUBADDR | Hub address          | This field stores the hub device address of the transaction forwarder.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | RW     |
| 6:0   | PRTADDR | Port address         | This field is the port number of the recipient transaction forwarder.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | RW     |

#### 34.6.4.10 Host Channel x Interrupt Register (USBHS\_HCINTx) (x=0-15)

This register indicates the state of the channel when USB and AHB-related events occur. When the host channel interrupt bit (HCINT bit in USBHS\_GINTSTS) in the module interrupt register is set to 1, the application must read this register. Before performing a read operation on the register, the application must first read the host all-channel interrupt (USBHS\_HAINT) register to obtain the accurate channel number for the host channel x interrupt register. The application must clear the corresponding bits in this register to clear the corresponding bits in the USBHS\_HAINT and USBHS\_GINTSTS registers.

Offset: 0x508+0x20\*x

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |       |       |       |       |      |     |     |       |        |     |      |    |    |    |    |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|-------|-------|-------|-------|------|-----|-----|-------|--------|-----|------|----|----|----|----|
| -      | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8     | 7     | 6     | 5     | 4    | 3   | 2   | 1     | 0      |     |      |    |    |    |    |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |       |       |       |       |      |     |     |       |        |     |      |    |    |    |    |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   | DTERR | FRMOR | BBERR | TXERR | NYET | ACK | NAK | STALL | AHBERR | CHH | XFRC |    |    |    |    |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   | RW    | RW    | RW    | RW    | RW   | RW  | RW  | RW    | RW     | RW  | RW   | RW | RW | RW | RW |

| Bit   | Name     | Function                                                                 | Description                                                                                                                                                                                                                                                  | Access |
|-------|----------|--------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:11 | Reserved | Reserved                                                                 | -                                                                                                                                                                                                                                                            | -      |
| 10    | DTERR    | Data synchroni-<br>zation error                                          | The application must clear this bit by writing 1.                                                                                                                                                                                                            | RW     |
| 9     | FRMOR    | Frame overflow<br>error                                                  | The application needs to write 1 to clear this bit.                                                                                                                                                                                                          | RW     |
| 8     | BBERR    | Crosstalk error                                                          | A typical cause of a crosstalk event is that an endpoint sends a packet, but the packet length exceeds the endpoint's maximum packet size.<br>The application needs to write 1 to clear this bit.                                                            | RW     |
| 7     | TXERR    | Communication<br>transaction er-<br>ror                                  | Indicates one of the following errors occurred on the USB:<br><ul style="list-style-type: none"> <li>● CRC check failed</li> <li>● Timeout</li> <li>● Bit stuffing error</li> <li>● Wrong EOP</li> </ul> The application needs to write 1 to clear this bit. | RW     |
| 6     | NYET     | NYET response<br>received                                                | The application needs to write 1 to clear this bit.                                                                                                                                                                                                          | RW     |
| 5     | ACK      | Receive/Trans-<br>mit ACK re-<br>sponses re-<br>ceived/transi-<br>mitted | The application must clear this bit by writing 1.                                                                                                                                                                                                            | RW     |
| 4     | NAK      | NAK response<br>received                                                 | The application needs to write 1 to clear this bit.                                                                                                                                                                                                          | RW     |
| 3     | STALL    | STALL response<br>received                                               | The application needs to write 1 to clear this bit.                                                                                                                                                                                                          | RW     |

| Bit | Name   | Function                | Description                                                                                                                                                                                                                                                                     | Access |
|-----|--------|-------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 2   | AHBERR | AHB error               | This error is generated only when in internal DMA mode and an AHB error occurs during an AHB read/write operation. The application can obtain the error address by reading the corresponding DMA channel address register.<br>The application must clear this bit by writing 1. | RW     |
| 1   | CHH    | Channel halted          | The transfer ended abnormally due to any USB transaction error or in response to an application inhibit request.<br>The application must clear this bit by writing 1.                                                                                                           | RW     |
| 0   | XFRC   | Transmission completion | The transfer completed normally without any errors.<br>The application needs to write 1 to clear this bit.                                                                                                                                                                      | RW     |

#### 34.6.4.11 Host Channel x Interrupt Mask Register (USBHS\_HCINTMSKx) (x=0-15)

This register is used to select the mask for the host channel interrupts.

Offset: 0x50C+0x20\*x

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |        |        |        |        |       |       |       |         |         |       |        |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|--------|--------|--------|--------|-------|-------|-------|---------|---------|-------|--------|
| -      | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9      | 8      | 7      | 6      | 5     | 4     | 3     | 2       | 1       | 0     |        |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |        |        |        |        |       |       |       |         |         |       |        |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | DTERRM | FRMORM | WBERRM | TXERRM | WNETM | WACKM | WNAKM | WSTALLM | WABERRM | WCHHM | WXRFCM |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RW     | RW     | RW     | RW     | RW    | RW    | RW    | RW      | RW      | RW    | RW     |

| Bit   | Name     | Function                                                | Description                                                                                              | Access |
|-------|----------|---------------------------------------------------------|----------------------------------------------------------------------------------------------------------|--------|
| 31:11 | Reserved | Reserved                                                | -                                                                                                        | -      |
| 10    | DTERRM   | Data synchroni-<br>zation error<br>mask                 | <ul style="list-style-type: none"> <li>0b0: Maskable interrupt</li> <li>0b1: Enable interrupt</li> </ul> | RW     |
| 9     | FRMORM   | Frame overflow<br>mask                                  | <ul style="list-style-type: none"> <li>0b0: Maskable interrupt</li> <li>0b1: Enable interrupt</li> </ul> | RW     |
| 8     | BBERRM   | Crosstalk error<br>mask                                 | <ul style="list-style-type: none"> <li>0b0: Maskable interrupt</li> <li>0b1: Enable interrupt</li> </ul> | RW     |
| 7     | TXERRM   | Communication<br>transaction er-<br>ror mask            | <ul style="list-style-type: none"> <li>0b0: Maskable interrupt</li> <li>0b1: Enable interrupt</li> </ul> | RW     |
| 6     | NYETM    | NYET response<br>receive inter-<br>rupt mask            | <ul style="list-style-type: none"> <li>0b0: Maskable interrupt</li> <li>0b1: Enable interrupt</li> </ul> | RW     |
| 5     | ACKM     | ACK response<br>receive/trans-<br>mit interrupt<br>mask | <ul style="list-style-type: none"> <li>0b0: Maskable interrupt</li> <li>0b1: Enable interrupt</li> </ul> | RW     |
| 4     | NAKM     | NAK response<br>receive inter-<br>rupt mask             | <ul style="list-style-type: none"> <li>0b0: Maskable interrupt</li> <li>0b1: Enable interrupt</li> </ul> | RW     |
| 3     | STALLM   | Receive STALL<br>response inter-<br>rupt mask           | <ul style="list-style-type: none"> <li>0b0: Maskable interrupt</li> <li>0b1: Enable interrupt</li> </ul> | RW     |
| 2     | AHBERRM  | AHBERR inter-<br>rupt mask                              | <ul style="list-style-type: none"> <li>0b0: Maskable interrupt</li> <li>0b1: Enable interrupt</li> </ul> | RW     |
| 1     | CHHM     | Channel stop<br>interrupt mask                          | <ul style="list-style-type: none"> <li>0b0: Maskable interrupt</li> <li>0b1: Enable interrupt</li> </ul> | RW     |



| Bit | Name | Function                          | Description                                                                                               | Access |
|-----|------|-----------------------------------|-----------------------------------------------------------------------------------------------------------|--------|
| 0   | XFCM | Transfer completed interrupt mask | <ul style="list-style-type: none"><li>● 0b0: Maskable interrupt</li><li>● 0b1: Enable interrupt</li></ul> | RW     |

#### 34.6.4.12 Host Channel x Transfer Size Register (USBHS\_HCTSIZx) (x=0-15)

This register is used to set the host channel transfer size and data PID.

Offset: 0x510+0x20\*x

| Offset | Bit Position |      |        |    |    |    |    |    |    |    |    |    |    |        |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|------|--------|----|----|----|----|----|----|----|----|----|----|--------|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| -      | 31           | 30   | 29     | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18     | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |      |        |    |    |    |    |    |    |    |    |    |    |        |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | DOPING       | DPID | PKTCNT |    |    |    |    |    |    |    |    |    |    | XFRSIZ |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access | RW           | RW   | RW     |    |    |    |    |    |    |    |    |    |    | RW     |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit   | Name   | Function      | Description                                                                                                                                                                                                                                                                                                                                                              | Access |
|-------|--------|---------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31    | DOPING | Do ping       | <p>This bit is only used for OUT transfers. Setting this bit to 1 instructs the host to perform the PING protocol.</p> <div>  <b>Notice</b><br/>           Do not set this bit to 1 for IN transfers. If this bit is set to 1 for IN transfers, the channel is disabled.         </div> | RW     |
| 30:29 | DPID   | Data PID      | <p>The application sets the initial synchronization PID for data communication in this field. The host retains the setting of this field throughout the transfer transaction.</p> <ul style="list-style-type: none"> <li>● 0b00: DATA0</li> <li>● 0b01: DATA2</li> <li>● 0b10: DATA1</li> <li>● 0b11: MDATA/SETUP</li> </ul>                                             | RW     |
| 28:19 | PKTCNT | Packet count  | <p>The application sets the number of data packets to be sent or received in this field. The host decrements the count value each time a packet is successfully sent or received. After this value reaches 0, the application will be interrupted to indicate that the operation completed normally.</p>                                                                 | RW     |
| 18:0  | XFRSIZ | Transfer size | <p>For OUT operations, this field is the number of data bytes sent by the host during the transfer. For IN operations, this field is the buffer size reserved by the application for transfers. For IN transactions (periodic and non-periodic), the application will program this field to an integer multiple of the maximum packet size.</p>                          | RW     |

**34.6.4.13 Host Channel x DMA Address Register (USBHS\_HCDMAx) (x=0-15)**

This register is used to set the DMA address when the host is in DMA mode.

Offset: 0x514+0x20\*x

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| -      | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0xFFFFFFFF   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | DMAADDR      |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access | RW           |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit  | Name    | Function    | Description                                                                                                                                                                                                                | Access |
|------|---------|-------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:0 | DMAADDR | DMA address | This field stores the address of the memory used for DMA transfers when the host obtains data from the device endpoint or sends data to the device endpoint. This register is incremented at the end of each AHB transfer. | RW     |

### 34.6.5 USBHS Device Mode Register

The device mode registers affect the operation of the module in device mode. The device mode registers must not be accessed in host mode because the results are ambiguous.

#### 34.6.5.1 Device Configuration Register (USBHS\_DCFG)

This register configures the module into device mode after power-on, execution of certain control commands, or enumeration. Do not change this register after initial programming.

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |       |    |     |    |    |    |          |   |          |   |      |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-------|----|-----|----|----|----|----------|---|----------|---|------|---|---|---|---|---|
| 0x800  | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15    | 14 | 13  | 12 | 11 | 10 | 9        | 8 | 7        | 6 | 5    | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x08220000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |       |    |     |    |    |    |          |   |          |   |      |   |   |   |   |   |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | PFIVL |    | DAD |    |    |    | Reserved |   | NZLSOHSK |   | DSPD |   |   |   |   |   |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RW    |    | RW  |    |    |    |          |   | RW       |   | RW   |   |   |   |   |   |

| Bit   | Name     | Function                             | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | Access |
|-------|----------|--------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:13 | Reserved | Reserved                             | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | -      |
| 12:11 | PFIVL    | Periodic frame interval              | Indicates the point in time within a frame when the periodic frame interrupt must be used to notify the application. This function can be used to determine if all isochronous communications for that frame are complete. <ul style="list-style-type: none"> <li>● 0b00: 80% (micro) frame interval</li> <li>● 0b01: 85% (micro) frame interval</li> <li>● 0b10: 90% (micro) frame interval</li> <li>● 0b11: 95% (micro) frame interval</li> </ul>                                                                                                                                                                                              | RW     |
| 10:4  | DAD      | Device address                       | The application must set this field according to the command parameters after executing each SetAddress control command.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | RW     |
| 3     | Reserved | Reserved                             | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | -      |
| 2     | NZLSOHSK | Non-zero-length status OUT handshake | During the OUT transaction of the control transfer status stage, when the module receives a non-zero-length packet, the application can use this field to select the handshake signal to be sent. <ul style="list-style-type: none"> <li>● 0b0: Send the received OUT packet (zero-length or non-zero-length) to the application and reply with a handshake signal based on the NAK and STALL bits of the endpoint in the device endpoint control register.</li> <li>● 0b1: When receiving a non-zero-length status OUT transaction, reply with a STALL handshake signal, and the received OUT packet is not sent to the application.</li> </ul> | RW     |

| Bit | Name | Function     | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | Access |
|-----|------|--------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 1:0 | DSPD | Device speed | <p>Indicates the speed at which the application requires the module to enumerate or the maximum speed supported by the application. However, the actual bus speed can only be determined after the chirp sequence is completed, and this speed is based on the speed of the USB host connected to the module.</p> <ul style="list-style-type: none"><li>● 0b00: High-speed</li><li>● 0b01: Setting prohibited</li><li>● 0b10: Setting prohibited</li><li>● 0b11: Full-speed (USB1.1 transceiver clock is 48MHz)</li></ul> | RW     |

## 34.6.5.2 Device Control Register (USBHS\_DCTL)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |          |        |        |        |        |      |   |        |        |      |        |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----------|--------|--------|--------|--------|------|---|--------|--------|------|--------|---|
| 0x804  | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11       | 10     | 9      | 8      | 7      | 6    | 5 | 4      | 3      | 2    | 1      | 0 |
| Reset  | 0x00000002   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |          |        |        |        |        |      |   |        |        |      |        |   |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | POPRGDNE | CGONAK | SGONAK | CGINAK | SGINAK | TCTL |   | GONSTS | GINSTS | SDIS | RWUSIG |   |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RW       | RW     | RW     | RW     | RW     | RW   |   | RO     | RO     | RW   | RW     |   |

| Bit   | Name     | Function                  | Description                                                                                                                                                                                                                                                                                                                                        | Access |
|-------|----------|---------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:12 | Reserved | Reserved                  | -                                                                                                                                                                                                                                                                                                                                                  | -      |
| 11    | POPRGDNE | Power-on programming done | The application uses this bit to indicate that the register has finished programming after waking up from Power-down mode.                                                                                                                                                                                                                         | RW     |
| 10    | CGONAK   | Clear global OUT NAK      | Writing to this bit will clear the global OUT NAK.                                                                                                                                                                                                                                                                                                 | RW     |
| 9     | SGONAK   | Set global OUT NAK        | Writing to this bit sets the global OUT NAK. Applications use this bit to send NAK handshake signals on all OUT endpoints. The application program can only set this bit to 1 when it is sure that the global OUT NAK active bit (GONAKEFF bit in USBFS_GINTSTS) in the module interrupt register has been cleared.                                | RW     |
| 8     | CGINAK   | Clear global IN NAK       | Writing to this bit will clear the global IN NAK.                                                                                                                                                                                                                                                                                                  | RW     |
| 7     | SGINAK   | Set global IN NAK         | Writing to this field sets the global non-periodic IN NAK. Applications use this bit to cause all non-periodic IN endpoints to send NAK handshake signals. The application program can only set this bit to 1 when it is sure that the global IN NAK active bit (GINAKEFF bit in USBFS_GINTSTS) in the module interrupt register has been cleared. | RW     |
| 6:4   | TCTL     | Test control              | <ul style="list-style-type: none"> <li>0b000: Test mode disabled</li> <li>0b001: Test_J mode</li> <li>0b010: Test_K mode</li> <li>0b011: Test_SE0_NAK Mode</li> <li>0b100: Test_Packet mode</li> <li>0b101: Test_Force_Enable</li> <li>Others: Setting prohibited</li> </ul>                                                                       | RW     |

| Bit | Name   | Function                 | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | Access |
|-----|--------|--------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 3   | GONSTS | Global OUT NAK status    | <ul style="list-style-type: none"> <li>0b0: Handshake will be sent according to FIFO status and NAK and STALL bit settings.</li> <li>0b1: No data is received regardless of whether there is free space in the RxFIFO. Reply NAK handshake to all received packets except SET-UP transactions. All OUT packets of type isochronous will be dropped.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | RO     |
| 2   | GINSTS | Global IN NAK status     | <ul style="list-style-type: none"> <li>0b0: Handshake will be replied based on data availability in transmit FIFO.</li> <li>0b1: Make all non-periodic IN endpoints reply with NAK handshake signals, regardless of the availability of data in the send FIFO.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | RO     |
| 1   | SDIS   | Soft disconnect          | <p>The application uses this bit to signal the USBHS module to perform a soft disconnect. When this bit is set, the host will not see that the device is attached, and the device will not receive signals on the USB. The module remains disconnected until the application program clears this bit.</p> <ul style="list-style-type: none"> <li>0b0: Normal operation. Clearing this bit after a soft disconnect causes the host to receive a device connected event. After reconnecting the device, the USB host restarts device enumeration.</li> <li>0b1: Make the host receive a device disconnect event.</li> </ul> <p>At full speed, the minimum time for soft disconnection is specified as follows:</p> <ul style="list-style-type: none"> <li>Suspended state: the minimum time is 1ms+2.5μs</li> <li>Idle state: 2.5μs</li> <li>Non-idle or suspend state: 2.5μs</li> </ul> | RW     |
| 0   | RWUSIG | Remote wake-up signaling | When the application program sets this bit to 1, the module will initiate a remote wake-up signal to wake up the USB host. The application must set this bit to bring the module out of the suspended state. According to the USB2.0 specification, the application must clear this bit within 1ms to 15ms of setting it.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | RW     |

### 34.6.5.3 Device Status Register (USBHS\_DSTS)

This register indicates the status of the module when USB-related events occur. When an interrupt occurs, the endpoint information must be read from the device all-segment endpoint interrupt register (USBHS\_DAINTE).

| Offset | Bit Position |    |    |    |    |    |    |    |       |       |    |    |    |    |    |    |    |    |    |    |    |          |   |   |   |      |         |   |         |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|-------|-------|----|----|----|----|----|----|----|----|----|----|----|----------|---|---|---|------|---------|---|---------|---|---|---|
| 0x808  | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23    | 22    | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10       | 9 | 8 | 7 | 6    | 5       | 4 | 3       | 2 | 1 | 0 |
| Reset  | 0x00000002   |    |    |    |    |    |    |    |       |       |    |    |    |    |    |    |    |    |    |    |    |          |   |   |   |      |         |   |         |   |   |   |
| Name   | Reserved     |    |    |    |    |    |    |    | LNSTS | FNSOF |    |    |    |    |    |    |    |    |    |    |    | Reserved |   |   |   | EERR | ENUMSPD |   | SUSPSTS |   |   |   |
| Access |              |    |    |    |    |    |    |    | RO    | RO    |    |    |    |    |    |    |    |    |    |    |    |          |   |   |   | RO   | RO      |   | RO      |   |   |   |

| Bit   | Name     | Function                 | Description                                                                                                                                                                                                                                                                                                                                                                                                                                               | Access |
|-------|----------|--------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:24 | Reserved | Reserved                 | -                                                                                                                                                                                                                                                                                                                                                                                                                                                         | -      |
| 23:22 | LNSTS    | USB bus status           | [23]: Logical level of D+.<br>[22]: Logical level of D-.                                                                                                                                                                                                                                                                                                                                                                                                  | RO     |
| 21:8  | FNSOF    | Receive SOF frame number | -                                                                                                                                                                                                                                                                                                                                                                                                                                                         | RO     |
| 7:4   | Reserved | Reserved                 | -                                                                                                                                                                                                                                                                                                                                                                                                                                                         | -      |
| 3     | EERR     | Erratic error            | The module sets this bit to 1 to report any undefined errors.<br>Due to undefined errors, the USBHS controller enters a suspend state and generate an interrupt via the early suspend bit (ESUSP bit in USBHS_GINTSTS) in the USBHS_GINTSTS register. If the early suspend interrupt is triggered by an undefined error, the application can only perform a soft disconnect to resume communication.                                                      | RO     |
| 2:1   | ENUMSPD  | Enumerated speed         | Indicates the speed at which the USBHS controller is enumerated after detecting the speed via the chirp sequence.<br><ul style="list-style-type: none"> <li>0b00: High-speed</li> <li>0b01: Reserved</li> <li>0b10: Reserved</li> <li>0b11: Full-speed (PHY clock runs at 48MHz).</li> </ul>                                                                                                                                                              | RO     |
| 0     | SUSPSTS  | Suspend status           | In device mode, this bit is set to 1 whenever a suspend state is detected on the USB. When the idle state on the USB bus persists for 3ms, the module enters a suspend state. The module exits the suspend state under the following conditions: <ul style="list-style-type: none"> <li>Activity on the USB data lines.</li> <li>The application writes to the remote wakeup signal bit (RWUSIG bit in USBHS_DCTL) in the USBHS_DCTL register.</li> </ul> | RO     |



#### 34.6.5.4 Device IN Endpoint General Interrupt Mask Register (USBHS\_DIEPMSK)

This register is used in conjunction with the USBHS\_DIEPINTx registers of all endpoints to generate interrupts on each IN endpoint. By performing a write operation on the corresponding bit of this register, the IN endpoint interrupt in the USBHS\_DIEPINTx register can be masked. By default, all status interrupts are masked.

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |      |          |    |    |   |        |          |        |        |          |     |          |      |       |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|------|----------|----|----|---|--------|----------|--------|--------|----------|-----|----------|------|-------|
| 0x810  | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13   | 12       | 11 | 10 | 9 | 8      | 7        | 6      | 5      | 4        | 3   | 2        | 1    | 0     |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |      |          |    |    |   |        |          |        |        |          |     |          |      |       |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | NAKM | Reserved |    |    |   | TXFURM | Reserved | INPNEM | INPNMM | TTXFEMSK | TOM | Reserved | EPDM | XFCRM |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RW   |          |    |    |   | RW     |          | RW     | RW     | RW       | RW  |          | RW   | RW    |

| Bit   | Name     | Function                                               | Description                                                                                              | Access |
|-------|----------|--------------------------------------------------------|----------------------------------------------------------------------------------------------------------|--------|
| 31:14 | Reserved | Reserved                                               | -                                                                                                        | -      |
| 13    | NAKM     | NAK interrupt mask                                     | <ul style="list-style-type: none"> <li>0b0: Maskable interrupt</li> <li>0b1: Enable interrupt</li> </ul> | RW     |
| 12:9  | Reserved | Reserved                                               | -                                                                                                        | -      |
| 8     | TXFURM   | Transmit FIFO underflow interrupt mask                 | <ul style="list-style-type: none"> <li>0b0: Maskable interrupt</li> <li>0b1: Enable interrupt</li> </ul> | RW     |
| 7     | Reserved | Reserved                                               | -                                                                                                        | -      |
| 6     | INEPNEM  | IN endpoint NAK active interrupt mask                  | <ul style="list-style-type: none"> <li>0b0: Maskable interrupt</li> <li>0b1: Enable interrupt</li> </ul> | RW     |
| 5     | INEPNMM  | IN token interrupt mask received on EP mismatch        | <ul style="list-style-type: none"> <li>0b0: Maskable interrupt</li> <li>0b1: Enable interrupt</li> </ul> | RW     |
| 4     | TTXFEMSK | IN token interrupt mask received when Tx FIFO is empty | <ul style="list-style-type: none"> <li>0b0: Maskable interrupt</li> <li>0b1: Enable interrupt</li> </ul> | RW     |
| 3     | TOM      | Timeout interrupt mask (Non synchronous endpoint)      | <ul style="list-style-type: none"> <li>0b0: Maskable interrupt</li> <li>0b1: Enable interrupt</li> </ul> | RW     |
| 2     | Reserved | Reserved                                               | -                                                                                                        | -      |
| 1     | EPDM     | Endpoint disabled interrupt mask                       | <ul style="list-style-type: none"> <li>0b0: Maskable interrupt</li> <li>0b1: Enable interrupt</li> </ul> | RW     |

| Bit | Name | Function                          | Description                                                                                               | Access |
|-----|------|-----------------------------------|-----------------------------------------------------------------------------------------------------------|--------|
| 0   | XFCM | Transfer completed interrupt mask | <ul style="list-style-type: none"><li>● 0b0: Maskable interrupt</li><li>● 0b1: Enable interrupt</li></ul> | RW     |

### 34.6.5.5 Device OUT Endpoint General Interrupt Mask Register (USBHS\_DOEPMSK)

This register works in conjunction with the individual USBHS\_DOEPINTx registers for all endpoints to generate interrupts on each OUT endpoint. OUT endpoint interrupts in the USBHS\_DOEPINTx registers can be masked by writing to the corresponding bit in this register. By default, all status interrupts are masked.

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |       |    |          |    |   |   |   |   |   |   |   |   |      |   |          |  |         |  |          |  |        |  |       |  |          |  |      |  |       |  |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-------|----|----------|----|---|---|---|---|---|---|---|---|------|---|----------|--|---------|--|----------|--|--------|--|-------|--|----------|--|------|--|-------|--|
| 0x814  | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13    | 12 | 11       | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1    | 0 |          |  |         |  |          |  |        |  |       |  |          |  |      |  |       |  |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |       |    |          |    |   |   |   |   |   |   |   |   |      |   |          |  |         |  |          |  |        |  |       |  |          |  |      |  |       |  |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | NYETM |    | Reserved |    |   |   |   |   |   |   |   |   | OPEM |   | Reserved |  | B2BSTUP |  | Reserved |  | OTEPDM |  | STUPM |  | Reserved |  | EPDM |  | XFCRM |  |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RW    |    |          |    |   |   |   |   |   |   |   |   | RW   |   |          |  | RW      |  |          |  | RW     |  | RW    |  |          |  | RW   |  | RW    |  |

| Bit   | Name     | Function                                                 | Description                                                                                                                                                                                                                                                                 | Access |
|-------|----------|----------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:15 | Reserved | Reserved                                                 | -                                                                                                                                                                                                                                                                           | -      |
| 14    | NYETM    | NYET interrupt mask                                      | <ul style="list-style-type: none"> <li>0b0: Maskable interrupt</li> <li>0b1: Enable interrupt</li> </ul>                                                                                                                                                                    | RW     |
| 13:9  | Reserved | Reserved                                                 | -                                                                                                                                                                                                                                                                           | -      |
| 8     | OPEM     | OUT packet error interrupt mask.                         | <ul style="list-style-type: none"> <li>0b0: Maskable interrupt</li> <li>0b1: Enable interrupt</li> </ul>                                                                                                                                                                    | RW     |
| 7     | Reserved | Reserved                                                 | -                                                                                                                                                                                                                                                                           | -      |
| 6     | B2BSTUP  | Continuous SETUP packet received interrupt mask          | <ul style="list-style-type: none"> <li>0b0: Maskable interrupt</li> <li>0b1: Enable interrupt</li> </ul> <div>  <b>Note</b><br/>Applicable only to control OUT endpoints.         </div> | RW     |
| 5     | Reserved | Reserved                                                 | -                                                                                                                                                                                                                                                                           | -      |
| 4     | OTEPDM   | OUT token interrupt mask received when endpoint disabled | <ul style="list-style-type: none"> <li>0b0: Maskable interrupt</li> <li>0b1: Enable interrupt</li> </ul> <div>  <b>Note</b><br/>Applicable only to control OUT endpoints.         </div> | RW     |
| 3     | STUPM    | SETUP phase done interrupt mask                          | <ul style="list-style-type: none"> <li>0b0: Maskable interrupt</li> <li>0b1: Enable interrupt</li> </ul> <div>  <b>Note</b><br/>Applicable only to control endpoints.         </div>     | RW     |
| 2     | Reserved | Reserved                                                 | -                                                                                                                                                                                                                                                                           | -      |
| 1     | EPDM     | Endpoint disabled interrupt mask                         | <ul style="list-style-type: none"> <li>0b0: Maskable interrupt</li> <li>0b1: Enable interrupt</li> </ul>                                                                                                                                                                    | RW     |

| Bit | Name | Function                          | Description                                                                                               | Access |
|-----|------|-----------------------------------|-----------------------------------------------------------------------------------------------------------|--------|
| 0   | XFCM | Transfer completed interrupt mask | <ul style="list-style-type: none"><li>● 0b0: Maskable interrupt</li><li>● 0b1: Enable interrupt</li></ul> | RW     |

### 34.6.5.6 Device All Endpoints Interrupt Register (USBHS\_DAINR)

When a active event occurs on an endpoint, the USBHS\_DAINR register interrupts the application via the device OUT endpoint interrupt bit or device IN endpoint interrupt bit (OEPINT or IEPINT bits in USBHS\_GINTSTS, respectively) in the USBHS\_GINTSTS register. Each endpoint corresponds to one interrupt bit, with a maximum of 16 interrupt bits for both OUT and IN endpoints. Bidirectional endpoints will use the corresponding IN and OUT interrupt bits. When the application sets or clears the corresponding bit in the device endpoint x interrupt register (USBHS\_DIEPINTx/USBHS\_DOEPINTx), the corresponding bit in this register is also set or cleared.

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |        |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|--------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0x818  | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15     | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |        |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | OEPINT       |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | IEPINT |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access | RW           |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RW     |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit   | Name   | Function                    | Description                                                                                                                                                                                                                                                                                                                                                                          | Access |
|-------|--------|-----------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:16 | OEPINT | OUT endpoint interrupt bits | Corresponding to OUT endpoints 15 to 0: <ul style="list-style-type: none"> <li>0b0: No interrupt has occurred for this endpoint</li> <li>0b1: Interrupt has occurred for this endpoint</li> </ul> <div>  <b>Note</b><br/>           Each bit corresponds to an OUT endpoint.         </div>       | RW     |
| 15:0  | IEPINT | IN endpoint interrupt bits  | Corresponding to IN Endpoint 15 to Endpoint 0: <ul style="list-style-type: none"> <li>0b0: No interrupt has occurred for this endpoint</li> <li>0b1: Interrupt has occurred for this endpoint</li> </ul> <div>  <b>Note</b><br/>           Each bit corresponds to an IN endpoint.         </div> | RW     |

### 34.6.5.7 Device All Endpoints Interrupt Mask Register (USBHS\_DAINTRSK)

The USBHS\_DAINTRSK register is used in conjunction with the device endpoint interrupt register to interrupt the application when an event occurs on the device endpoint. However, the corresponding bit in the USBHS\_DAINTR register is still set to 1 for this interrupt.

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |         |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0x81C  | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15      | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |         |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | OEPINTM      |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | IEPINTM |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access | RW           |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RW      |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit   | Name    | Function                         | Description                                                                                                                                                                                                                                                                           | Access |
|-------|---------|----------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:16 | OEPINTM | OUT endpoint interrupt mask bits | Corresponding to OUT endpoints 15 to 0:<br>● 0b0: Maskable interrupt<br>● 0b1: Enable interrupt<br><div>  <b>Note</b><br/>           Each bit corresponds to an OUT endpoint.         </div>         | RW     |
| 15:0  | IEPINTM | IN endpoint interrupt mask bits  | Corresponding to IN endpoint 15 to endpoint 0:<br>● 0b0: Maskable interrupt<br>● 0b1: Enable interrupt<br><div>  <b>Note</b><br/>           Each bit corresponds to an IN endpoint.         </div> | RW     |

## 34.6.5.8 Device Threshold Control Register (USBHS\_DTHRCTL)

| Offset | Bit Position |    |    |    |       |          |           |    |    |    |    |    |    |    |         |          |    |    |    |           |    |    |   |   |   |   |   |          |             |   |   |   |
|--------|--------------|----|----|----|-------|----------|-----------|----|----|----|----|----|----|----|---------|----------|----|----|----|-----------|----|----|---|---|---|---|---|----------|-------------|---|---|---|
| 0x830  | 31           | 30 | 29 | 28 | 27    | 26       | 25        | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17      | 16       | 15 | 14 | 13 | 12        | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4        | 3           | 2 | 1 | 0 |
| Reset  | 0x0C100020   |    |    |    |       |          |           |    |    |    |    |    |    |    |         |          |    |    |    |           |    |    |   |   |   |   |   |          |             |   |   |   |
| Name   | Reserved     |    |    |    | ARPEN | Reserved | RXTHRLLEN |    |    |    |    |    |    |    | RXTHREN | Reserved |    |    |    | TXTHRLLEN |    |    |   |   |   |   |   | ISOTHREN | NONISOTHREN |   |   |   |
| Access |              |    |    |    | RW    |          | RW        |    |    |    |    |    |    |    | RW      |          |    |    |    | RW        |    |    |   |   |   |   |   | RW       | RW          |   |   |   |

| Bit   | Name        | Function                                  | Description                                                                                                                                                                                                                                                                                                                                                                                                                            | Access |
|-------|-------------|-------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:28 | Reserved    | Reserved                                  | -                                                                                                                                                                                                                                                                                                                                                                                                                                      | -      |
| 27    | ARPEN       | Arbiter residency enable                  | This bit controls the residency of internal DMA arbitration for IN endpoints. When the threshold is enabled and this bit is set to 1, the DMA retains its arbitration for IN endpoints that receive tokens. This method avoids underflow situations. Residency is enabled by default.                                                                                                                                                  | RW     |
| 26    | Reserved    | Reserved                                  | -                                                                                                                                                                                                                                                                                                                                                                                                                                      | -      |
| 25:17 | RXTHRLLEN   | Receive threshold length                  | This field specifies the size of the receive threshold in double-word units. This field specifies the amount of data the module receives on the USB before initiating a transfer on the AHB. The minimum threshold length is eight double words. It is recommended that RXTHRLLEN matches the set AHB burst transfer length.                                                                                                           | RW     |
| 16    | RXTHREN     | Receive threshold enable                  | When this bit is set to 1, the module enables the threshold for the receive direction.                                                                                                                                                                                                                                                                                                                                                 | RW     |
| 15:11 | Reserved    | Reserved                                  | -                                                                                                                                                                                                                                                                                                                                                                                                                                      | -      |
| 10:2  | TXTHRLLEN   | Transmit threshold length                 | This field specifies the size of the transmit threshold in double-word units. This field specifies the amount of data in the corresponding endpoint transmit FIFO before the module initiates a transfer on the USB. The minimum threshold length is eight double words. This field controls the thresholds for synchronous and asynchronous IN endpoints. It is recommended that TXTHRLLEN matches the set AHB burst transfer length. | RW     |
| 1     | ISOTHREN    | ISO IN endpoint threshold enable          | When this bit is set to 1, the module enables the threshold for synchronous IN endpoints.                                                                                                                                                                                                                                                                                                                                              | RW     |
| 0     | NONISOTHREN | Asynchronous IN endpoint threshold enable | When this bit is set to 1, the module enables the threshold for asynchronous IN endpoints.                                                                                                                                                                                                                                                                                                                                             | RW     |

### 34.6.5.9 Device IN Endpoint FIFO Empty Interrupt Mask Register (USBHS\_DIEPEMPMSK)

This register is used to control the generation of IN endpoint FIFO empty interrupts.

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |           |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |  |  |  |  |  |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-----------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|--|--|--|--|--|--|
| 0x834  | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15        | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |  |  |  |  |  |  |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |           |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |  |  |  |  |  |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | INEPTXFEM |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |  |  |  |  |  |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RW        |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |  |  |  |  |  |

| Bit   | Name      | Function                                | Description                                                                                                                                                                                                                                                                                                                                                                                                 | Access |
|-------|-----------|-----------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:16 | Reserved  | Reserved                                | -                                                                                                                                                                                                                                                                                                                                                                                                           | -      |
| 15:0  | INEPTXFEM | IN EP Tx FIFO empty interrupt mask bits | <p>These bits are used as the mask bits for USBHS_DIEPINTx.</p> <p>Corresponding to IN endpoint 15 to endpoint 0:</p> <ul style="list-style-type: none"> <li>0b0: Maskable interrupt</li> <li>0b1: Enable interrupt</li> </ul> <div>  <b>Note</b><br/> Each bit corresponds to a TXFE interrupt of an IN endpoint. </div> | RW     |



## 34.6.5.10 Device Single Endpoint Interrupt Register (USBHS\_DEACHINT)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |         |          |    |    |    |    |    |   |   |   |   |   |   |   |   |   |         |          |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---------|----------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---------|----------|
| 0x838  | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16      | 15       | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0       |          |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |         |          |    |    |    |    |    |   |   |   |   |   |   |   |   |   |         |          |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    | OEPIINT | Reserved |    |    |    |    |    |   |   |   |   |   |   |   |   |   | IEPIINT | Reserved |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RW      |          |    |    |    |    |    |   |   |   |   |   |   |   |   |   | RW      |          |

| Bit   | Name     | Function                     | Description                                                                                                                                         | Access |
|-------|----------|------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:18 | Reserved | Reserved                     | -                                                                                                                                                   | -      |
| 17    | OEPIINT  | OUT endpoint 1 interrupt bit | <ul style="list-style-type: none"> <li>0b0: No interrupt occurred for OUT endpoint 1</li> <li>0b1: Interrupt occurred for OUT endpoint 1</li> </ul> | RW     |
| 16:2  | Reserved | Reserved                     | -                                                                                                                                                   | -      |
| 1     | IEPIINT  | IN endpoint 1 interrupt bit  | <ul style="list-style-type: none"> <li>0b0: No interrupt occurred for IN endpoint 1</li> <li>0b1: Interrupt occurred for IN endpoint 1</li> </ul>   | RW     |
| 0     | Reserved | Reserved                     | -                                                                                                                                                   | -      |

## 34.6.5.11 Device Single Endpoint Interrupt Mask Register (USBHS\_DEACHINTMASK)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |          |          |    |    |    |    |    |    |   |   |   |   |   |   |   |          |          |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----------|----------|----|----|----|----|----|----|---|---|---|---|---|---|---|----------|----------|---|
| 0x83C  | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17       | 16       | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2        | 1        | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |          |          |    |    |    |    |    |    |   |   |   |   |   |   |   |          |          |   |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    | OEPIINTM | Reserved |    |    |    |    |    |    |   |   |   |   |   |   |   | IEPIINTM | Reserved |   |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    | RW       |          |    |    |    |    |    |    |   |   |   |   |   |   |   | RW       |          |   |

| Bit   | Name     | Function                          | Description                                                                                              | Access |
|-------|----------|-----------------------------------|----------------------------------------------------------------------------------------------------------|--------|
| 31:18 | Reserved | Reserved                          | -                                                                                                        | -      |
| 17    | OEPIINTM | OUT endpoint 1 interrupt mask bit | <ul style="list-style-type: none"> <li>0b0: Maskable interrupt</li> <li>0b1: Enable interrupt</li> </ul> | RW     |
| 16:2  | Reserved | Reserved                          | -                                                                                                        | -      |
| 1     | IEPIINTM | IN endpoint 1 interrupt mask bit  | <ul style="list-style-type: none"> <li>0b0: Maskable interrupt</li> <li>0b1: Enable interrupt</li> </ul> | RW     |
| 0     | Reserved | Reserved                          | -                                                                                                        | -      |

## 34.6.5.12 Device IN Endpoint 1 Interrupt Mask Register (USBHS\_DIEPEACHMSK1)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |      |          |    |    |   |        |          |        |        |          |     |          |      |       |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|------|----------|----|----|---|--------|----------|--------|--------|----------|-----|----------|------|-------|
| 0x844  | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13   | 12       | 11 | 10 | 9 | 8      | 7        | 6      | 5      | 4        | 3   | 2        | 1    | 0     |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |      |          |    |    |   |        |          |        |        |          |     |          |      |       |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | NAKM | Reserved |    |    |   | TXFURM | Reserved | INPNEM | INPNMM | TTXFEMSK | TOM | Reserved | EPDM | XFCRM |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RW   |          |    |    |   | RW     |          | RW     | RW     | RW       | RW  |          | RW   | RW    |

| Bit   | Name     | Function                                              | Description                                                                                              | Access |
|-------|----------|-------------------------------------------------------|----------------------------------------------------------------------------------------------------------|--------|
| 31:14 | Reserved | Reserved                                              | -                                                                                                        | -      |
| 13    | NAKM     | NAK interrupt mask                                    | <ul style="list-style-type: none"> <li>0b0: Maskable interrupt</li> <li>0b1: Enable interrupt</li> </ul> | RW     |
| 12:9  | Reserved | Reserved                                              | -                                                                                                        | -      |
| 8     | TXFURM   | Transmit FIFO underflow interrupt mask                | <ul style="list-style-type: none"> <li>0b0: Maskable interrupt</li> <li>0b1: Enable interrupt</li> </ul> | RW     |
| 7     | Reserved | Reserved                                              | -                                                                                                        | -      |
| 6     | INEPNEM  | IN endpoint NAK active interrupt mask                 | <ul style="list-style-type: none"> <li>0b0: Maskable interrupt</li> <li>0b1: Enable interrupt</li> </ul> | RW     |
| 5     | INEPNMM  | IN token interrupt mask received on EP mismatch       | <ul style="list-style-type: none"> <li>0b0: Maskable interrupt</li> <li>0b1: Enable interrupt</li> </ul> | RW     |
| 4     | TTXFEMSK | IN token interrupt mask received when TxFIFO is empty | <ul style="list-style-type: none"> <li>0b0: Maskable interrupt</li> <li>0b1: Enable interrupt</li> </ul> | RW     |
| 3     | TOM      | Timeout interrupt mask (Non-synchronous endpoint)     | <ul style="list-style-type: none"> <li>0b0: Maskable interrupt</li> <li>0b1: Enable interrupt</li> </ul> | RW     |
| 2     | Reserved | Reserved                                              | -                                                                                                        | -      |
| 1     | EPDM     | Endpoint disabled interrupt mask                      | <ul style="list-style-type: none"> <li>0b0: Maskable interrupt</li> <li>0b1: Enable interrupt</li> </ul> | RW     |
| 0     | XFRM     | Transfer completed interrupt mask                     | <ul style="list-style-type: none"> <li>0b0: Maskable interrupt</li> <li>0b1: Enable interrupt</li> </ul> | RW     |

## 34.6.5.13 Device OUT Endpoint 1 Interrupt Mask Register (USBHS\_DOEPEACHMSK1)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |       |          |    |    |    |   |      |          |         |          |        |       |          |      |       |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-------|----------|----|----|----|---|------|----------|---------|----------|--------|-------|----------|------|-------|
| 0x884  | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14    | 13       | 12 | 11 | 10 | 9 | 8    | 7        | 6       | 5        | 4      | 3     | 2        | 1    | 0     |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |       |          |    |    |    |   |      |          |         |          |        |       |          |      |       |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | NYETM | Reserved |    |    |    |   | OPEM | Reserved | B2BSTUP | Reserved | OTEPDM | STUPM | Reserved | EPDM | XFCRM |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RW    |          |    |    |    |   | RW   |          | RW      |          | RW     | RW    |          | RW   | RW    |

| Bit   | Name     | Function                                                 | Description                                                                                                                                        | Access |
|-------|----------|----------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:15 | Reserved | Reserved                                                 | -                                                                                                                                                  | -      |
| 14    | NYETM    | NYET interrupt mask                                      | <ul style="list-style-type: none"> <li>0b0: Maskable interrupt</li> <li>0b1: Enable interrupt</li> </ul>                                           | RW     |
| 13:9  | Reserved | Reserved                                                 | -                                                                                                                                                  | -      |
| 8     | OPEM     | OUT packet error interrupt mask                          | <ul style="list-style-type: none"> <li>0b0: Maskable interrupt</li> <li>0b1: Enable interrupt</li> </ul>                                           | RW     |
| 7     | Reserved | Reserved                                                 | -                                                                                                                                                  | -      |
| 6     | B2BSTUP  | Continuous SETUP packet received interrupt mask          | <ul style="list-style-type: none"> <li>0b0: Maskable interrupt</li> <li>0b1: Enable interrupt</li> </ul> Applicable only to control OUT endpoints. | RW     |
| 5     | Reserved | Reserved                                                 | -                                                                                                                                                  | -      |
| 4     | OTEPDM   | OUT token interrupt mask received when endpoint disabled | <ul style="list-style-type: none"> <li>0b0: Maskable interrupt</li> <li>0b1: Enable interrupt</li> </ul> Applicable only to control OUT endpoints. | RW     |
| 3     | STUPM    | SETUP phase done interrupt mask                          | <ul style="list-style-type: none"> <li>0b0: Maskable interrupt</li> <li>0b1: Enable interrupt</li> </ul> Applicable only to control endpoints.     | RW     |
| 2     | Reserved | Reserved                                                 | -                                                                                                                                                  | -      |
| 1     | EPDM     | Endpoint disabled interrupt mask                         | <ul style="list-style-type: none"> <li>0b0: Maskable interrupt</li> <li>0b1: Enable interrupt</li> </ul>                                           | RW     |
| 0     | XFRM     | Transfer completed interrupt mask                        | <ul style="list-style-type: none"> <li>0b0: Maskable interrupt</li> <li>0b1: Enable interrupt</li> </ul>                                           | RW     |

## 34.6.5.14 Device IN Endpoint 0 Control Register (USBHS\_DIEPCTL0)

This register is used to control control transfer endpoint 0.

| Offset | Bit Position |       |          |      |      |        |    |    |    |    |       |          |       |        |          |        |          |    |    |    |    |    |   |   |   |   |       |   |   |   |   |   |
|--------|--------------|-------|----------|------|------|--------|----|----|----|----|-------|----------|-------|--------|----------|--------|----------|----|----|----|----|----|---|---|---|---|-------|---|---|---|---|---|
| 0x900  | 31           | 30    | 29       | 28   | 27   | 26     | 25 | 24 | 23 | 22 | 21    | 20       | 19    | 18     | 17       | 16     | 15       | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5     | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00008000   |       |          |      |      |        |    |    |    |    |       |          |       |        |          |        |          |    |    |    |    |    |   |   |   |   |       |   |   |   |   |   |
| Name   | EPENA        | EPDIS | Reserved | SNAK | CNAK | TXFNUM |    |    |    |    | STALL | Reserved | EPTYP | NAKSTS | Reserved | USBAEP | Reserved |    |    |    |    |    |   |   |   |   | MPSIZ |   |   |   |   |   |
| Access | RW           | RW    |          | RW   | RW   | RW     |    |    |    |    | RW    |          | RO    | RO     |          | RO     |          |    |    |    |    |    |   |   |   |   | RW    |   |   |   |   |   |

| Bit   | Name     | Function         | Description                                                                                                                                                                                                                                                                                                                                                                                                                                       | Access |
|-------|----------|------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31    | EPENA    | Endpoint enable  | The application sets this bit to 1 to initiate a data transfer on endpoint 0.<br>The module clears this bit before any of the following interrupts are triggered on this endpoint: <ul style="list-style-type: none"> <li>● SETUP phase done</li> <li>● Endpoint disable</li> <li>● Transmission completion</li> </ul>                                                                                                                            | RW     |
| 30    | EPDIS    | Endpoint disable | The application can set this bit to 1 even before the transfer on this endpoint is completed to stop data transmission on the endpoint. The application must wait for an endpoint disable interrupt to occur before the endpoint is considered disabled. The module clears this bit before the endpoint disable interrupt bit is set to 1. The application can set this bit to 1 only after the endpoint enable bit of this endpoint is set to 1. | RW     |
| 29:28 | Reserved | Reserved         | -                                                                                                                                                                                                                                                                                                                                                                                                                                                 | -      |
| 27    | SNAK     | Set NAK          | Writing to this bit sets the NAK bit of the endpoint to 1.<br>With this bit, the application can control the sending of NAK handshake signals on the endpoint. The module can also set this bit of the endpoint to 1 after the endpoint receives the SETUP packet.                                                                                                                                                                                | RW     |
| 26    | CNAK     | Clear NAK        | Writing to this bit clears the NAK bit of the endpoint.                                                                                                                                                                                                                                                                                                                                                                                           | RW     |
| 25:22 | TXFNUM   | TxFIFO number    | This value is set to the FIFO number allocated to IN endpoint 0. Only TX-FIFO0 can be used.                                                                                                                                                                                                                                                                                                                                                       | RW     |
| 21    | STALL    | STALL handshake  | The application can only set this bit to 1. When the endpoint receives a SETUP token, the module clears this bit. If the NAK bit, global IN NAK, or global OUT NAK and this bit are all set to 1, the STALL bit takes precedence.                                                                                                                                                                                                                 | RW     |
| 20    | Reserved | Reserved         | -                                                                                                                                                                                                                                                                                                                                                                                                                                                 | -      |

| Bit   | Name            | Function            | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                              | Access |
|-------|-----------------|---------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 19:18 | EPTYP           | Endpoint type       | Hardware setting is '0b00', indicating a control-type endpoint.                                                                                                                                                                                                                                                                                                                                                                                                          | RO     |
| 17    | NAKSTS          | NAK status          | <ul style="list-style-type: none"> <li>● 0b0: The module replies with a non-NAK handshake based on FIFO status</li> <li>● 0b1: The module replies with a NAK handshake on this endpoint</li> </ul> <p>When this bit is set to 1 (either by the application or the module), the module stops sending data even if there is still data available in the Tx FIFO. Regardless of how this bit is set, the module always responds to SETUP packets with an ACK handshake.</p> | RO     |
| 16    | <i>Reserved</i> | <i>Reserved</i>     | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | -      |
| 15    | USBAEP          | USB active endpoint | This bit is always set to 1, indicating that control endpoint 0 is always active in all configurations and interfaces.                                                                                                                                                                                                                                                                                                                                                   | RO     |
| 14:2  | <i>Reserved</i> | <i>Reserved</i>     | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | -      |
| 1:0   | MPSIZ           | Maximum packet size | <p>The application must program this field to the maximum packet size of the current logical endpoint.</p> <ul style="list-style-type: none"> <li>● 0b00: 64 bytes</li> <li>● 0b01: 32 bytes</li> <li>● 0b10: 16 bytes</li> <li>● 0b11: 8 bytes</li> </ul>                                                                                                                                                                                                               | RW     |

### 34.6.5.15 Device IN Endpoint x Control Register (USBHS\_DIEPCTLx) (x=1-15)

Applications use this register to control the behavior of individual logical endpoints (except endpoint 0).

Offset: 0x900+0x20\*x

| Offset | Bit Position |       |         |                |      |      |        |    |    |    |       |          |       |        |            |        |          |    |    |    |       |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|-------|---------|----------------|------|------|--------|----|----|----|-------|----------|-------|--------|------------|--------|----------|----|----|----|-------|----|---|---|---|---|---|---|---|---|---|---|
| -      | 31           | 30    | 29      | 28             | 27   | 26   | 25     | 24 | 23 | 22 | 21    | 20       | 19    | 18     | 17         | 16     | 15       | 14 | 13 | 12 | 11    | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |       |         |                |      |      |        |    |    |    |       |          |       |        |            |        |          |    |    |    |       |    |   |   |   |   |   |   |   |   |   |   |
| Name   | EPENA        | EPDIS | SODDFRM | SD0PID/SEVNFRM | SNAK | CNAK | TXFNUM |    |    |    | STALL | Reserved | EPTYP | NAKSTS | EONUM/DPID | USBAEP | Reserved |    |    |    | MPSIZ |    |   |   |   |   |   |   |   |   |   |   |
| Access | RW           | RW    | RW      | RW             | RW   | RW   | RW     |    |    |    | RW    |          | RO    | RO     | RO         | RO     |          |    |    |    | RW    |    |   |   |   |   |   |   |   |   |   |   |

| Bit | Name           | Function                     | Description                                                                                                                                                                                                                                                                                                                                                                                                                                       | Access |
|-----|----------------|------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31  | EPENA          | Endpoint enable              | The application sets this bit to initiate data sending on the endpoint. The module clears this bit before any of the following interrupts are triggered on this endpoint: <ul style="list-style-type: none"> <li>● Endpoint disabled</li> <li>● Transmission completion</li> </ul>                                                                                                                                                                | RW     |
| 30  | EPDIS          | Endpoint disable             | The application can set this bit to 1 even before the transfer on this endpoint is completed to stop data transmission on the endpoint. The application must wait for an endpoint disable interrupt to occur before the endpoint is considered disabled. The module clears this bit before the endpoint disable interrupt bit is set to 1. The application can set this bit to 1 only after the endpoint enable bit of this endpoint is set to 1. | RW     |
| 29  | SODDFRM        | Set odd frame                | Applies only to synchronous IN and OUT endpoints.<br>Writing to this field sets the even/odd frame (EONUM) field to odd frames.                                                                                                                                                                                                                                                                                                                   | RW     |
| 28  | SD0PID/SEVNFRM | Set DATA0 PID/Set even frame | <b>SD0PID:</b> Applicable only to interrupt/bulk IN endpoints. Writing to this field sets the endpoint data PID (DPID) field in this register to DATA0.<br><b>SEVNFRM:</b> Applicable only to synchronous IN endpoints. Writing to this field sets the even/odd frames (EONUM) field to even frames.                                                                                                                                              | RW     |
| 27  | SNAK           | Set NAK                      | Writing to this bit sets the NAK bit of the endpoint to 1.<br>With this bit, the application can control the sending of NAK handshake signals on the endpoint. The module can also set this bit of the IN endpoint to 1 when a transfer complete interrupt occurs or after SETUP is received on the endpoint.                                                                                                                                     | RW     |

| Bit   | Name            | Function                         | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | Access |
|-------|-----------------|----------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 26    | CNAK            | Clear NAK                        | Writing to this bit clears the NAK bit of the endpoint.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | RW     |
| 25:22 | TXFNUM          | TxFIFO number                    | These bits are used to specify the FIFO number associated with this endpoint. A separate FIFO number must be set for each active IN endpoint. This field is only active for IN endpoints.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | RW     |
| 21    | STALL           | STALL handshake                  | Setting this bit by the application causes the device to reply STALL to all tokens from the USB host. If the NAK bit, global IN NAK, or global OUT NAK is set at the same time as this bit, the STALL bit takes precedence. Only the application can clear this bit, not the module.                                                                                                                                                                                                                                                                                                                                                                                                                                        | RW     |
| 20    | <i>Reserved</i> | <i>Reserved</i>                  | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | -      |
| 19:18 | EPTYP           | Endpoint type                    | <ul style="list-style-type: none"> <li>0b00: Control</li> <li>0b01: Isochronous</li> <li>0b10: Bulk</li> <li>0b11: Interrupt</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | RO     |
| 17    | NAKSTS          | NAK status                       | <ul style="list-style-type: none"> <li>0b0: The module replies with a non-NAK handshake based on FIFO status</li> <li>0b1: The module replies with a NAK handshake on this endpoint</li> </ul> <p>When an application or module sets this bit:</p> <ul style="list-style-type: none"> <li>For non-synchronous IN endpoints: The module stops transmitting any data through the IN endpoint even if there is data available in the TxFIFO.</li> <li>For synchronous IN endpoints: The module transmits zero-length packets even if there is data available in the TxFIFO.</li> </ul> <p>Regardless of how this bit is set, the module always responds to SETUP packets with an ACK handshake.</p>                            | RO     |
| 16    | EONUM/<br>DPID  | Even/Odd frame/Endpoint data PID | <p><b>EONUM:</b><br/>Applies only to synchronous IN endpoints. Indicates the number of the frame in which the module sends/receives synchronized data for this endpoint. The application must program the even/odd frame number through the SEVNFRM and SODDFRM fields in this register for this endpoint to send/receive isochronous data.</p> <ul style="list-style-type: none"> <li>0b0: Even frame</li> <li>0b1: Odd frame</li> </ul> <p><b>DPID:</b><br/>Applies to interrupt/bulk IN endpoints only. Contains the PID of packets that will be received or transmitted on this endpoint. After an endpoint is activated, the application must program the PID of the first packet to be received or transmitted on</p> | RO     |

| Bit   | Name     | Function            | Description                                                                                                                                                                                                                                                                                                                                   | Access |
|-------|----------|---------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
|       |          |                     | this endpoint. The application programs the DATA0 or DATA1 PID using the SD0PID register field.<br><ul style="list-style-type: none"> <li>● 0b0: DATA0</li> <li>● 0b1: DATA1</li> </ul>                                                                                                                                                       |        |
| 15    | USBAEP   | USB active endpoint | Indicates whether this endpoint is active in the current configuration and interface. The module clears this bit to 0 for all endpoints except endpoint 0 when a USB reset is detected. After receiving the SetConfiguration and SetInterface commands, the application must program the endpoint register accordingly and set this bit to 1. | RO     |
| 14:11 | Reserved | Reserved            | -                                                                                                                                                                                                                                                                                                                                             | -      |
| 10:0  | MPSIZ    | Maximum packet size | The application must program this field to the maximum packet size of the current logical endpoint. This value is in bytes.                                                                                                                                                                                                                   | RW     |



### 34.6.5.16 Device IN Endpoint x Interrupt Register (USBHS\_DIEPINTx) (x=0-15)

This register indicates the status of the endpoint when USB and AHB related events occur. When the IN endpoint interrupt bit (IEPINT bit in USBHS\_GINTSTS) in the module interrupt register is set to 1, the application must read this register. Before the application can read this register, it must first read the device all-endpoint interrupt (USBHS\_DAINTE) register to obtain the correct endpoint number of the device endpoint x interrupt register. The application must clear the corresponding bit in this register in order to clear the corresponding bits in the USBHS\_DAINTE and USBHS\_GINTSTS registers.

Offset: 0x908+0x20\*x

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |     |    |      |   |           |   |          |   |            |   |      |   |        |  |          |  |       |  |     |  |          |  |        |  |      |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-----|----|------|---|-----------|---|----------|---|------------|---|------|---|--------|--|----------|--|-------|--|-----|--|----------|--|--------|--|------|
| -      | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12  | 11 | 10   | 9 | 8         | 7 | 6        | 5 | 4          | 3 | 2    | 1 | 0      |  |          |  |       |  |     |  |          |  |        |  |      |
| Reset  | 0x00000080   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |     |    |      |   |           |   |          |   |            |   |      |   |        |  |          |  |       |  |     |  |          |  |        |  |      |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | NAK |    | BERR |   | PKTDRPSTS |   | Reserved |   | TXFIFOUDRN |   | TXFE |   | INEPNE |  | Reserved |  | TTXFE |  | TOC |  | Reserved |  | EPDISD |  | XFRC |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RW  |    | RW   |   | RW        |   |          |   | RW         |   | RO   |   | RW     |  |          |  | RW    |  | RW  |  |          |  | RW     |  | RW   |

| Bit   | Name        | Function                  | Description                                                                                                                                                                                                                                                         | Access |
|-------|-------------|---------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:14 | Reserved    | Reserved                  | -                                                                                                                                                                                                                                                                   | -      |
| 13    | NAK         | NAK interrupt             | The module generates this interrupt when the device transmits or receives a NAK.<br>For synchronous IN endpoints, this interrupt is also generated when a zero-length data packet is sent due to no data available in the Tx FIFO.<br>Write 1 to clear by software. | RW     |
| 12    | BERR        | Babble error interruption | Write 1 to clear by software.                                                                                                                                                                                                                                       | RW     |
| 11    | PKTDRPSTS   | Packet drop status        | This bit indicates to the application that an ISOC OUT data packet has been discarded. This bit has no corresponding interrupt mask bit and does not generate an interrupt.<br>Write 1 to clear by software.                                                        | RW     |
| 10:9  | Reserved    | Reserved                  | -                                                                                                                                                                                                                                                                   | -      |
| 8     | TXFI-FOUDRN | Transmit FIFO underflow.  | The module generates this interrupt when it detects an underflow in the transmit FIFO of this endpoint.<br>Relevance: This interrupt is enable only when the threshold is enabled.<br>Write 1 to clear by software.                                                 | RW     |
| 7     | TXFE        | Transmit FIFO empty       | This interrupt is asserted when the Tx FIFO for this endpoint is half or fully empty. Whether the Tx FIFO is in half empty or full empty status is determined by the Tx FIFO empty level bit (TXFELVL bit in USBHS_GAHBCFG) in the USBHS_GAHBCFG register.          | RO     |

| Bit | Name            | Function                               | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | Access |
|-----|-----------------|----------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 6   | INEPNE          | IN endpoint NAK effective              | <p>This bit can be cleared when the application clears the IN endpoint NAK by writing to the CNAK bit in USBHS_DIEPCTLx.</p> <p>This interrupt indicates that the module has sampled the NAK set (by the application or the module), and the result has taken effect. This interrupt indicates that the IN endpoint NAK bit set by the application program has acted in the module.</p> <p>This interrupt does not guarantee that a NAK handshake was sent on the USB.</p> <p>The STALL bit has priority over the NAK bit.</p> <p>Writing 1 to this bit by software can also clear it.</p> | RW     |
| 5   | <i>Reserved</i> | <i>Reserved</i>                        | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | -      |
| 4   | TTXFE           | IN token received when TxFIFO is empty | <p>Applies only to non-periodic IN endpoints.</p> <p>When the TxFIFO (periodic/non-periodic) corresponding to this endpoint is empty and an IN token is received, an interrupt is generated.</p> <p>Write 1 to clear by software.</p>                                                                                                                                                                                                                                                                                                                                                      | RW     |
| 3   | TOC             | Timeout condition                      | <p>Applies to control IN endpoints only.</p> <p>Indicates that this endpoint timed out on the most recently received IN token response.</p> <p>Write 1 to clear by software.</p>                                                                                                                                                                                                                                                                                                                                                                                                           | RW     |
| 2   | <i>Reserved</i> | <i>Reserved</i>                        | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | -      |
| 1   | EPDISD          | Endpoint disabled interrupt            | <p>This bit indicates that the endpoint has been disabled by the application.</p> <p>Write 1 to clear by software.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | RW     |
| 0   | XFRC            | Transfer completed interrupt           | <p>This field indicates that transfers set on this endpoint have completed transfers on USB and AHB.</p> <p>Write 1 to clear by software.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                              | RW     |

### 34.6.5.17 Device IN Endpoint 0 Transfer Size Register (USBHS\_DIEPTSZ0)

The application must modify this register before enabling endpoint 0. After enabling endpoint 0 by setting the endpoint enable bit (EPENA in USBHS\_DIEPCTL0) in the device control endpoint 0 control register, the module modifies this register. The application can read this register only after the module has cleared the endpoint enable bit.

Non-zero endpoints use registers for endpoints 1-15.

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |        |          |    |    |    |    |    |    |    |    |   |   |   |        |   |   |   |   |   |   |  |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|--------|----------|----|----|----|----|----|----|----|----|---|---|---|--------|---|---|---|---|---|---|--|
| 0x910  | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19     | 18       | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6      | 5 | 4 | 3 | 2 | 1 | 0 |  |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |        |          |    |    |    |    |    |    |    |    |   |   |   |        |   |   |   |   |   |   |  |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    | PKTCNT | Reserved |    |    |    |    |    |    |    |    |   |   |   | XFRSIZ |   |   |   |   |   |   |  |
| Access |              |    |    |    |    |    |    |    |    |    |    |    | RW     |          |    |    |    |    |    |    |    |    |   |   |   | RW     |   |   |   |   |   |   |  |

| Bit   | Name     | Function      | Description                                                                                                                                                                                                                                                                                                                                                                                   | Access |
|-------|----------|---------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:21 | Reserved | Reserved      | -                                                                                                                                                                                                                                                                                                                                                                                             | -      |
| 20:19 | PKTCNT   | Packet count  | Indicates the number of data packets included in one data transfer for endpoint 0.<br>This field will be decremented each time a packet (max size or short) is read from the TxFIFO.                                                                                                                                                                                                          | RW     |
| 18:7  | Reserved | Reserved      | -                                                                                                                                                                                                                                                                                                                                                                                             | -      |
| 6:0   | XFRSIZ   | Transfer size | Indicates the amount of data included in one data transfer for endpoint 0, in bytes. The module interrupts the application only after the application has transferred this data. The transfer size can be set to the endpoint's maximum packet size to break at the end of each packet.<br>The module decrements this field each time a packet from external memory is written to the TxFIFO. | RW     |

### 34.6.5.18 Device IN Endpoint x Transfer Size Register (USBHS\_DIEPTSIZE<sub>x</sub>) (x=1-15)

The application must modify this register before enabling the endpoint. After the endpoint is enabled via the endpoint enable bit (EPENA bit in USBHS\_DIEPCTL<sub>x</sub>) in the USBHS\_DIEPCTL<sub>x</sub> register, the module modifies this register. The application can only read this register after the module clears the endpoint enable bit.

Offset: 0x910+0x20\*x

| Offset | Bit Position |      |    |        |    |    |    |    |    |    |    |    |    |        |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |
|--------|--------------|------|----|--------|----|----|----|----|----|----|----|----|----|--------|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|--|
| -      | 31           | 30   | 29 | 28     | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18     | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |  |
| Reset  | 0x00000000   |      |    |        |    |    |    |    |    |    |    |    |    |        |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |
| Name   | Reserved     | MCNT |    | PKTCNT |    |    |    |    |    |    |    |    |    | XFRSIZ |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |
| Access |              | RW   |    | RW     |    |    |    |    |    |    |    |    |    | RW     |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |

| Bit   | Name     | Function      | Description                                                                                                                                                                                                                                                                                                                                                                                                   | Access |
|-------|----------|---------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31    | Reserved | Reserved      | -                                                                                                                                                                                                                                                                                                                                                                                                             | -      |
| 30:29 | MCNT     | Multi count   | For periodic IN endpoints, this field indicates the number of packets that must be sent per frame on the USB. The module uses this field to calculate the data PID for synchronous IN endpoints. <ul style="list-style-type: none"> <li>● 0b00: Setting prohibited</li> <li>● 0b01: 1 packet</li> <li>● 0b10: 2 packets</li> <li>● 0b11: 3 packets</li> </ul>                                                 | RW     |
| 28:19 | PKTCNT   | Packet count  | Indicates the number of data packets included in one data transfer on this endpoint. This field will be decremented each time a packet (max size or short) is read from the TxFIFO.                                                                                                                                                                                                                           | RW     |
| 18:0  | XFRSIZ   | Transfer size | This field contains the amount of data included in one data transfer of the current endpoint, in bytes. The module interrupts the application only after the application has transferred this data. The transfer size can be set to the endpoint's maximum packet size to break at the end of each packet. The module decrements this field each time a packet from external memory is written to the TxFIFO. | RW     |

### 34.6.5.19 Device IN Endpoint x DMA Address Register (USBHS\_DIEPDMAx) (x=0-15)

This register is used to set the DMA address when the device endpoint is in DMA mode.

Offset: 0x914+0x20\*x

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| -      | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0xFFFFFFFF   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | DMAADDR      |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access | RW           |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit  | Name    | Function    | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | Access |
|------|---------|-------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:0 | DMAADDR | DMA address | <p>This bit contains the starting address of the external storage area when data is stored on the endpoint using DMA.</p> <div>  <b>Notice</b><br/>           For control endpoints, the storage area pointed to by this field is also used to store control OUT data packets and SETUP transaction data packets. When three or more SETUP data packets are received consecutively, the SETUP data packets in the memory are overwritten. This register is incremented for every AHB transfer. The application must set a double word aligned address.         </div> | RW     |

## 34.6.5.20 Device IN Endpoint x Transmit FIFO Status Register (USBHS\_DTXFSTSx) (x=0-15)

Offset: 0x918+0x20\*x

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |           |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-----------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| -      | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15        | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000800   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |           |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | INEPTFSAV |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RO        |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit   | Name      | Function                           | Description                                                                                                                                                                                                                                                                                                                                        | Access |
|-------|-----------|------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:16 | Reserved  | Reserved                           | -                                                                                                                                                                                                                                                                                                                                                  | -      |
| 15:0  | INEPTFSAV | IN endpoint TxFIFO space available | Indicates the available free space size in the end-point TxFIFO. In 32-bit words: <ul style="list-style-type: none"> <li>● 0: The endpoint TxFIFO is full.</li> <li>● 1: 1 word available</li> <li>● 2: 2 words available</li> <li>● .....</li> <li>● n: n words are available (n is the value of the register USBHS_DIEPTFx.INEPTXFD).</li> </ul> | RO     |

## 34.6.5.21 Device OUT Endpoint 0 Control Register (USBHS\_DOEPTL0)

This register is used to control control transfer endpoint 0.

| Offset | Bit Position |       |          |      |      |          |    |    |    |       |      |       |        |          |        |          |    |    |    |    |    |    |   |   |   |       |   |   |   |   |   |   |
|--------|--------------|-------|----------|------|------|----------|----|----|----|-------|------|-------|--------|----------|--------|----------|----|----|----|----|----|----|---|---|---|-------|---|---|---|---|---|---|
| 0xB00  | 31           | 30    | 29       | 28   | 27   | 26       | 25 | 24 | 23 | 22    | 21   | 20    | 19     | 18       | 17     | 16       | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6     | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00008000   |       |          |      |      |          |    |    |    |       |      |       |        |          |        |          |    |    |    |    |    |    |   |   |   |       |   |   |   |   |   |   |
| Name   | EPENA        | EPDIS | Reserved | SNAK | CNAK | Reserved |    |    |    | STALL | SNPM | EPTYP | NAKSTS | Reserved | USBAEP | Reserved |    |    |    |    |    |    |   |   |   | MPSIZ |   |   |   |   |   |   |
| Access | RW           | RW    |          | RW   | RW   |          |    |    |    | RW    | RW   | RO    | RO     |          | RO     |          |    |    |    |    |    |    |   |   |   | RW    |   |   |   |   |   |   |

| Bit   | Name     | Function         | Description                                                                                                                                                                                                                                                                                                            | Access |
|-------|----------|------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31    | EPENA    | Endpoint enable  | The application sets this bit to 1 to initiate a data transfer on endpoint 0.<br>The module clears this bit before any of the following interrupts are triggered on this endpoint: <ul style="list-style-type: none"> <li>● SETUP phase done</li> <li>● Endpoint disable</li> <li>● Transmission completion</li> </ul> | RW     |
| 30    | EPDIS    | Endpoint disable | The application cannot disable control OUT endpoint 0.                                                                                                                                                                                                                                                                 | RW     |
| 29:28 | Reserved | Reserved         | -                                                                                                                                                                                                                                                                                                                      | -      |
| 27    | SNAK     | Set NAK          | Writing to this bit sets the NAK bit of the endpoint to 1.<br>With this bit, the application can control the sending of NAK handshake signals on the endpoint.<br>The module can also set this bit of the endpoint to 1 after the endpoint receives the SETUP packet.                                                  | RW     |
| 26    | CNAK     | Clear NAK        | Writing to this bit clears the NAK bit of the endpoint.                                                                                                                                                                                                                                                                | RW     |
| 25:22 | Reserved | Reserved         | -                                                                                                                                                                                                                                                                                                                      | -      |
| 21    | STALL    | STALL handshake  | The application can only set this bit when the endpoint receives a SETUP token, and the module clears it. If the NAK bit, global OUT NAK, and this bit are all set to 1, the STALL bit takes precedence. Regardless of how this bit is set, the module always responds to SETUP packets with an ACK handshake.         | RW     |
| 20    | SNPM     | Snoop mode       | This bit is used to configure the endpoint in listen mode. In listen mode, the module does not check the OUT packet for correctness before transferring it to the application memory.                                                                                                                                  | RW     |
| 19:18 | EPTYP    | Endpoint type    | Hardware setting is '0b00', indicating a control-type endpoint.                                                                                                                                                                                                                                                        | RO     |

| Bit  | Name            | Function            | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | Access |
|------|-----------------|---------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 17   | NAKSTS          | NAK status          | <ul style="list-style-type: none"> <li>● 0b0: The module replies with a non-NAK handshake based on FIFO status</li> <li>● 0b1: The module replies with a NAK handshake on this endpoint</li> </ul> <p>When the application or module sets this bit to 1, the module stops receiving data even if there is space in the RxFIFO to accommodate the received packet. Regardless of how this bit is set, the module always responds to SETUP packets with an ACK handshake.</p> | RO     |
| 16   | <i>Reserved</i> | <i>Reserved</i>     | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | -      |
| 15   | USBAEP          | USB active endpoint | This bit is always set to 1, indicating that control endpoint 0 is always active in all configurations and interfaces.                                                                                                                                                                                                                                                                                                                                                      | RO     |
| 14:2 | <i>Reserved</i> | <i>Reserved</i>     | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | -      |
| 1:0  | MPSIZ           | Maximum packet size | <p>The maximum packet size of control OUT endpoint 0 is the same as the value programmed in control IN endpoint 0.</p> <ul style="list-style-type: none"> <li>● 0b00: 64 bytes</li> <li>● 0b01: 32 bytes</li> <li>● 0b10: 16 bytes</li> <li>● 0b11: 8 bytes</li> </ul>                                                                                                                                                                                                      | RW     |



### 34.6.5.22 Device OUT Endpoint x Control Register (USBHS\_DOEPCTLx) (x=1-15)

Applications use this register to control the behavior of individual logical endpoints (except Endpoint 0).

Offset: 0xB00+0x20\*x

| Offset | Bit Position |       |                |                |      |      |          |    |    |    |       |      |       |        |            |        |          |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |
|--------|--------------|-------|----------------|----------------|------|------|----------|----|----|----|-------|------|-------|--------|------------|--------|----------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|--|
| -      | 31           | 30    | 29             | 28             | 27   | 26   | 25       | 24 | 23 | 22 | 21    | 20   | 19    | 18     | 17         | 16     | 15       | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |  |
| Reset  | 0x00000000   |       |                |                |      |      |          |    |    |    |       |      |       |        |            |        |          |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |
| Name   | EPENA        | EPDIS | SDIPID/SODDFRM | SDOPID/SEVNFRM | SNAK | CNAK | Reserved |    |    |    | STALL | SNPM | EPTYP | NAKSTS | EONUM/DPID | USBAEP | Reserved |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |
| Access | RW           | RW    | RW             | RW             | RW   | RW   |          |    |    |    |       | RW   | RW    | RO     | RO         | RO     |          | RO |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |
|        |              |       |                |                |      |      |          |    |    |    |       |      |       |        |            |        |          |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |
|        |              |       |                |                |      |      |          |    |    |    |       |      |       |        |            |        |          |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |
|        |              |       |                |                |      |      |          |    |    |    |       |      |       |        |            |        |          |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |
|        |              |       |                |                |      |      |          |    |    |    |       |      |       |        |            |        |          |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |
|        |              |       |                |                |      |      |          |    |    |    |       |      |       |        |            |        |          |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |
|        |              |       |                |                |      |      |          |    |    |    |       |      |       |        |            |        |          |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |
|        |              |       |                |                |      |      |          |    |    |    |       |      |       |        |            |        |          |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |
|        |              |       |                |                |      |      |          |    |    |    |       |      |       |        |            |        |          |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |
|        |              |       |                |                |      |      |          |    |    |    |       |      |       |        |            |        |          |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |
|        |              |       |                |                |      |      |          |    |    |    |       |      |       |        |            |        |          |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |
|        |              |       |                |                |      |      |          |    |    |    |       |      |       |        |            |        |          |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |
|        |              |       |                |                |      |      |          |    |    |    |       |      |       |        |            |        |          |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |
|        |              |       |                |                |      |      |          |    |    |    |       |      |       |        |            |        |          |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |
|        |              |       |                |                |      |      |          |    |    |    |       |      |       |        |            |        |          |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |
|        |              |       |                |                |      |      |          |    |    |    |       |      |       |        |            |        |          |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |
|        |              |       |                |                |      |      |          |    |    |    |       |      |       |        |            |        |          |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |
|        |              |       |                |                |      |      |          |    |    |    |       |      |       |        |            |        |          |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |
|        |              |       |                |                |      |      |          |    |    |    |       |      |       |        |            |        |          |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |
|        |              |       |                |                |      |      |          |    |    |    |       |      |       |        |            |        |          |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |
|        |              |       |                |                |      |      |          |    |    |    |       |      |       |        |            |        |          |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |
|        |              |       |                |                |      |      |          |    |    |    |       |      |       |        |            |        |          |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |
|        |              |       |                |                |      |      |          |    |    |    |       |      |       |        |            |        |          |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |
|        |              |       |                |                |      |      |          |    |    |    |       |      |       |        |            |        |          |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |
|        |              |       |                |                |      |      |          |    |    |    |       |      |       |        |            |        |          |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |
|        |              |       |                |                |      |      |          |    |    |    |       |      |       |        |            |        |          |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |
|        |              |       |                |                |      |      |          |    |    |    |       |      |       |        |            |        |          |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |
|        |              |       |                |                |      |      |          |    |    |    |       |      |       |        |            |        |          |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |
|        |              |       |                |                |      |      |          |    |    |    |       |      |       |        |            |        |          |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |
|        |              |       |                |                |      |      |          |    |    |    |       |      |       |        |            |        |          |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |
|        |              |       |                |                |      |      |          |    |    |    |       |      |       |        |            |        |          |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |
|        |              |       |                |                |      |      |          |    |    |    |       |      |       |        |            |        |          |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |
|        |              |       |                |                |      |      |          |    |    |    |       |      |       |        |            |        |          |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |
|        |              |       |                |                |      |      |          |    |    |    |       |      |       |        |            |        |          |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |
|        |              |       |                |                |      |      |          |    |    |    |       |      |       |        |            |        |          |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |
|        |              |       |                |                |      |      |          |    |    |    |       |      |       |        |            |        |          |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |
|        |              |       |                |                |      |      |          |    |    |    |       |      |       |        |            |        |          |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |
|        |              |       |                |                |      |      |          |    |    |    |       |      |       |        |            |        |          |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |
|        |              |       |                |                |      |      |          |    |    |    |       |      |       |        |            |        |          |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |
|        |              |       |                |                |      |      |          |    |    |    |       |      |       |        |            |        |          |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |
|        |              |       |                |                |      |      |          |    |    |    |       |      |       |        |            |        |          |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |
|        |              |       |                |                |      |      |          |    |    |    |       |      |       |        |            |        |          |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |
|        |              |       |                |                |      |      |          |    |    |    |       |      |       |        |            |        |          |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |
|        |              |       |                |                |      |      |          |    |    |    |       |      |       |        |            |        |          |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |
|        |              |       |                |                |      |      |          |    |    |    |       |      |       |        |            |        |          |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |
|        |              |       |                |                |      |      |          |    |    |    |       |      |       |        |            |        |          |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |
|        |              |       |                |                |      |      |          |    |    |    |       |      |       |        |            |        |          |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |
|        |              |       |                |                |      |      |          |    |    |    |       |      |       |        |            |        |          |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |
|        |              |       |                |                |      |      |          |    |    |    |       |      |       |        |            |        |          |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |
|        |              |       |                |                |      |      |          |    |    |    |       |      |       |        |            |        |          |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |
|        |              |       |                |                |      |      |          |    |    |    |       |      |       |        |            |        |          |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |
|        |              |       |                |                |      |      |          |    |    |    |       |      |       |        |            |        |          |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |
|        |              |       |                |                |      |      |          |    |    |    |       |      |       |        |            |        |          |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |
|        |              |       |                |                |      |      |          |    |    |    |       |      |       |        |            |        |          |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |
|        |              |       |                |                |      |      |          |    |    |    |       |      |       |        |            |        |          |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |
|        |              |       |                |                |      |      |          |    |    |    |       |      |       |        |            |        |          |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |
|        |              |       |                |                |      |      |          |    |    |    |       |      |       |        |            |        |          |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |
|        |              |       |                |                |      |      |          |    |    |    |       |      |       |        |            |        |          |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |
|        |              |       |                |                |      |      |          |    |    |    |       |      |       |        |            |        |          |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |
|        |              |       |                |                |      |      |          |    |    |    |       |      |       |        |            |        |          |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |
|        |              |       |                |                |      |      |          |    |    |    |       |      |       |        |            |        |          |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |
|        |              |       |                |                |      |      |          |    |    |    |       |      |       |        |            |        |          |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |
|        |              |       |                |                |      |      |          |    |    |    |       |      |       |        |            |        |          |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |
|        |              |       |                |                |      |      |          |    |    |    |       |      |       |        |            |        |          |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |
|        |              |       |                |                |      |      |          |    |    |    |       |      |       |        |            |        |          |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |
|        |              |       |                |                |      |      |          |    |    |    |       |      |       |        |            |        |          |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |
|        |              |       |                |                |      |      |          |    |    |    |       |      |       |        |            |        |          |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |
|        |              |       |                |                |      |      |          |    |    |    |       |      |       |        |            |        |          |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |
|        |              |       |                |                |      |      |          |    |    |    |       |      |       |        |            |        |          |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |
|        |              |       |                |                |      |      |          |    |    |    |       |      |       |        |            |        |          |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |
|        |              |       |                |                |      |      |          |    |    |    |       |      |       |        |            |        |          |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |
|        |              |       |                |                |      |      |          |    |    |    |       |      |       |        |            |        |          |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |
|        |              |       |                |                |      |      |          |    |    |    |       |      |       |        |            |        |          |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |
|        |              |       |                |                |      |      |          |    |    |    |       |      |       |        |            |        |          |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |
|        |              |       |                |                |      |      |          |    |    |    |       |      |       |        |            |        |          |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |
|        |              |       |                |                |      |      |          |    |    |    |       |      |       |        |            |        |          |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |
|        |              |       |                |                |      |      |          |    |    |    |       |      |       |        |            |        |          |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |
|        |              |       |                |                |      |      |          |    |    |    |       |      |       |        |            |        |          |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |
|        |              |       |                |                |      |      |          |    |    |    |       |      |       |        |            |        |          |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |
|        |              |       |                |                |      |      |          |    |    |    |       |      |       |        |            |        |          |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |
|        |              |       |                |                |      |      |          |    |    |    |       |      |       |        |            |        |          |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |
|        |              |       |                |                |      |      |          |    |    |    |       |      |       |        |            |        |          |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |
|        |              |       |                |                |      |      |          |    |    |    |       |      |       |        |            |        |          |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |
|        |              |       |                |                |      |      |          |    |    |    |       |      |       |        |            |        |          |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |
|        |              |       |                |                |      |      |          |    |    |    |       |      |       |        |            |        |          |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |
|        |              |       |                |                |      |      |          |    |    |    |       |      |       |        |            |        |          |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |

| Bit | Name           | Function                     | Description                                                                                                                                                                                                                                                                                                                                                                                                                                    | Access |
|-----|----------------|------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31  | EPENA          | Endpoint enable              | The application sets this bit to initiate data sending on the endpoint.<br>The module clears this bit before any of the following interrupts are triggered on this endpoint: <ul style="list-style-type: none"> <li>● Endpoint disable</li> <li>● Transmission completion</li> </ul>                                                                                                                                                           | RW     |
| 30  | EPDIS          | Endpoint disable             | An application can set this bit to stop data sending/receiving on an endpoint even before the transfer is complete on that endpoint. The application must wait for an endpoint disable interrupt to occur before the endpoint is considered disabled. The module clears this bit before the endpoint disable interrupt bit is set to 1. The application can set this bit to 1 only after the endpoint enable bit of this endpoint is set to 1. | RW     |
| 29  | SD1PID/SODDFRM | Set DATA1 PID/set odd frame  | <b>SD1PID:</b> Applicable only to interrupt/bulk OUT endpoints. Writing to this field sets the endpoint data PID (DPID) field in this register to DATA1.<br><b>SODDFRM:</b> Applicable only to synchronous OUT endpoints. Writing to this field sets the Even/odd frame (EONUM) field to odd frames.                                                                                                                                           | RW     |
| 28  | SD0PID/SEVNFRM | Set DATA0 PID/Set even frame | <b>SD0PID:</b> Applicable only to interrupt/bulk OUT endpoints. Writing to this field sets the endpoint data PID (DPID) field in this register to DATA0.<br><b>SEVNFRM:</b> Applicable only to synchronous OUT endpoints. Writing to this field sets the Even/odd frames (EONUM) field to even frames.                                                                                                                                         | RW     |

| Bit   | Name            | Function                         | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | Access |
|-------|-----------------|----------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 27    | SNAK            | Set NAK                          | Writing to this bit sets the NAK bit of the endpoint to 1.<br>With this bit, the application can control the sending of NAK handshake signals on the endpoint. The module can also set this bit of the OUT endpoint to 1 when a transfer complete interrupt occurs or after SETUP is received on the endpoint.                                                                                                                                                                                                                                                                                                                                                           | RW     |
| 26    | CNAK            | Clear NAK                        | Writing to this bit clears the NAK bit of the endpoint.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | RW     |
| 25:22 | <i>Reserved</i> | <i>Reserved</i>                  | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | -      |
| 21    | STALL           | STALL handshake                  | The application can only set this bit when the endpoint receives a SETUP token, and the module clears it. If the NAK bit, global OUT NAK, and this bit are all set to 1, the STALL bit takes precedence. Only the application can clear this bit, not the module.                                                                                                                                                                                                                                                                                                                                                                                                        | RW     |
| 20    | SNPM            | Snoop mode                       | This bit is used to configure the endpoint in listen mode. In the monitor mode, the module will no longer check the correctness of the received data.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | RW     |
| 19:18 | EPTYP           | Endpoint type                    | <ul style="list-style-type: none"> <li>● 0b00: Control</li> <li>● 0b01: Isochronous</li> <li>● 0b10: Bulk</li> <li>● 0b11: Interrupt</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | RO     |
| 17    | NAKSTS          | NAK status                       | <ul style="list-style-type: none"> <li>● 0b0: The module replies with a non-NAK handshake based on FIFO status</li> <li>● 0b1: The module replies with a NAK handshake on this endpoint</li> </ul> <p>When the application or module sets this bit to 1, the module will stop receiving any data on the OUT endpoint, even if there is space available in the RxFIFO to accommodate incoming packets. Regardless of how this bit is set, the module always responds to SETUP packets with an ACK handshake.</p>                                                                                                                                                          | RO     |
| 16    | EONUM/<br>DPID  | Even/Odd frame/Endpoint data PID | <p><b>EONUM:</b><br/>Applies only to synchronous OUT endpoints. Indicates the number of the frame in which the module sends/receives synchronized data for this endpoint. The application must program the even/odd frame number through the SEVNFRM and SODDFRM fields in this register for this endpoint to send/receive isochronous data.</p> <ul style="list-style-type: none"> <li>● 0b0: Even Frame</li> <li>● 0b1: Odd Frame</li> </ul> <p><b>DPID:</b><br/>Applies to interrupt/bulk OUT endpoints only. Contains the PID of packets that will be received or sent on this endpoint. After an endpoint is activated, the application must program the PID of</p> | RO     |

| Bit   | Name     | Function            | Description                                                                                                                                                                                                                                                                                                                                   | Access |
|-------|----------|---------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
|       |          |                     | the first packet to be received or sent on this endpoint. The application programs the DATA0 or DATA1 PID using the SD0PID register field.<br><ul style="list-style-type: none"> <li>● 0b0: DATA0</li> <li>● 0b1: DATA1</li> </ul>                                                                                                            |        |
| 15    | USBAEP   | USB active endpoint | Indicates whether this endpoint is active in the current configuration and interface. The module clears this bit to 0 for all endpoints except endpoint 0 when a USB reset is detected. After receiving the SetConfiguration and SetInterface commands, the application must program the endpoint register accordingly and set this bit to 1. | RO     |
| 14:11 | Reserved | Reserved            | -                                                                                                                                                                                                                                                                                                                                             | -      |
| 10:0  | MPSIZ    | Maximum packet size | The application must program this field to the maximum packet size of the current logical endpoint. This value is in bytes.                                                                                                                                                                                                                   | RW     |

### 34.6.5.23 Device OUT Endpoint x Interrupt Register (USBHS\_DOEPINTx) (x=0-15)

This register indicates the status of the endpoint when USB and AHB related events occur. When the OUT endpoint interrupt bit (OEPINT bit in USBHS\_GINTSTS) in the USBHS\_GINTSTS register is set to 1, the application must read this register. Before the application can read this register, it must first read the USBHS\_DAIN register to obtain the correct endpoint number of the USBHS\_DOEPINTx register. The application must clear the corresponding bit in this register in order to clear the corresponding bits in the USBHS\_DAIN and USBHS\_GINTSTS registers.

Offset: 0xB08+0x20\*x

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |      |          |    |    |    |    |   |   |   |   |   |         |          |         |      |          |        |      |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|------|----------|----|----|----|----|---|---|---|---|---|---------|----------|---------|------|----------|--------|------|
| -      | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15   | 14       | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4       | 3        | 2       | 1    | 0        |        |      |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |      |          |    |    |    |    |   |   |   |   |   |         |          |         |      |          |        |      |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | NYET | Reserved |    |    |    |    |   |   |   |   |   | B2BSTUP | Reserved | OTEPDIS | STUP | Reserved | EPDISD | XFRC |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RW   |          |    |    |    |    |   |   |   |   |   | RW      |          | RW      | RW   |          | RW     | RW   |

| Bit   | Name     | Function                                  | Description                                                                                                                                                                                                                                                                                                        | Access |
|-------|----------|-------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:15 | Reserved | Reserved                                  | -                                                                                                                                                                                                                                                                                                                  | -      |
| 14    | NYET     | NYET interrupt                            | When the non-synchronous OUT endpoint replies with a NYET handshake signal, the module generates this interrupt. Writing 1 to this bit by software can also clear it.                                                                                                                                              | RW     |
| 13:7  | Reserved | Reserved                                  | -                                                                                                                                                                                                                                                                                                                  | -      |
| 6     | B2BSTUP  | Received consecutive SETUP packet         | Applicable only to control OUT endpoints. This bit indicates that this endpoint has received more than three consecutive SETUP packets. Writing 1 to this bit by software can also clear it.                                                                                                                       | RW     |
| 5     | Reserved | Reserved                                  | -                                                                                                                                                                                                                                                                                                                  | -      |
| 4     | OTEPDIS  | OUT token received when endpoint disabled | Applicable only to control OUT endpoints. Indicates that an OUT token was received when the endpoint was not yet enabled, generating an interrupt. Write 1 to clear by software.                                                                                                                                   | RW     |
| 3     | STUP     | SETUP phase done                          | Applicable only to control OUT endpoints. Indicates that the SETUP phase of the control endpoint has been completed and no more consecutive SETUP data packets is received in the current control transfer. On this interrupt, the application can decode the received SETUP packet. Write 1 to clear by software. | RW     |
| 2     | Reserved | Reserved                                  | -                                                                                                                                                                                                                                                                                                                  | -      |
| 1     | EPDISD   | Endpoint disabled interrupt               | This bit indicates that the endpoint has been disabled by the application. Write 1 to clear by software.                                                                                                                                                                                                           | RW     |

| Bit | Name | Function                     | Description                                                                                                                      | Access |
|-----|------|------------------------------|----------------------------------------------------------------------------------------------------------------------------------|--------|
| 0   | XFRC | Transfer completed interrupt | This field indicates that transfers set on this end-point have completed transfers on USB and AHB. Write 1 to clear by software. | RW     |

### 34.6.5.24 Device OUT Endpoint 0 Transfer Size Register (USBHS\_DOEPTSIZE0)

The application must modify this register before enabling endpoint 0. After enabling endpoint 0 by setting the endpoint enable bit (EPENA in USBHS\_DIEPCTL0) in the device control endpoint 0 control register, the module modifies this register. The application can read this register only after the module has cleared the endpoint enable bit.

Non-zero endpoints use registers for endpoints 1-15.

| Offset | Bit Position |         |          |    |    |    |    |    |    |    |    |    |        |          |    |    |    |    |    |    |    |    |   |        |   |   |   |   |   |   |   |   |  |
|--------|--------------|---------|----------|----|----|----|----|----|----|----|----|----|--------|----------|----|----|----|----|----|----|----|----|---|--------|---|---|---|---|---|---|---|---|--|
| 0xB10  | 31           | 30      | 29       | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19     | 18       | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8      | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |  |
| Reset  | 0x00000000   |         |          |    |    |    |    |    |    |    |    |    |        |          |    |    |    |    |    |    |    |    |   |        |   |   |   |   |   |   |   |   |  |
| Name   | Reserved     | STUPCNT | Reserved |    |    |    |    |    |    |    |    |    | PKTCNT | Reserved |    |    |    |    |    |    |    |    |   | XFRSIZ |   |   |   |   |   |   |   |   |  |
| Access |              | RW      |          |    |    |    |    |    |    |    |    |    | RW     |          |    |    |    |    |    |    |    |    |   | RW     |   |   |   |   |   |   |   |   |  |

| Bit   | Name     | Function           | Description                                                                                                                                                                                                                                                                                                                                                                                             | Access |
|-------|----------|--------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31    | Reserved | Reserved           | -                                                                                                                                                                                                                                                                                                                                                                                                       | -      |
| 30:29 | STUPCNT  | SETUP packet count | This field specifies the number of SETUP data packets the endpoint can receive consecutively. <ul style="list-style-type: none"> <li>0b00: Setting prohibited</li> <li>0b01: 1 packet</li> <li>0b10: 2 packets</li> <li>0b11: 3 packets</li> </ul>                                                                                                                                                      | RW     |
| 28:20 | Reserved | Reserved           | -                                                                                                                                                                                                                                                                                                                                                                                                       | -      |
| 19    | PKTCNT   | Packet count       | The number of data packets that should be received in one transfer.<br>Before the endpoint is enabled, the software sets this bit. After the transmission starts, every time a data packet is received, the value of this field is automatically reduced.                                                                                                                                               | RW     |
| 18:7  | Reserved | Reserved           | -                                                                                                                                                                                                                                                                                                                                                                                                       | -      |
| 6:0   | XFRSIZ   | Transfer size      | Indicates the amount of data included in one data transfer for endpoint 0, in bytes. The module interrupts the application only after the application has transferred this data. The transfer size can be set to the endpoint's maximum packet size to break at the end of each packet.<br>The module decrements this field each time a packet is read from the Rx FIFO and written to external memory. | RW     |

### 34.6.5.25 Device OUT Endpoint x Transfer Size Register (USBHS\_DOEPTSIZx) (x=1-15)

The application must modify this register before enabling the endpoint. After the endpoint is enabled via the endpoint enable bit (EPENA bit in USBHS\_DOEPCTLx) in the USBHS\_DOEPCTLx register, the module modifies this register. The application can only read this register after the module clears the endpoint enable bit.

Offset: 0xB10+0x20\*x

| Offset | Bit Position |                |    |        |    |    |    |    |    |    |    |    |    |        |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |
|--------|--------------|----------------|----|--------|----|----|----|----|----|----|----|----|----|--------|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|--|
| -      | 31           | 30             | 29 | 28     | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18     | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |  |
| Reset  | 0x00000000   |                |    |        |    |    |    |    |    |    |    |    |    |        |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |
| Name   | Reserved     | RXDPID/STUPCNT |    | PKTCNT |    |    |    |    |    |    |    |    |    | XFRSIZ |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |
| Access |              | RW             |    | RW     |    |    |    |    |    |    |    |    |    | RW     |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |

| Bit   | Name               | Function                                   | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | Access |
|-------|--------------------|--------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31    | <i>Reserved</i>    | <i>Reserved</i>                            | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | -      |
| 30:29 | RXDPID/<br>STUPCNT | Received data<br>PID/SETUP<br>packet count | <b>RXDPID:</b> Only applicable to synchronous OUT end-points. This is the PID of the last packet received by this endpoint. <ul style="list-style-type: none"> <li>● 0b00: DATA0</li> <li>● 0b01: DATA2</li> <li>● 0b10: DATA1</li> <li>● 0b11: MDATA</li> </ul> <b>STUPCNT:</b> Only applicable to control OUT end-points. This field specifies the number of SETUP data packets the endpoint can receive consecutively. <ul style="list-style-type: none"> <li>● 0b00: Setting prohibited</li> <li>● 0b01: 1 packet</li> <li>● 0b10: 2 packets</li> <li>● 0b11: 3 packets</li> </ul> | RW     |
| 28:19 | PKTCNT             | Packet count                               | Indicates the number of data packets included in one data transfer on this endpoint. This field will be decremented after every packet (max size or short) written to the RxFIFO.                                                                                                                                                                                                                                                                                                                                                                                                      | RW     |

| Bit  | Name   | Function      | Description                                                                                                                                                                                                                                                                                                                                                                                                                          | Access |
|------|--------|---------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 18:0 | XFRSIZ | Transfer size | <p>This field contains the amount of data included in one data transfer of the current endpoint, in bytes. The module interrupts the application only after the application has transferred this data. The transfer size can be set to the endpoint's maximum packet size to break at the end of each packet.</p> <p>The module decrements this field each time a packet is read from the RxFIFO and written to external memory.</p> | RW     |



### 34.6.5.26 Device OUT Endpoint x DMA Address Register (USBHS\_DOEPMAX) (x=0-15)

This register is used to set the DMA address when the device endpoint is in DMA mode.

Offset: 0xB14+0x20\*x

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| -      | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0xFFFFFFFF   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | DMAADDR      |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access | RW           |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit  | Name    | Function    | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | Access |
|------|---------|-------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:0 | DMAADDR | DMA address | <p>This bit contains the starting address of the external storage area when data is sent on the endpoint using DMA.</p> <div>  <b>Notice</b><br/>           For control endpoints, the storage area pointed to by this field is also used to store control OUT data packets and SETUP transaction data packets. When three or more SETUP data packets are received consecutively, the SETUP data packets in the memory are overwritten. This register is incremented for every AHB transfer. The application must set a double word aligned address.         </div> | RW     |

### 34.6.6 USBHS Clock Gating Control Register

Control HCLK and PHY clock by gating the clock control register to reduce power consumption.

#### 34.6.6.1 Clock Gating Control Register (USBHS\_GCCTL)

This register is available in both host mode and device mode.

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |      |   |          |   |         |   |          |   |          |  |         |  |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|------|---|----------|---|---------|---|----------|---|----------|--|---------|--|
| 0xE00  | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7    | 6 | 5        | 4 | 3       | 2 | 1        | 0 |          |  |         |  |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |      |   |          |   |         |   |          |   |          |  |         |  |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   | SUSP |   | PHYSLEEP |   | ENL1GTG |   | Reserved |   | GATEHCLK |  | STPPCLK |  |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   | RO   |   | RO       |   | RW      |   |          |   | RW       |  | RW      |  |

| Bit  | Name     | Function                                     | Description                                                                                                                                                                                                                                                                                        | Access |
|------|----------|----------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:8 | Reserved | Reserved                                     | -                                                                                                                                                                                                                                                                                                  | -      |
| 7    | SUSP     | DeepSleep (L1 suspended) status bit          | <ul style="list-style-type: none"> <li>0b0: Not in DeepSleep state.</li> <li>0b1: In DeepSleep state.</li> </ul>                                                                                                                                                                                   | RO     |
| 6    | PHYSLEEP | PHY SLEEP mode status bit (PHY in sleep).    | <ul style="list-style-type: none"> <li>0b0: PHY is not in Sleep state.</li> <li>0b1: PHY is in Sleep state.</li> </ul>                                                                                                                                                                             | RO     |
| 5    | ENL1GTG  | L1 clock gating (Enable sleep clock gating). | This bit is used for clock gating control in L1 mode.                                                                                                                                                                                                                                              | RW     |
| 4:2  | Reserved | Reserved                                     | -                                                                                                                                                                                                                                                                                                  | -      |
| 1    | GATEHCLK | Gate HCLK                                    | When USB communication is suspended or the session is disabled, the application sets this bit to 1 to stop the clock to all modules except the AHB bus slave interface, host interface, and wake-up logic. The application clears this bit when USB communication resumes or a new session starts. | RW     |
| 0    | STPPCLK  | Stop PHY clock                               | When USB communication is suspended, the session is disabled, or the device is disconnected, the application sets this bit to 1 to stop the PH clock. The application program clears this bit when USB communication resumes.                                                                      | RW     |

## 35 USB2.0 Full-Speed Module (USBFS)

### 35.1 Introduction

USB full-speed (USBFS) controller provides a set of USB communication solutions for portable devices. The USBFS controller supports host mode and device mode, and a full-speed PHY is integrated inside the chip. In host mode, the USBFS controller supports full-speed (FS, 12Mbit/s) and low-speed (LS, 1.5Mbit/s) transceivers, while in device mode it only supports full-speed (FS, 12Mbit/s) transceivers. USBFS controller supports all four transmission modes defined by USB 2.0 protocol (Control transfer, Bulk transfer, Interrupt transfer, and Isochronous transfer). The USBFS controller supports LPM (Link Power Management) function.

The protocol is as follows:

- Universal Serial Bus Revision 2.0 Specification
- USB 2.0 Link Power Management Addendum Engineering Change Notice to the USB 2.0 specification, July 16, 2007
- Errata for USB 2.0 ECN: Link Power Management (LPM) - 7/2007

### 35.2 Features

#### 35.2.1 General Features

- Built-in on-chip USB2.0 full-speed PHY
- Supports host mode and device mode
- Supports FS SOF and low-speed "Keep-alive" tokens with the following functions:
  - ▶ SOF pulse pin output function
  - ▶ The SOF pulse can serve as an internal event source to trigger operations of modules such as TIMER and DMA.
  - ▶ Configurable frame period
  - ▶ Configurable frame end interrupt
- The module embeds DMA and supports software-configurable AHB burst transmission types
- Supports power-saving functions, such as USB suspend, stopping RAM clock, and stopping PHY domain clock
- Equipped with a dedicated 2.5KB RAM featuring advanced FIFO control
  - ▶ The RAM space can be partitioned into different FIFOs for flexible and efficient utilization of RAM
  - ▶ Each FIFO can store multiple data packets
  - ▶ Dynamic allocation of memory area
  - ▶ The FIFO size can be configured to non-power-of-two values for continuous memory unit utilization
- Within one frame, no application intervention is required to achieve maximum USB bandwidth
- The mode (Host Mode or Device Mode) can be automatically determined based on the ID line level

#### 35.2.2 Host Mode Features

- Host mode supports USB2.0 full-speed (FS, 12Mbit/s) and low-speed (LS, 1.5Mbit/s) transmissions
- The VBUS voltage needs to be generated by an external power supply chip
- Up to 16 host channels (Pipes) : Each channel can be dynamically reconfigured to support any type of USB transfer

- Built-in hardware scheduler capable of:
  - ▶ Store up to 8 interrupt plus isochronous transfer requests in a periodic hardware queue
  - ▶ Store up to 8 control and bulk transfer requests in a non-periodic hardware queue
- Manages a shared RX FIFO, a periodic TX FIFO, and a non-periodic TX FIFO to efficiently utilize USB data RAM

### 35.2.3 Device Mode Features

- Slave mode supports USB2.0 full-speed (FS, 12Mbit/s) transmission
- 1 bidirectional control endpoint 0
- 15 OUT endpoints, configurable to support bulk transfers, interrupt transfers, or isochronous transfers.
- 15 IN endpoints, configurable to support bulk transfers, interrupt transfers, or isochronous transfers.
- Contains 16 transmit FIFOs (one for each IN endpoint) and one receive FIFO (shared by all OUT endpoints)
- Supports remote wake-up function
- Supports soft disconnect function
- The VBUS PIN supports 5V withstand voltage

## 35.3 Functional Description

### 35.3.1 System Block Diagram

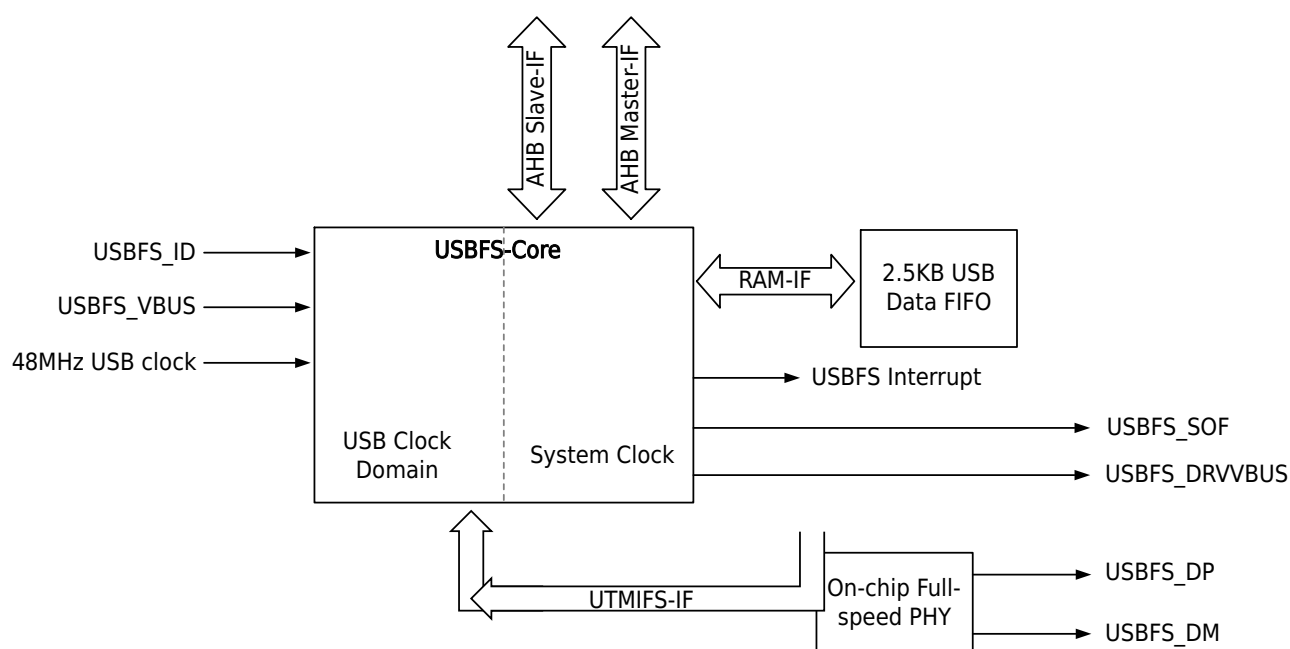


Figure 35-1 USBFS System Block Diagram

### 35.3.2 Pin Description

Table 35-1 USBFS Pin Description

| Pin Name      | Direction    | Applicable Mode | Corresponding Pins | Function Description                 |
|---------------|--------------|-----------------|--------------------|--------------------------------------|
| USBFS_VBUS    | Input        | Equipment       | PA9                | Power port, 5V withstand voltage     |
| USBFS_DP      | Input/Output | Host, device    | PA12               | Differential data D+ signal          |
| USBFS_DM      | Input/Output | Host, device    | PA11               | Differential data D-signal           |
| USBFS_DRVVBUS | Output       | Host            | PC9, PB8           | External power chip enable signal    |
| USBFS_ID      | Input        | Host            | PA10               | USB A-B device identification signal |
| USBFS_SOF     | Output       | Host, device    | PA8                | SOF output pulse signal              |

Since the USBFS\_DP and USBFS\_DM pins are multiplexed with universal GPIO, it is recommended to disable the digital function of the corresponding pins when using USBFS. The USBFS\_DP and USBFS\_DM functions are analog functions, independent of the associated PFSR register settings. For details, see the relevant description in the "Port Controller (GPIO) " chapter. In addition, when the USBFS function is not used, the digital function pin corresponding to the USBFS\_DP and USBFS\_DM pins consumes additional current when flipped.

### 35.3.3 Clock and Work Mode

The clock used by USBFS must be configured to 48MHz. This 48MHz clock is generated by the internal PLL circuit. The PLL clock source must select the external high speed oscillator. Before using the USBFS Block, the USBFS clock must be properly configured in the CMU Block.

USBFS can operate as either a host or device, and includes an on-chip full-speed PHY.

The pull-up and pull-down resistors are integrated inside the on-chip full-speed PHY, and USBFS can automatically select them based on the current mode and connection status.

During USBFS operation, the VCC voltage range is 3.0-3.63V.

### 35.3.4 Mode Decision

USBFS determines the current operating mode using the following two methods:

- Method 1: Automatic identification based on the status of USBFS\_ID line. When the USBFS\_ID line is detected at high level, the module works in the device mode. When the USBFS\_ID line is detected at low level, the module works in host status.
- Method 2: Force host/device mode. By setting the FDMOD or FHMOD bit in the USBFS\_GUSBCFG register to 1, the module will ignore the USBFS\_ID line level and forcibly worked in device or host mode.

### 35.3.5 Host Function

#### Function Introduction

When USBFS works in host mode, VBUS is the 5V power supply pin specified by the USB protocol. The internal PHY does not support 5V power supply, so an external USB power chip is required to power the device. USBFS\_DRVVBUS is used to enable the external USB power chip. The overcurrent detection of the external power supply chip can be implemented through the external interrupt IRQ of this MCU. In host mode, USBFS\_VBUS can be used as GPIO.

A typical USB host mode system architecture diagram is as follows:

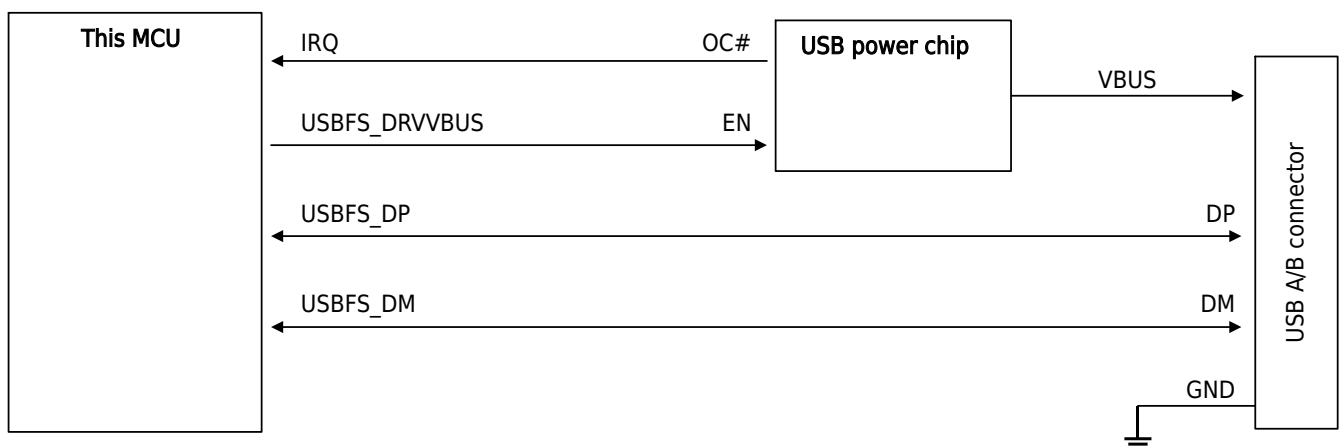


Figure 35-2 USBFS Host Mode System Architecture Diagram

#### Host Port Power Supply

This MCU cannot output 5V to supply VBUS. To achieve this, a USB power supply chip or a basic power switch (if the circuit board provides 5V power) must be added externally to the microcontroller to drive the 5V VBUS line. The external power chip can be driven by any GPIO output or USBFS\_DRVVBUS. When

the application determines to use GPIO to control an external device for providing VBUS, the port power bit (USBFS\_HPRT.PWPR) in the host port control and status register must still be set to 1.

### Host Detects Device Connection and Disconnection

The USB device will be detected immediately after connection. The USBFS module will issue a host port interrupt signal, which is triggered by the device connection bit (USBFS\_HPRT.PCDET) in the host port control and status register.

A device disconnect event will trigger a disconnect detection interrupt (USBFS\_GINTSTS.DISCINT).

### Host Enumeration

After a device connection is detected, if a new device is connected, the host must initiate the enumeration process by sending a USB reset and configuration command to the new device.

The application sets the port reset bit (USBFS\_HPRT.PRST) in the host port control and status register to 1, maintaining this state for a minimum of 10ms and a maximum of 20ms, thereby driving the USB reset signal (single-ended zero) through the USB driver. The application calculates the duration of this process, then clears the port reset bit.

After the USB reset sequence is completed, the port enable/disable change bit (USBFS\_HPRT.PENCHNG) triggers a host port interrupt immediately, notifying the application that the enumerated device speed can be read from the port speed field (PSPD in USBFS\_HPRT) in the host port control and status register, and that the host has started driving SOF (FS) or Keep-alive tokens (LS). At this point the host is ready to enumerate the device by sending commands to the device.

### Host Suspend

The application suspends the USB bus by setting the port suspend bit (PSUSP in USBFS\_HPRT) in the host port control and status register to 1. The USBFS module stops sending SOF and enters suspend status.

The bus can exit suspend status through autonomous activity (remote wake-up) from a remote device. In this scenario, the remote wake-up signal will trigger a remote wake-up interrupt (USBFS\_GINTSTS.WKUPINT). The hardware automatically resets the port resume bit (USBFS\_HPRT.PRES) in the host port control and status register and drives the resume signal via USB autonomously. The application must time the resume window, then clear the port resume bit to exit the suspend status and restart SOF.

If the host initiates exit from suspend status, the application must set the port resume bit to 1 to activate the resume signaling on the host port, time the resume window, and finally clear the port resume bit.

### Host Channel

The USBFS module implements 16 host channels. Each host channel can be used for USB host transfers (USB pipes). The host can handle up to 8 transfer requests at the same time. If the application has more than 8 transfer requests pending, the host controller driver (HCD) must reassign the channel for unprocessed transfer requests after the channel is released from the previous task (i.e., upon receiving the transmission complete and channel halt interrupts).

Each host channel can be configured to support input/output and periodic/non-periodic transactions. Each host channel uses dedicated control (HCCHARx) register, transfer configuration (HCTSIZx) register/interrupt (HCINTx) register, and its associated interrupt mask register (HCINTMSKx).

#### Host Channel Control

The application can control the host channel through the host channel x characteristic register (HCCHARx) with the following settings:

- Channel enabled/disabled
- Set the target USB device speed: FS/LS
- Set the target USB device address
- Set the endpoint number on the target USB device for communication with this channel
- Sets the transfer direction on this channel: IN/OUT
- Set the type of USB transfer on this channel: control/bulk/interrupt/isochronous
- Set the maximum packet length of the device endpoint for communication with this channel
- Set the frame for periodic transfer: odd frame/even frame

#### Host Channel Transfer

The host channel transfer size register (HCTSIZx) allows applications to program the transmission size parameter and read the transfer status. The settings for this register must be completed before setting the channel enable bit to 1 in the host channel characteristics register. After enabling the endpoint, the packet count field immediately becomes read-only, while the USBFS module updates this field based on the current transmission status.

The following transfer parameters can be programmed:

- Transfer size in bytes
- The number of packets that make up the entire transfer size
- Initial Data PID

#### Host Channel Status/Interrupt

Host channel x interrupt register (HCINTx) indicates the endpoint status upon occurrence of USB and AHB related events. When the host channel interrupt bit (USBFS\_GINTSTS.HCINT) in the interrupt register is set to 1, the application must read these registers to obtain detailed information. Before reading these registers, the application must first read the host all channels interrupt (HCAINT) register to obtain the channel number of the host channel-x interrupt register. The application must clear the corresponding bit in this register to clear the corresponding bits in the HCAINT and GINTSTS registers. The USBFS\_HCINTMSKx register also provides mask bits for each interrupt source on every channel.

The host module provides the following status check and interrupt generation functions:

- Transfer complete interrupt, indicating both the application (AHB) and USB side have completed data transfer
- The channel stops due to transmission completion, USB transaction error, or a disable command issued by the application
- The associated transmit FIFO is half-empty or in full empty status (IN endpoint)
- ACK response received
- NAK response received
- STALL response received
- USB transaction error due to CRC check failure, timeout, bit stuffing error, or incorrect EOP
- Crosstalk error
- Frame overflow
- Error in toggling bit for data synchronization



## Host Scheduler

The host module has a built-in hardware scheduler that autonomously reorders and manages USB transaction requests issued by applications. At the start of each frame, the host first executes periodic (synchronous and interrupt) transactions, followed by non-periodic (control and bulk) transactions, to comply with the USB specification's guarantee of high priority for synchronous and interrupt transfers.

The host processes USB transactions through request queues (a periodic request queue and a non-periodic request queue). Each request queue can store up to 8 entries. Each entry represents a USB transaction request initiated by an application but not yet responded to, storing the IN or OUT channel number used to execute the USB transaction, along with other relevant information. The order in which USB transaction requests are written to the queue determines the execution sequence of transactions on the USB interface.

At the start of each frame, the host first processes the periodic request queue, then the non-periodic request queue. If at the end of the current frame, a pending synchronous or interrupt-type USB transfer transaction request scheduled for the current frame remains in suspend status, the host will issue an incomplete periodic transfer interrupt (USBFS\_GINTSTS.IPXFR). The USBFS module is responsible for managing both periodic and non-periodic request queues. Periodic transmit FIFO and queue status register (HPTXSTS) and non-periodic transmit FIFO and queue status register (HNPTXSTS) are both read-only registers. Applications can use them to read the status of each request queue. Including:

- Number (maximum 8) of currently available free entries in the periodic (non-periodic) request queue
- Periodic (non-periodic) TxFIFO (OUT transaction) currently available free space
- IN/OUT tokens, host channel number, and other status information

Since each request queue can store up to 8 USB transaction requests, the application can send host USB transaction requests to the scheduler in advance. The actual communication will appear on the USB bus no later than after the scheduler completes processing the 8 pending periodic transactions and 8 non-periodic transactions.

To issue a transaction request to the host scheduler (queue), the application must read the USBFS\_HPTXSTS.PTXQSAV or USBFS\_HNPTXSTS.NPTQSAV bit to ensure there is at least one available space in the periodic (non-periodic) request queue to store the current request.

## 35.3.6 Device Function

### Function Introduction

When the USBFS works in device mode, VBUS is the 5V power supply pin specified by the USB protocol and is a 5V withstand voltage pin. This module continuously monitors the VBUS line level status to connect or disconnect devices. The USBFS\_VBUS port must be connected to the VBUS power supply through a resistor of at least 1kΩ.

A typical USB device mode system block architecture diagram is as follows:

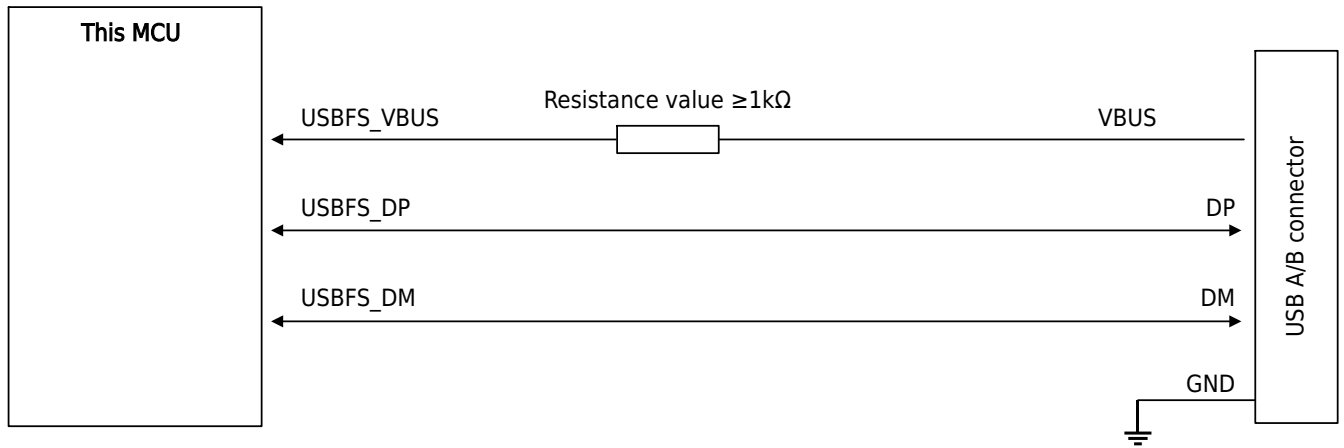


Figure 35-3 USBFS Device Mode System Architecture Diagram

### Power Supply State of Device

When the module detects that USBFS\_VBUS is at a high level, it enables the USB device to enter the power supply state. Then, USBFS automatically connects the DP pull-up resistor, signals a full-speed device connection to the host, and generates a session request interrupt (USBFS\_GINTSTS.VBUSINT), indicating entry into the powered state.

In addition, the USBFS\_VBUS input ensures the host provides a active VBUS level during USB operation. If a low level is detected on VBUS (e.g., caused by power supply interference or host port shutdown), USBFS will automatically disconnect.

Under power supply, the USBFS expects to receive a reset signal from the host. Other USB operations cannot be performed. Upon receiving the reset signal, a reset interrupt is immediately generated (USBFS\_GINTSTS.USBRSST). After the reset signal ends, an enumeration complete interrupt (USBFS\_GINTSTS.ENUMDNE) will be generated, and USBFS will then enter the default state.

### Device Default State

By default, USBFS expects to receive a SET\_ADDRESS command from the host. Other USB operations cannot be performed. When a active SET\_ADDRESS command is decoded on USB, the application writes the corresponding address value to the device address field (USBFS\_DCFG.DAD) in the device configuration register. The USBF immediately enters the address state and is ready to respond to host transactions using the configured USB address.

### Device Suspend State

The USBFS device continuously monitors USB activity. After the USB idle time reaches 3ms, an early suspend interrupt (USBFS\_GINTSTS.ESUSP) will be issued, and after 3ms, the suspend interrupt (USBFS\_GINTSTS.USBSUSP) will confirm the device enters suspend status. Then, the device suspend bit (USBFS\_DSTS.SUSPSTS) in the devices status register is automatically set to 1, and USBFS immediately enters the suspend status.

The device can exit the suspend status by itself. In this case, the application will set the remote wake-up signal bit (USBFS\_DCTL.RWUSIG) in the device control register to 1 and clear it within 1ms to 15ms.

However, if the device detects a resume signal from the host, it will generate a resume interrupt (USBFS\_GINTSTS.WKUPINT), and the device suspend bit will be automatically cleared.

### Device Soft-off

The power supply state can be exited via software using the soft disconnect function. Setting the soft disconnect bit (USBFS\_DCTL.SDIS) in the device control register to 1 removes the DP pull-up resistor. At

this point, even though the USB cable is not physically unplugged from the host port, the host will still detect a device disconnect and trigger an Interrupt.

## Device Endpoint

### Endpoint Type

The USBFS block implements the following USB endpoints:

- Control endpoint 0:
  - ▶ Bidirectional and handles only control messages
  - ▶ Use a separate set of registers to handle IN and OUT transactions
  - ▶ Dedicated control (USBFS\_DIEPCTL0/USBFS\_DOEPCTL0) register, transfer configuration (USBFS\_DIEPTSIZ0/USBFS\_DOEPSIZ0) register, and status interrupt (USBFS\_DIEPINTx/USBFS\_DOEPINT0) register. The available bit groups in the control and transmission size registers differ slightly from those in other endpoints
- 15 IN endpoints:
  - ▶ Each endpoint can be configured to support isochronous, bulk, or interrupt transfer types.
  - ▶ Each endpoint has dedicated control (USBFS\_DIEPCTLx) register, transfer configuration (USBFS\_DIEPTSIZx) register, and status interrupt (USBFS\_DIEPINTx) register
  - ▶ The device IN endpoint universal interrupt mask register (USBFS\_DIEPMSK) can be used to enable/disable the same type of interrupt sources on all IN endpoints (including EP0)
  - ▶ Supports interrupt for incomplete isochronous IN transfer (USBFS\_GINTSTS.IISOIXFR). This interrupt is triggered when at least one isochronous IN endpoint has an incomplete transfer in the current frame. This interrupt is triggered together with the periodic frame interrupt (USBFS\_GINTSTS/EOPF)
- 15 OUT endpoints:
  - ▶ Each endpoint can be configured to support isochronous, bulk, or interrupt transfer types.
  - ▶ Each endpoint has dedicated control (USBFS\_DOEPCTLx) register, transfer configuration (USBFS\_DOEPSIZx) register, and status interrupt (USBFS\_DOEPINTx) register.
  - ▶ The device OUT endpoint universal interrupt mask register (USBFS\_DOEPMSK) can be used to enable/disable the same type of interrupt sources on all OUT endpoints (including EP0).
  - ▶ Supports for incomplete isochronous OUT transfer interrupts (USBFS\_GINTSTS.INCOMPISOOOUT), which will trigger when at least one transfer on an isochronous OUT endpoint is incomplete in the current frame. This interrupt is triggered together with the periodic frame interrupt (USBFS\_GINTSTS/EOPF).

### Endpoint Control

Applications can control the endpoint through the device endpoint x IN/OUT control register (DIEPCTLx/DOEPCTLx) with the following operations:

- Endpoint enable/disable
- Enable endpoint in current configuration
- Set USB transfer type (Isochronous, Bulk, and Interrupt)
- Supported packet size setting
- Set the Tx-FIFO number associated with the IN endpoint
- Configure the desired data0/data1 PID for receiving or transmitting (applicable only to bulk/interrupt transfers)
- Set the corresponding odd/even frame for receive or transmit transactions (isochronous transfer only)

- The NAK Bit can be set to reply NAK to the host request regardless of the current FIFO status
- The STALL bit can be set to make the hardware respond with STALL to all host tokens for this endpoint
- The OUT endpoint can be configured in listen mode, where CRC checking is disabled for received data

### Endpoint Transfer

The device endpoint x transmission size register (DIEPTSIZE<sub>x</sub>/DOEPTSIZE<sub>x</sub>) allows the application to program transfer size parameters and read transfer status. The setting of this register must be completed before setting the endpoint enable bit in the endpoint control register to 1. After enabling the endpoint, these fields immediately become read-only, while the USBFS block updates them based on the current transmission status.

The following transfer parameters can be programmed:

- Transfer size in bytes
- The number of packets that make up the entire transfer

### Endpoint Status/Status

Device endpoint x interrupt register (DIEPINT<sub>x</sub>/DOEPINT<sub>x</sub>) indicates the endpoint status upon occurrence of USB and AHB related events. When the OUT endpoint interrupt bit or IN endpoint interrupt bit (corresponding to the USBFS\_GINTSTS.OEPINT/IEPINT) in the module interrupt register is set to 1, the application must read these registers to obtain detailed information. Before an application can read these registers, it must first read the device overall endpoint interrupt (USBFS\_DAINTE) register to obtain the endpoint number of the device endpoint x interrupt register. The application must clear the corresponding bit in this register to clear the corresponding bits in the DAINTE and GINTSTS registers.

The module provides the following status check and interrupt generation functions:

- Transfer complete interrupt, indicating to the application that both the AHB and the USB side have completed the data transfer
- Setup phase completed (only for OUT endpoints of control transfer type)
- The associated transmit FIFO is half-empty or in full-empty status (IN endpoint)
- A NAK reply has been sent to the host (IN endpoint for isochronous transfers only)
- IN token received while TxFIFO is empty (only for IN endpoints of bulk and interrupt transfer types)
- Received OUT token when endpoint is not enabled
- Babble error detected
- Application closes endpoint enable
- The application sets NAK to take effect for the endpoint (only applicable to IN endpoints of isochronous transfer type)
- Received more than 3 consecutive setup packets (only applicable to control-type OUT endpoints)
- Timeout detected (only for IN endpoints of control transfer type)

### 35.3.7 SOF Pulse Pin Output Function

In both host and device modes, USBFS can monitor, track, and configure SOF frames, and also features SOF pulse output functionality. The SOF pulse is output through the USBFS\_SOF pin, with a pulse width of 16 HCLK clock periods.

#### Host SOF

In host mode, the number of PHY clocks occurring between two consecutive SOF (FS) or keep-alive (LS) tokens can be programmed in the host frame interval register (HFIR), allowing the application to control the SOF frame period. An interrupt is generated at the start of frame (SOF bit in USBFS\_GINTSTS). The

current frame number and the remaining time until the next SOF can be tracked by the application in the host frame number register (HFNUM).

Using the SOFEN bit in USBFS\_SYCTLREG register, a SOF pulse signal with a width of 16 HCLK system clock periods can be output from the USBFS\_SOF pin simultaneously with any issued SOF token.

In addition, the SOF pulse can also serve as an internal event to trigger operations of external modules such as DMA transfers and TIMER counting.

### Device SOF

In device mode, each time the USB receives a SOF token, it will trigger a start-of-frame interrupt (SOF bit in USBFS\_GINTSTS). The corresponding frame number can be read from the device status register (FNSOF bit in USBFS\_DSTS). Using the SOFEN bit in the USBFS system control register USBFS\_SYCTLREG, a 16 HCLK system clock period SOF pulse signal can be generated and output on the USBFS\_SOF pin for external availability.

In addition, the SOF pulse can also work as an internal event to trigger operations of external modules such as DMA transfers and TIMER counting.

The periodic frame end interrupt (GINTSTS/EOPF) is used to notify the application when 80%, 85%, 90%, or 95% of the frame interval time has elapsed, depending on the periodic frame interval field (PFIVL bit in the USBFS\_DCFG) in the device configuration register. This function can be used to determine whether all isochronous communications for this frame are completed.

### 35.3.8 Power Consumption Control

When the USBFS module is not in use, the HCLK and PHY clocks of the USBFS module can be stopped through the CMU module to reduce power consumption.

When using the USB module, if the device USB session has not started or the device is not connected, power reduction techniques can be employed in the suspend status.

- Stop PHY clock (STPPCLK bit in USBFS\_GCCTL)

When the stop PHY clock bit in the clock gating control register is set to 1, most 48MHz internal clock domains in the USBFS full speed module are gated off by clock gating. Even if the application still provides a clock input, the dynamic power consumption caused by clock signal toggling in the block will be saved. Additionally, most units of the transceiver will be turned off, with only the portion responsible for detecting asynchronous recovery events or remote wake-up events remaining active.

- HCLK gating (GATEHCLK bit in USBFS\_GCCTL)

When the GATEHCLK bit in the clock gating control register is set to 1, most HCLK system clock domains inside the USBFS module are gated off by clock gating. Only the register read and write interface remains active. Even when the application still provides a clock input, the dynamic power consumption caused by clock signal toggling in the block is saved.

To save dynamic power consumption, the clock is only provided to the USB data FIFO when it is accessed by the USBFS module.

### 35.3.9 Dynamic Update of USBFS\_HFIR Register

In host mode, the USB module has the function to dynamically fine-tune the frame period, enabling synchronization between the external device and the SOF frame. If the USBFS\_HFIR register is modified within the current SOF frame, the SOF period will be adjusted accordingly in the next frame. For detailed specifications, refer to the figure below.

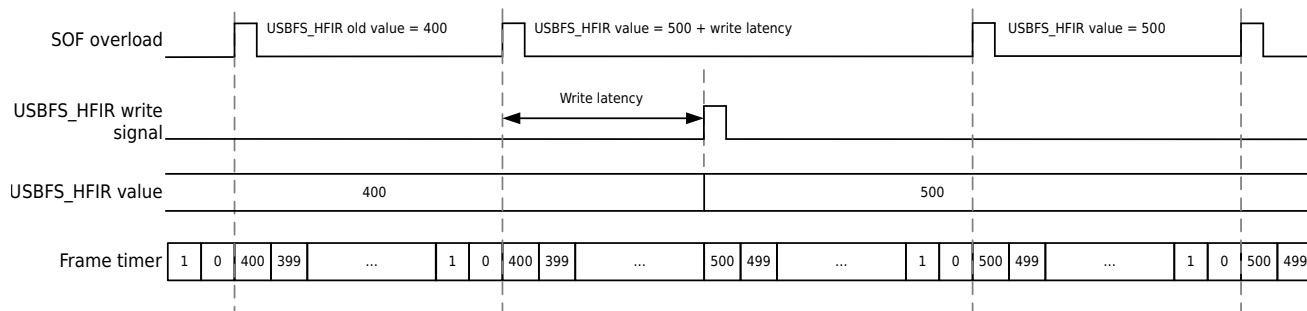


Figure 35-4 Schematic Diagram of USBFS Dynamic Update for USBFS\_HFIR Register

### 35.3.10 Data FIFO

The USBFS system features 2.5KB of dedicated RAM and employs an efficient FIFO control mechanism. In the USBFS module, the packet FIFO controller divides the RAM space into multiple TxFIFOs (Before USB transfer, the application pushes data into it for temporary storage) and a single Rx FIFO (The data received from the USB is briefly stored in it before being read by the application).

The number and organization of FIFOs built in RAM depend on the device's role. In device mode, a Tx FIFO is configured for each active IN endpoint. The FIFO size is software-configurable to better meet application requirements.

### 35.3.11 Host FIFO Architecture

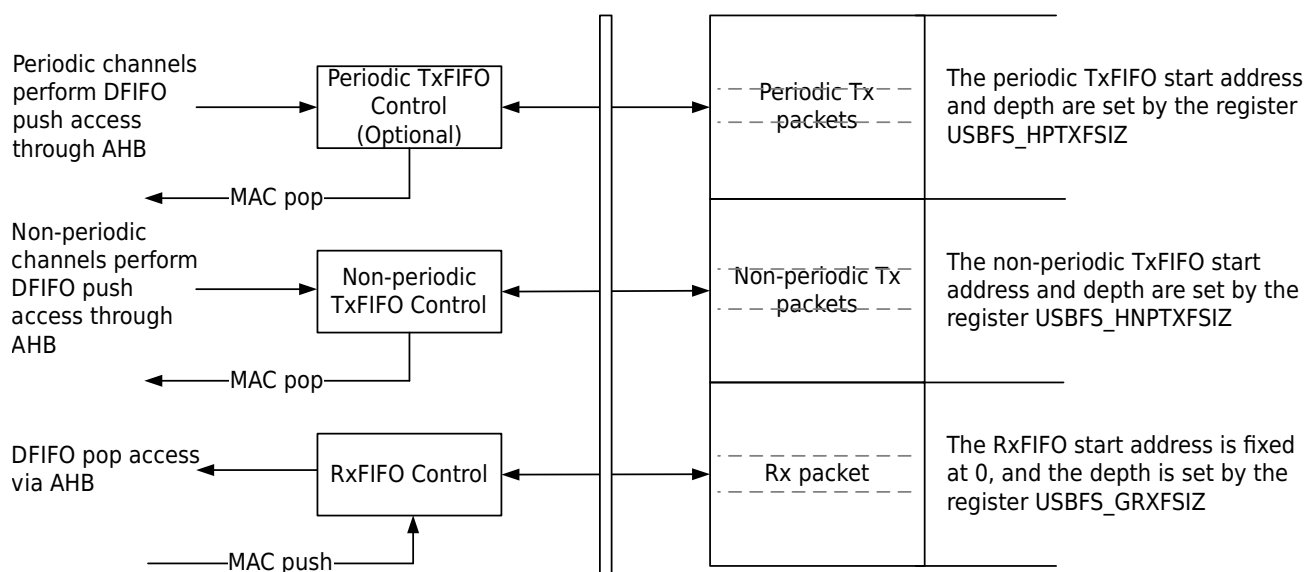


Figure 35-5 Schematic Diagram of FIFO Architecture in Host Mode

### Host Rx FIFO

The host uses a receive FIFO to handle all periodic and non-periodic transactions. The FIFO serves as a receive buffer to store data received from USB (the data portion of received packets) until it is transferred to system memory. As long as there is space in the FIFO, data packets from the device's IN endpoint are received and stored sequentially. The status of each received data packet (including the host target channel, byte count, data PID, and checksum of the received data) is also stored in the FIFO. The size of the receive FIFO is configured in the receive FIFO size register (GRXFSIZ).

The single receive FIFO architecture enables the USB host to efficiently fill the receive data buffer:

- All IN configuration host channels share the same RAM buffer (shared FIFO)

- For any sequence of IN tokens driven by host software, the USBFS module can fill the receive FIFO up to its limit.

The application receives an RxFIFO non-empty interrupt whenever at least one data packet is available for reading in the RxFIFO. The application reads packet information from the receiving status and out-stack register and finally retrieves the data from the RxFIFO.

### Host TxFIFO

The host uses one transmit FIFO for all non-periodic (control and bulk) OUT transactions and the other transmit FIFO for all periodic (isochronous and interrupt) OUT transactions. FIFO serves as a transmit buffer to store data (transmit packets) for USB transmission. The periodic (non-periodic) TxFIFO size is configured in the host periodic (non-periodic) transmit FIFO size (HPTXFSIZ/HNPTXFSIZ) register.

The two TxFIFOs operate based on priority, with periodic communication having higher priority. Therefore, within one USB frame, periodic communication is processed first. At the start of the frame, the built-in host scheduler first processes the periodic request queue, then handles the non-periodic request queue.

The dual transmit FIFO architecture enables the USB host to separately optimize the management of periodic and non-periodic transmit data buffers:

- All host channels configured to support periodic (non-periodic) OUT transactions share the same RAM buffer (shared FIFO).
- For any sequence of OUT tokens driven by host software, the USBFS module can fill the periodic (non-periodic) transmit FIFO up to the limit.

The USBFS module issues a periodic TxFIFO empty interrupt (USBFS\_GINTSTS.PTXFE) whenever the periodic TxFIFO is half-empty or full-empty, depending on the value of the periodic Tx-FIFO empty level bit (USBFS\_GAHBCFG.PTXFELVL) in the AHB configuration register. As long as there is free space in both the periodic TxFIFO and the periodic request queue, the application can pre-write transmit data. The available space can be determined by reading the host periodic transmit FIFO and queue status register (HPTXSTS).

Whenever the non-periodic TxFIFO is half-empty or full-empty, the USBFS module generates a non-periodic TxFIFO empty interrupt (USBFS\_GINTSTS.NPTXFE), depending on the non-periodic TxFIFO empty level bit (USBFS\_GAHBCFG.TXFELVL) in the AHB configuration register. As long as there is free space in both the non-periodic TxFIFO and the non-periodic request queue, the application can write to transmit data. The available space of both can be checked by reading the host non-periodic transmit FIFO and queue status register (HNPTXSTS).



### 35.3.12 Device FIFO Architecture

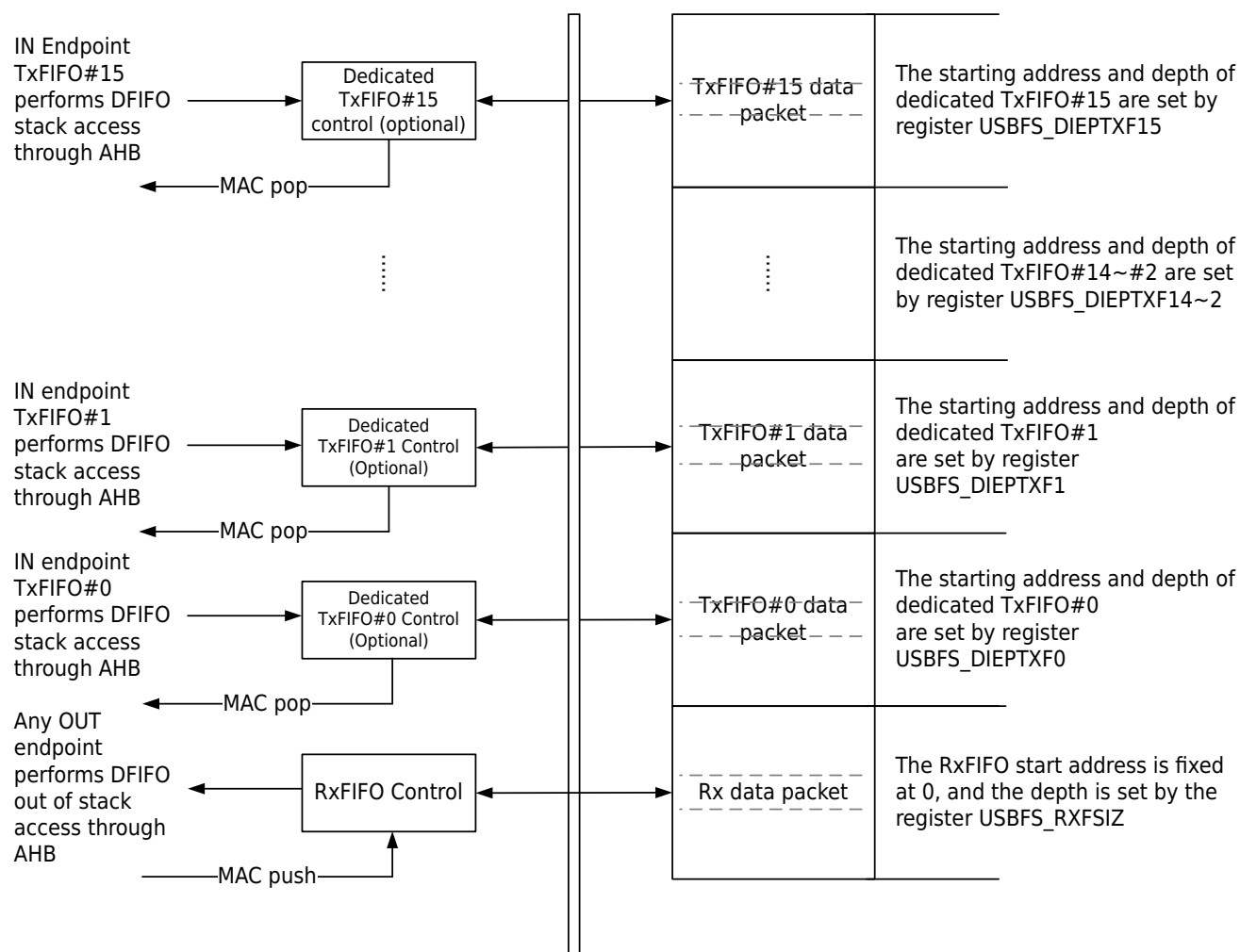


Figure 35-6 Schematic Diagram of FIFO Architecture in USBFS Device Mode

#### Device Rx FIFO

USBFS device uses a single receive FIFO to receive data sent to all OUT endpoints. As long as there is free space in the Rx FIFO, the received data packets are sequentially filled into the Rx FIFO. In addition to active data, the received data packet status (including OUT endpoint target number, byte count, data PID, and activation of the received data) is also stored by the module. When no available space exists, the device responds with a NAK to the host transaction and triggers an interrupt on the addressed endpoint. The size of the receive FIFO is configured in the receive FIFO size register (GRXFSIZ).

The single receive FIFO architecture enables USB devices to more efficiently fill the receive RAM buffer:

- All OUT endpoints share the same RAM buffer (shared FIFO)
- For any host sequence OUT token, the USBFS module can fill the receive FIFO up to its limit

The application will continuously receive the Rx FIFO non-empty interrupt (USBFS\_GINTSTS.RXFNE) as long as at least one data packet is available for reading in the Rx FIFO. The application reads data packet information from the receive status read and out-stack register (GRXSTSP), and finally retrieves the corresponding data from the receive FIFO by reading the out-stack address associated with the endpoint.



## Device Tx FIFO

The module provides dedicated FIFO for each IN endpoint. The application configures the FIFO size for IN endpoint 0 through Device Endpoint 0 Transmit FIFO Size Register (USBFS\_DIEPTXF0), and configures the FIFO size for IN endpoint x through Device IN Endpoint Transmit FIFO Size Register (USBFS\_DIEPTXFx).

### 35.3.13 FIFO RAM Allocation

#### Host Mode

##### Receive FIFO RAM Allocation

Status information is written to the FIFO with each receive data packet. Therefore, at least (maximum data packet size/4) + 1 space must be allocated for receive data packets. If multiple isochronous channels are enabled, the allocated space for receiving continuous data packets must be at least twice the (maximum data packet size/4) + 1. Generally, the recommended space is twice the (maximum data packet size/4 + 1). This allows the USB to receive subsequent data packets while the previous packet is being transferred to the CPU.

The transmission completion status information and the last received data packet of this endpoint are written together into the FIFO. Therefore, a location must be allocated for this purpose.

When running in DMA mode, the DMA address register of each host channel will be stored in the FIFO. Therefore, one location must be reserved in the FIFO for each channel to store its address register.

##### Transmit FIFO RAM Allocation

The minimum RAM required for the host non-periodic transmit FIFO is the size of the largest data packet transmitted across all supported non-periodic OUT channels.

Typically, the recommended space is twice the maximum packet size, so that while USB is transmitting the current packet, AHB can fill the next packet into the transmit FIFO.

The minimum RAM required for the host to periodically send FIFO is the size of the largest data packet transmitted across all supported periodic OUT channels. If there is at least one isochronous OUT endpoint, the space must be at least twice the maximum packet size in this channel.

When running in DMA mode, the DMA address register of each host channel will be stored in the FIFO. Therefore, one location must be reserved in the FIFO for each channel to store its address register.

#### Device Mode

##### Receive FIFO RAM Allocation

The application shall allocate RAM for SETUP packets: 11 locations must be reserved in the receive FIFO to receive SETUP data packets on the control endpoint. The USBFS module will not write any other data to these bits reserved for SETUP data packets. A location will be allocated for the global OUT NAK.

Status information is written to the FIFO with each receive data packet. Therefore, at least (maximum packet size/4) + 1 space must be allocated for receive data packets. If multiple isochronous endpoints are enabled, the allocated space for receiving consecutive data packets must be at least twice the (maximum packet size/4) + 1. Generally, the recommended space is twice the (Maximum packet size/4 + 1). This allows the USB to receive subsequent data packets while the previous data packet is being transferred to the CPU.

The transfer completion status information and the last received data packet of this endpoint are pushed into the FIFO together. Normally, it is recommended to allocate one bit for each OUT endpoint.

##### Transmit FIFO RAM Allocation

The minimum RAM space required for each IN endpoint's transmit FIFO is the maximum data packet size of that specific IN endpoint.

### 35.3.14 System Performance

With a large-capacity RAM buffer, highly configurable FIFO size, 32-bit FIFO fast access via AHB push/pop registers, and especially the advanced FIFO control mechanism, optimal USB and system performance can be achieved. In fact, regardless of the current USB sequence, the USBFS can efficiently fill the available RAM space through this mechanism. With these features:

- The application has enough margin to calculate and correct the CPU load, thereby optimizing CPU bandwidth utilization:
  - ▶ The application can first accumulate a large amount of data to transmit, then send it out via USB.
  - ▶ Provides sufficient timing margin to read data from the receive FIFO.
- The USB module can keep working at full speed, that is, to provide the maximum full-speed bandwidth (as much hardware as possible to run automatically, and as little software as possible to participate)
  - ▶ The USB module can accumulate a large amount of sending data in advance at its disposal, enabling automatic management of USB data transmission.
  - ▶ There is ample free space in the receive buffer, which can be automatically filled with data from USB.

The USBFS module can efficiently fill the 2.5KB RAM buffer, and the 2.5KB transmit/receive data is sufficient to accommodate the data volume of a full speed frame. Therefore, the USB system can achieve maximum USB bandwidth within a single frame without application intervention.

### 35.3.15 Interrupt and Event

The USBFS interrupts include the following two types: the STOP mode wake-up interrupt USBFS\_WKUP and the USBFS global interrupt USBFS\_GLB.

#### ● USBFS\_WKUP Interrupt

The USBFS\_WKUP interrupt is used to wake up the system from STOP mode via USBFS\_DP or USBFS\_DM. The interrupt enable bit for this function is INT\_WUPEN.USF\_WUEN.

To wake up the system from STOP mode using USBFS, ensure the USBFS controller is in suspend status and configure the corresponding filter range and enable the filter function in the USBFS\_SYCTLREG register.

#### ● USBFS\_GLB Interrupt

USBFS\_GLB interrupt is the primary interrupt that software needs to handle. The global interrupt flag bit can be read from the USBFS\_GINTSTS Register.

Table 35-2 USBFS\_GLB Interrupt Event List

| Interrupt Flag | Description                               | Operating Mode | Internal Event Source |
|----------------|-------------------------------------------|----------------|-----------------------|
| WKUPINT        | Resume/Remote wake-up interrupt           | Host or device | -                     |
| VBUSVINT       | VBUS active interrupt                     | Device         | -                     |
| DISCINT        | Disconnect interrupt                      | Host           | -                     |
| CIDSCHG        | Connector ID line status change interrupt | Host or device | -                     |

| Interrupt Flag     | Description                                                      | Operating Mode | Internal Event Source |
|--------------------|------------------------------------------------------------------|----------------|-----------------------|
| PTXFE              | Periodic TxFIFO empty interrupt                                  | Host           | -                     |
| LPMINT             | LPM interrupt                                                    | Host or device | -                     |
| HCINT              | Host channel interrupt                                           | Host           | -                     |
| HPRTINT            | Host port interrupt                                              | Host           | -                     |
| DATAFSUSP          | Data acquisition suspend                                         | Device         | -                     |
| IPXFR/INCOMPISOOUT | Incomplete periodic transfer/Incomplete OUT isochronous transfer | Device         | -                     |
| IISOIXFR           | Incomplete IN isochronous transfer                               | Device         | -                     |
| OEPINT             | OUT endpoint interrupt                                           | Device         | -                     |
| IEPINT             | IN endpoint interrupt                                            | Device         | -                     |
| EOPF               | Periodic frame end interrupt                                     | Device         | -                     |
| ISOODRP            | Discard synchronization OUT data packet interrupt                | Device         | -                     |
| ENUMDNE            | Enumeration complete                                             | Device         | -                     |
| USBRST             | USB reset interrupt                                              | Device         | -                     |
| USBSUSP            | USB suspend interrupt                                            | Device         | -                     |
| ESUSP              | Early suspend interrupt                                          | Device         | -                     |
| GONAKEFF           | Global OUT NAK active interrupt                                  | Device         | -                     |
| GINAKEFF           | Global non-periodic IN NAK active interrupt                      | Device         | -                     |
| NPTXFE             | Non-periodic TxFIFO empty interrupt                              | Host           | -                     |
| RXFNE              | RxFIFO not empty interrupt                                       | Host or device | -                     |
| SOF                | Frame start interrupt                                            | Host or device | Yes                   |
| MMIS               | Mode mismatch interrupt                                          | Host or device | -                     |

## 35.4 Programming Module

### 35.4.1 Module Initialization

#### Context

The application must perform the module initialization sequence. For details on how to determine the mode, see [Mode Determination](#).

This section describes the initialization process after power-on of the USBFS controller. Whether working in host mode or device mode, the application must follow the initialization sequence. Initialize all module global registers based on the module configuration:

### Procedure

- Step1.** Program the following fields in the USBFS\_GAHBCFG register:
- Global interrupt mask bit GINTMSK=1
  - RxFIFO not empty (RXFNE bit in USBFS\_GINTSTS)
  - Periodic TxFIFO empty threshold
- Step2.** Program the following fields in the USBFS\_GUSBCFG register:
- FS timeout calibration field
  - USB turnaround time field
- Step3.** The software must unmask the following bits in the USBFS\_GINTMSK register:
- Mode mismatch interrupt mask
- Step4.** By reading the CMOD bit in USBFS\_GINTSTS, the software can determine whether the USBFS controller is operating in host mode or device mode.

## 35.4.2 Host Initialization

### Context

To initialize the module as the host, the application must perform the following steps:

### Procedure

- Step1.** Program USBFS\_GINTMSK.HPRTIM bit to 1 to unmask the interrupt.
- Step2.** Program the USBFS\_HCFG register to select full-speed host.
- Step3.** Program the PWPR bit in USBFS\_HPRT to 1 to supply VBUS to the USB bus.
- Step4.** Wait for the PCDET interrupt in USBFS\_HPRT. This indicates that a device is connected to the host port.
- Step5.** Program the PRST bit in USBFS\_HPRT to 1 to issue a reset signal on the USB bus.
- Step6.** Wait at least 10ms for allow the reset process to complete.
- Step7.** Program the PRST bit in USBFS\_HPRT to 0.
- Step8.** Wait for the PENCHNG interrupt in USBFS\_HPRT.
- Step9.** Read the PSPD bit in USBFS\_HPRT to obtain the enumeration speed.
- Step10.** Use the selected PHY clock and configure the HFIR register accordingly.
- Step11.** Program the FSLSPCS field in the USBFS\_HCFG register according to the device speed detected in Step 9. If the FSLSPCS is modified, a port reset must be performed.
- Step12.** Program the USBFS\_GRXFSIZ register to select the size of the receive FIFO.
- Step13.** Program the USBFS\_HNPTXFSIZ register to select the size and start address of the non-periodic transmit FIFO for non-periodic communication transactions.
- Step14.** Program the USBFS\_HPTXFSIZ register to select the size and start address of the periodic communication transmit FIFO for periodic communication transactions.

### Post Request

To communicate with the device, the system software must initialize and enable at least one channel.

## 35.4.3 Device Initialization

### Context

During power on or after switching from host mode to device mode, the application must perform the following steps to initialize the module as a device.

### Procedure

- Step1.** Program the following fields in the USBFS\_DCFG register:

- Device speed
- Non zero length status OUT handshake signal

**Step2.** Program the USBFS\_GINTMSK register to unmask the following interrupts:

- USB reset
- Enumeration complete
- Early suspend
- USB suspend
- SOF

**Step3.** Wait for the VBUSVINT interrupt in USBFS\_GINTSTS, indicating the power supply state is entered.

**Step4.** Wait for the USBRST interrupt in USBFS\_GINTSTS. This indicates that a reset signal has been detected on USB, and the reset process lasts approximately 10ms after receiving this interrupt.

**Step5.** Wait for the ENUMDNE interrupt in USBFS\_GINTSTS. This interrupt indicates the end of the reset process on USB. Upon receiving this interrupt, the application must read the USBFS\_DSTS register to determine the enumeration speed and perform the steps listed in the endpoint initialization on enumeration completion.

## Results

At this point, the device is ready to receive SOF data packets and perform control transfers on control endpoint 0.

## 35.4.4 DMA Mode

The USB uses the AHB host interface to fetch transmit data packet data (AHB to USB) and receive data updates (USB to AHB). The AHB host interface uses programmed DMA address (HCDMAx register in host mode and DIEPDMAx/DOEPDMAx registers in device mode) to access data buffers.

## 35.4.5 Host Programming Model

### 35.4.5.1 Channel Initialization

#### Context

The application must initialize one or more channels before communicating with the connected devices.

To initialize and enable the channel, the application must perform the following steps:

#### Procedure

**Step1.** Program the USBFS\_GINTMSK register to unmask the interrupt for the following bits:

- The non-periodic transmit FIFO for OUT transactions is empty (applicable when working at the pipeline transaction level and the programmed packet count field value is greater than 1).
- The non-periodic transmit FIFO for OUT transactions is half-empty (applicable when working at the pipeline transaction level and the programmed packet count field value is greater than 1).

**Step2.** Program the USBFS\_HAINTMSK register to enable the selected channel interrupt.

**Step3.** Program the USBFS\_HCINTMSK register to enable the interrupts related to communication transactions reflected in the host channel interrupt register.

- Step4.** Program the selected channel's USBFS\_HCTSIZx register to specify the total transmission size in bytes and the expected number of packets, including short packets. The application must program the PID field with the initial data PID (used for the first OUT transaction or expected to be obtained from the first IN transaction).
- Step5.** Program the selected channel's USBFS\_HCCHARx register to specify the endpoint features of the device, such as type, speed, direction, etc. (Only when the application is ready to send or receive data packets can the channel be enabled by setting the channel enable bit to 1.)

### 35.4.5.2 Channel Halted

The application can disable any channel by programming the USBFS\_HCCHARx registers by setting the CHDIS and CHENA bits to 1. This will cause the USBFS host to clear any previously issued requests on this channel (if any) and generate a channel stop interrupt. The application must wait for the CHH interrupt in USBFS\_HCINTx before reassigning the channel to other communication transactions. The USBFS host will not interrupt communication transactions already initiated on the USB.

Before disabling a channel, the application must ensure that there is at least one free space in the non-periodic request queue (when disabling non-periodic channels) or the periodic request queue (when disabling periodic channels). The application can clear the request queue by programming the USBFS\_HCCHARx register to set the CHDIS bit to 1 and clear the CHENA bit to 0 when the request queue is full (before disabling the channel). The application will disable the channel when any of the following conditions occur:

- STALL, TXERR, BBERR or DTERR interrupt received in USBFS\_HCINTx of IN or OUT channel. The application must be capable of receiving other interrupts (DTERR, Nak, Data, TXERR) from the same channel before receiving the channel stop signal.
- Received a DISCINT (disconnect device) interrupt in USBFS\_GINTSTS. (The application will disable all enabled channels).
- The application aborted the transfer before it could complete normally.

When in DMA mode, the application cannot stop an integral periodic transfer by writing to the register.

## 35.4.6 Device Programming Model

### 35.4.6.1 Endpoint Initialization on USB Reset

#### Procedure

- Step1.** Set the NAK bit to 1 for all OUT endpoints
- In USBFS\_DOEPCTLx, SNAK=1 (for all OUT endpoints)
- Step2.** Unmask the following interrupt bits
- In USBFS\_DAINTRSK, INEP0=1 (control 0 IN endpoint)
  - In USBFS\_DAINTRSK, OUTEP0=1 (control 0 OUT endpoint)
  - In DOEPMSK, STUP=1
  - In DOEPMSK, XFRC=1
  - In DIEPMSK, XFRC=1
  - In DIEPMSK, TOC=1
- Step3.** Configure data FIFO RAM for each FIFO
- Program the USBFS\_GRXFSIZ register to enable receiving OUT data and setup data for control transfers. This register must be at least equal to 1 maximum packet size for control endpoint 0 + 2 words (for control OUT packet status) + 10 words (for SETUP packet).

- Program the USBFS\_TX0FSIZ register (depending on the selected FIFO number) to enable transmit control IN data. The register must be at least equal to 1 maximum packet size of control endpoint 0.

**Step4.** Program the following fields in the endpoint-related register to enable control OUT endpoint 0 to receive SETUP packets

- In USBFS\_DOEPTSIZE0, STUPCNT=3 (receives up to 3 consecutive SETUP packets)

### Results

At this point, all initialization work required for receiving SETUP packets is completed.

### 35.4.6.2 Endpoint Initialization Upon USB Enumeration Completion

#### Procedure

- Step1.** Upon enumeration completion interrupt (ENUMDNE in USBFS\_GINTSTS), read the USBFS\_DSTS register to determine the device's enumeration speed.
- Step2.** Program the MPSIZ field in USBFS\_DIEPCTL0 to set the maximum packet size. This step configures control endpoint 0. The maximum packet size of the control endpoint depends on the enumeration speed.

### Results

At this point, the device is ready to receive SOF packets and is configured to perform control transfers on control endpoint 0.

### 35.4.6.3 Endpoint Initialization Upon Receiving the SetAddress Command

#### Context

This section describes the operations that the application must perform when receiving a SetAddress command in the SETUP packet.

#### Procedure

- Step1.** Program the USBFS\_DCFG register using the device address received in the SetAddress command.
- Step2.** Program the module to issue an IN data packet during the status phase.

### 35.4.6.4 Endpoint Initialization Upon Receiving SetConfiguration/SetInterface Command

#### Context

This section describes the operations that the application must perform upon receiving a SetConfiguration or SetInterface command in the SETUP packet.

#### Procedure

- Step1.** Upon receiving the SetConfiguration command, the application must program the endpoint registers to configure them using the features of active endpoints in the new configuration.
- Step2.** Upon receiving the SetInterface command, the application must program the endpoint register of the endpoint specified by the command.
- Step3.** Endpoints that were active in previous configurations or other settings become inactive in new configurations or other settings. These inactive endpoints must be disabled.
- Step4.** Use the USBFS\_DAINMSK register to enable interrupts for active endpoints and mask interrupts for inactive endpoints.
- Step5.** Set up data FIFO RAM for each FIFO.
- Step6.** After configuring all required endpoints, the application must program the module to transmit IN packets for the status stage.



## Results

At this point, the device module is ready to receive and transmit any type of data packet.

### 35.4.6.5 Endpoint Activation

#### Context

This section describes the steps required to activate a device endpoint or configure an existing device endpoint as a new type.

#### Procedure

- Step1.** In the USBFS\_DIEPCTLx register (for IN or bidirectional endpoints) or USBFS\_DOEPCTLx register (for OUT or bidirectional endpoints), program the features of the desired endpoint in the following fields.
- Maximum packet size
  - USB active endpoint bit set to 1
  - Endpoint initial data synchronization bit (for interrupt and bulk endpoints)
  - Endpoint type
  - TxFIFO number
- Step2.** After the endpoint is enabled, the module starts decoding tokens sent to the endpoint and replies with a active handshake signal if the received token is enabled.

### 35.4.6.6 Endpoint Disabled

#### Context

This section describes the steps required to disable an existing endpoint.

#### Procedure

- Step1.** To disable an endpoint, clear the USB active endpoint bit in the USBFS\_DIEPCTLx register (for IN or bidirectional endpoints) or the USBFS\_DOEPCTLx register (for OUT or bidirectional endpoints).
- Step2.** When an endpoint is disabled, the module ignores tokens sent to it, causing a USB timeout.



## 35.5 Operational Module

### 35.5.1 SETUP and OUT Data Transmission

This section describes the internal data flow and application operation steps during data OUT transfers and SETUP transactions.

#### Packet Reading

This section describes how to read data packets (OUT and SETUP data packets) from the receive FIFO.

1. When the RXFNE interrupt (USBFS\_GINTSTS register) is captured, the application must read the receive status pop up register (USBFS\_GRXSTSP).
2. The application can mask the RXFNE interrupt (in USBFS\_GINTSTS) by writing RXFNE=0 (in USBFS\_GINTMSK) until it reads the data packet from the receive FIFO.
3. If the byte counting of the received packet is not 0, pop up these data from the receive data FIFO and store them in memory. If the byte counting of the received packet is 0, no data will be popped from the receive data FIFO.
4. The packet status read from the receive FIFO has the following states:
  - Global OUT NAK:  
PKTSTS=Global OUT NAK, BCNT=0x000, the values of EPNUM and DPID are irrelevant.  
These data indicate that the global OUT NAK bit has taken effect.
  - SETUP packet:  
PKTSTS=SETUP, BCNT=0x008, EPNUM=Control EP number, DPID=D0. These data indicate that SETUP packets received on the specified endpoint can now be read from the receive FIFO.
  - Establishment phase completed:  
PKTSTS=Setup phase complete, BCNT=0x0, EPNUM=Control EP number, DPID value is irrelevant.  
This data indicates that the setup phase of the specified endpoint is complete and the data phase has been initiated. After this status entry pops from the receive FIFO, the module will generate a setup interrupt on the corresponding control OUT endpoint.
  - OUT packet:  
PKTSTS=DataOUT, BCNT=Size of received OUT data packet (BCNT: 0-1024), EPNUM=Endpoint number of the received data packet, DPID=Actual data PID.
  - Data transmission complete:  
PKTSTS=OUT data transmission complete, BCNT=0x0, EPNUM=OUT EP number that completed data transmission, DPID value is irrelevant.  
This data indicates the completion of OUT data transmission at the specified OUT endpoint. After this status entry pops from the receive FIFO, the module will trigger a "Transmission Complete" interrupt on the designated OUT endpoint.
5. After popping data from the receive FIFO, the RXFNE interrupt must be unmasked (USBFS\_GINTSTS).
6. Each time the application detects an RXFNE interrupt in USBFS\_GINTSTS, steps 1 through 5 will be repeated. Reading an empty receive FIFO may result in undefined module behavior.

#### SETUP Transaction

Describe the block's method for handling SETUP packets and the sequence for application processing of SETUP transactions.

#### Application Requirements:

1. To receive a SETUP packet, the control OUT endpoint STUPCNT field (USBFS\_DOEPTSIZE) must be programmed to a non-zero value. If the application programs the STUPCNT field to a non-zero value, the module will receive the SETUP packet and write it to the receive FIFO, regardless of the NAK status and the EPENA bit setting in USBFS\_DOEPCTL. For each SETUP packet received by the control endpoint, the STUPCNT field decrements. If the STUPCNT field is not programmed to the appropriate value before receiving the SETUP packet, the module can still receive the SETUP packet and decrement the STUPCNT field. However, the application may not be able to determine the correct number of SETUP packets received during the setup phase of the control transmission.
  - In USBFS\_DOEPTSIZE, STUPCNT=3
2. The application must always allocate some additional space in the receive data FIFO to accommodate up to three consecutive SETUP packets on the control endpoint.
  - Reserved space of 10 words. The first SETUP packet requires 3 words, the "Setup Phase Complete" status double-word requires 1 word, and an additional 6 words are needed to store two extra SETUP packets.
  - Each SETUP packet requires 3 words to store 8 bytes of SETUP data and 4 bytes of SETUP status. The module will reserve these spaces in the receive FIFO.
  - This FIFO is exclusively used for storing SETUP packets and will never utilize this space for data packets.
3. The application must read 2 words of SETUP data packet from the receive FIFO.
4. The application must read and discard the "Establishment Phase Complete" status word from the receive FIFO.

#### Internal Data Flow:

1. Upon receiving a SETUP packet, the module writes the received data into the receive FIFO without checking the available space in the receive FIFO, regardless of the endpoint's NAK and STALL bit settings.
  - The module internally sets the IN NAK and OUTNAK bits of the control IN/OUT endpoints that receive the SETUP packet to 1.
2. For each SETUP packet received over USB, the module writes 3 words of data into the receive FIFO and decrements the STUPCNT field by 1.
  - The first word contains internal control information used by the module.
  - The second word contains the first 4 bytes of the SETUP command.
  - The third word contains the last 4 bytes of the SETUP command.
3. When the setup phase ends and the data IN/OUT phase starts, the block writes a status entry ("Setup Phase Complete" word) into the receive FIFO, indicating the setup phase is complete.
4. On the AHB side, the SETUP packet is read by the application.
5. When the application pops the "Setup Phase Complete" word from the receive FIFO, the module will interrupt the application using the STUP interrupt (USBFS\_DOEPINT), indicating that it can process the received SETUP packet.
  - The module will clear the endpoint enable bit of control OUT endpoint.

#### Application Programming Sequence:

1. Program the USBFS\_DOEPTSIZE register.
  - STUPCNT=3
2. Wait for the RXFNE interrupt (USBFS\_GINTSTS) and read the data packet from the receive FIFO.
3. The STUP interrupt trigger (USBFS\_DOEPINT) indicates the successful completion of SETUP data transmission.

- When this interrupt occurs, the application must read the USBFS\_DOEPTSIZE register to determine the number of received SETUP packets and process the last received SETUP packet.

#### Processing More Than Three Consecutive SETUP Packets:

According to the USB2.0 specification, the host typically does not send more than three consecutive SETUP packets to the same endpoint during SETUP packet errors. However, the USB 2.0 specification does not limit the number of consecutive SETUP packets that the host can send to the same endpoint. When this occurs, the USBFS controller will generate an interrupt (B2BSTUP in USBFS\_DOEPINTx).

### Set Global OUT NAK to 1

#### Internal Data Flow:

1. If the application sets the global OUT NAK (SGONAK bit in USBFS\_DCTL) to 1, the module will stop writing any data other than SETUP packets to the receive FIFO. Regardless of the available space size in the receive FIFO, the device will respond with a NAK to asynchronous OUT tokens sent by the host, while directly ignoring synchronous OUT data packets.
2. The module writes the global OUT NAK to the receive FIFO. The application must reserve sufficient space for this.
3. When the application pops the global OUT NAK word from the receive FIFO, the block sets the GONAKEFF interrupt (USBFS\_GINTSTS) to 1.
4. When the application detects this interrupt, it will assume the module is in global OUT NAK mode. The application can clear this interrupt by clearing the SGONAK bit in USBFS\_DCTL.

#### Application Programming Sequence:

1. To stop receiving any type of data into the receive FIFO, the application must program the following fields to set the global OUT NAK bit to 1.
  - In USBFS\_DCTL, SGONAK=1
2. Wait for the GONAKEFF interrupt in USBFS\_GINTSTS. Once triggered, this interrupt indicates that the module has stopped receiving any type of data except SETUP packets.
  - If the application has set the SGONAK bit in USBFS\_DCTL to 1, the application can receive active OUT packets before the module generates a GONAKEFF interrupt (USBFS\_GINTSTS).
3. Applications can temporarily mask this interrupt by writing to the GINAKEFFM bit in the USBFS\_GINTMSK register.
  - In the USBFS\_GINTMSK register, GINAKEFFM=0.
4. When the application is ready to exit the global OUT NAK mode, the SGONAK bit in USBFS\_DCTL must be cleared. This operation also clears the GONAKEFF interrupt (USBFS\_GINTSTS).
  - In CGONAK, USBFS\_DCTL=1
5. If the application has previously masked this interrupt, the interrupt must be unmasked as follows:
  - In GINTMSK, GINAKEFFM=1

### Disable the enabled OUT endpoint

The application must disable the enabled OUT endpoint using the following sequence.

#### Application Programming Sequence:

1. Before disabling any OUT endpoint, the application must enable the global OUT NAK mode in the module.
  - In USBFS\_DCTL, SGONAK=1
2. Wait for the GONAKEFF interrupt (USBFS\_GINTSTS)
3. Disable the OUT endpoint by programming the following fields:

- In USBFS\_DOEPCTLx, EPDIS=1
  - In USBFS\_DOEPCTLx, SNAK=1
4. Wait for the EPDISD interrupt (USBFS\_DOEPINTx), which indicates that the OUT endpoint has been completely disabled. When an EPDISD interrupt is triggered, the module will also clear the following bits:
    - In USBFS\_DOEPCTLx, EPDIS=0
    - In USBFS\_DOEPCTLx, EPENA=0
  5. The application must clear the global OUT NAK bit to start receiving data from other OUT endpoints that are not disabled.
    - In USBFS\_DCTL, SGONAK=0

### Universal Asynchronous OUT Data Transmission

This section describes a general asynchronous OUT data transfer (control, bulk, or interrupt).

#### Application Requirements:

1. Before establishing an OUT transfer, the application must allocate a buffer in memory to hold all the data to be received as part of the OUT transfer.
2. For OUT transfer, the transfer size field in the endpoint's transfer size register must be a multiple of the endpoint's maximum packet size (and word-aligned).
  - $\text{Transfer Size}[\text{EPNUM}] = n \times (\text{MPSIZ}[\text{EPNUM}] + 4 - (\text{MPSIZ}[\text{EPNUM}] \bmod 4))$
  - $\text{Packet Count}[\text{EPNUM}] = n$
  - $n > 0$
3. When an OUT endpoint interrupt occurs, the application must read the endpoint's transfer size register to calculate the amount of active data in memory. The received active data amount may be less than the programmed transfer size.
  - $\text{Active data amount in memory} = \text{Initial transfer amount set by application program} - \text{remaining transfer amount after module update}$
  - $\text{Number of received USB data packets} = \text{Initial number of packets set by application} - \text{remaining number of packets after module update}$

#### Internal Data Flow:

1. The application must set the transfer size and packet count fields in the endpoint-related register, clear the NAK bit, and enable the endpoint to receive data.
2. After the NAK bit is cleared, the module starts to receive data and writes the data into the receive FIFO (as long as there is space in the receive FIFO). For each data packet received over USB, the packet and its status are written to the receive FIFO. Each packet (either a packet with data volume reaching the maximum packet size or a short packet) written to the receive FIFO will decrement the packet count field of this endpoint by 1.
  - If the received packet has an inactive CRC, it is automatically cleared from the receive FIFO.
  - After replying with an ACK for the data packet on USB, the module discards the non-isochronous OUT packet retransmitted by the host due to undetected ACK. The application will not detect multiple consecutive OUT packets on the same endpoint with identical data PIDs. In this case, the packet count does not decrement.
  - If there is no space in the receive FIFO, isochronous or asynchronous packets will be ignored and not written into the receive FIFO. In addition, an asynchronous OUT token will receive a NAK handshake response.

- In all three scenarios mentioned above, the packet count will not decrement as no data is written to the receive FIFO.
3. When the packet count becomes 0 or a short packet is received on the endpoint, the NAK bit of that endpoint will be set to 1. When the NAK bit is set to 1, both synchronous and asynchronous data packets will be ignored and not written into the receive FIFO. Additionally, asynchronous OUT tokens will receive a NAK handshake response.
  4. After data is written to the receive FIFO, the application will read data from the receive FIFO and write it to external storage, one packet at a time, endpoint by endpoint.
  5. After writing each data packet to the external storage via AHB, the endpoint's transmission size is automatically decremented by the size of the data packet.
  6. Under the following conditions, the OUT data transmission completion status of the OUT endpoint will be written to the receive FIFO:
    - Transfer size is 0 and packet count is 0
    - The last OUT data packet written to the receive FIFO is a short packet (Packet size: 0 to maximum packet size-1)
  7. When the application pops this status entry (OUT data transfer completed) and generates a transfer completed interrupt for this endpoint, it also clears the endpoint enable bit.

#### Application Programming Sequence:

1. Program the USBFS\_DOEPTSIZE register using the transfer size and corresponding packet count.
2. Program the USBFS\_DOEPCTL register using endpoint features, and set the EPENA and CNAK bits to 1.
  - In USBFS\_DOEPCTL, EPENA=1
  - In USBFS\_DOEPCTL, CNAK=1
3. Wait for the RXFNE interrupt (in USBFS\_GINTSTS) and read the data packet from the receive FIFO.
  - This step can be repeated multiple times, depending on the transfer size.
4. Trigger the XFRC interrupt (USBFS\_DOEPINT) to indicate the successful completion of an asynchronous OUT data transmission.
5. Read the USBFS\_DOEPTSIZE register to determine the active data amount.

#### Universal Synchronous OUT Data Transmission

This section describes regular synchronous OUT data transmission.

#### Application Requirements:

1. All application requirements for asynchronous OUT data transmissions apply equally to synchronous OUT data transmission
2. For synchronous OUT data transmissions, the transmission size and packet count fields must always be set to the number of packets corresponding to the maximum packet size receivable in a single frame. Synchronous OUT data transmission transactions must be completed within a single frame.
3. Before the periodic frame ends (EOPF interrupt in USBFS\_GINTSTS), the application must read all synchronous OUT packets (data entries and status entries) from the receive FIFO.
4. To receive data in the next frame, a synchronous OUT endpoint must be enabled after EOPF (USBFS\_GINTSTS) and before SOF (USBFS\_GINTSTS).

#### Internal Data Flow:

1. The internal data flow of synchronous OUT endpoints is essentially the same as that of asynchronous OUT endpoints, with minor differences.

2. When enabling a synchronous OUT endpoint by setting the endpoint enable bit to 1 and clearing the NAK bit, the corresponding even/odd frame bit must also be set to 1. The module will receive data from a specific frame on the synchronized OUT endpoint only when the following conditions are met:
  - EONUM (in USBFS\_DOEPCTLx)=FNSOF[0] (in USBFS\_DSTS)
3. When the application reads a complete synchronous OUT packet (data and status) from the receive FIFO, the module updates the RXDPID field in USBFS\_DOEPTSIZx based on the data PID of the last synchronous OUT packet read from the receive FIFO.

#### Application Programming Sequence:

1. Program the USBFS\_DOEPTSIZx register using the transmission size and corresponding packet count.
2. Program the USBFS\_DOEPCTLx register using endpoint features, and set the endpoint enable bit, clear NAK bit, and odd/even frame bit to 1.
  - EPENA1
  - CNAK=1
  - EONUM=(0: Even/1: Odd)
3. Wait for the RXFNE interrupt (in USBFS\_GINTSTS) and read the data packet from the receive FIFO.
  - This step can be repeated multiple times, depending on the transfer size.
4. XFRC interrupt (in USBFS\_DOEPINTx) indicates the completion of a synchronous OUT data transmission. This interrupt does not necessarily indicate that the data in memory is enabled.
5. For synchronous OUT transfers, the application may not always detect this interrupt. Instead, the application may detect the IISOXFRM interrupt in USBFS\_GINTSTS.
6. Read the USBFS\_DOEPTSIZx register to determine the received transmission size and verify the activity of the received data within the frame. The application shall consider the received data in memory as active only when one of the following conditions is met:
  - RXDPID=D0 (in USBFS\_DOEPTSIZx) and the number of active USB data packets received=1
  - RXDPID=D1 (in USBFS\_DOEPTSIZx) and the number of active USB data packets received=2
  - RXDPID=D2 (in USBFS\_DOEPTSIZx) and the number of active USB data packets received=3

The number of USB packets receiving active data=Initial packet count programmed by the application-remaining packet count after module update.

The application can discard inactive packets.

#### Incomplete Synchronous OUT Data Transmission

This section describes the application programming sequence when packet loss occurs in synchronous OUT data packets.

##### Internal Data Flow:

1. For synchronous OUT endpoints, the XFRC interrupt (in USBFS\_DOEPINTx) may not always be triggered. If the module discards the synchronized OUT data packet, the application may fail to detect the XFRC interrupt (USBFS\_DOEPINTx) under the following conditions:
  - When the receive FIFO cannot accommodate a complete ISO OUT packet, the module will discard the received ISO OUT data.
  - The received synchronization OUT packet has a CRC error.
  - The module received a corrupted synchronization OUT token.
  - The application reads data from the receive FIFO at a very slow speed.
2. If the block detects a periodic frame end before the transfer completion of all synchronous OUT endpoints, it will trigger an incomplete synchronous OUT data interrupt (IISOXFRM in USBFS\_GINTSTS),



indicating that at least one synchronous OUT endpoint did not trigger an XFRC interrupt (in USBFS\_DOEPINTx). At this time, endpoints with unfinished transfers remain enabled, but there are no ongoing active transfers on this USB endpoint.

#### Application Programming Sequence:

1. Hardware triggers the IISOXFRM interrupt (USBFS\_GINTSTS) indicating that at least one synchronous OUT endpoint has an incomplete transfer in the current frame.
2. If this occurs due to incomplete reading of synchronous OUT data from the endpoint, the application must ensure all synchronous OUT data (including data entries and status entries) are first read from the receive FIFO before proceeding with further processing.
  - After reading all data from the receive FIFO, the application can detect the XFRC interrupt (USBFS\_DOEPINTx). In this case, the application must re-enable the endpoint to receive synchronous OUT data in the next frame.
3. When the application receives the IISOXFRM interrupt (in USBFS\_GINTSTS), it must read the control registers (USBFS\_DOEPCTLx) of all synchronous OUT endpoints to determine which endpoints have incomplete transmission in the current frame. When both of the following conditions are met simultaneously, it indicates that the endpoint transfer is incomplete:
  - EONUM bit (in USBFS\_DOEPCTLx)=FNSOF[0] (in USBFS\_DSTS)
  - EPENA=1 (in USBFS\_DOEPCTLx)
4. Before detecting the SOF interrupt (in USBFS\_GINTSTS), the previous operation must be fully completed to ensure the current frame number has not changed.
5. For synchronous OUT endpoints with incomplete transmission, the application must discard the data in memory and disable the endpoint by setting the EPDIS bit in USBFS\_DOEPCTLx to 1.
6. Wait for the EPDIS interrupt (in USBFS\_DOEPINTx), and enable the endpoint to receive new data in the next frame.
  - Since the module may require some time to disable the endpoint, the application may fail to receive data in the next frame after detecting inactive synchronization data.

#### Disable Asynchronous OUT Endpoint

This section describes how the application can stop an asynchronous endpoint.

1. Place the module in global OUT NAK mode.
2. Disable the required endpoint
  - When disabling an endpoint, please set STALL=1 (in USBFS\_DOEPCTL) instead of setting the SNAK bit to 1 in USBFS\_DOEPCTL. The priority of the STALL bit is always higher than that of the NAK bit.
3. When the application no longer needs the endpoint to respond with a STALL handshake, the STALL bit (in USBFS\_DOEPCTLx) must be cleared to zero.
4. If an application receives a SetFeature.Endpoint Halt or ClearFeature.Endpoint Halt command from the host to set or clear the STALL state of an endpoint, it must set the STALL bit to 1 or clear it on the control endpoint before initiating the status phase transfer.

### 35.5.2 IN Data Transmission

#### Packet Write

This section describes how the application writes data packets to the endpoint FIFO when the dedicated transmit FIFO is enabled.

1. The application can choose between polling mode or interrupt mode.

- In polling mode, the application monitors the status of the endpoint transmit data FIFO by reading the USBFS\_DTXFSTSx register to determine whether there is sufficient space in the data FIFO.
  - In interrupt mode, the application waits for the TXFE interrupt (in USBFS\_DIEPINTx), then reads the USBFS\_DTXFSTSx register to determine if there is sufficient space in the data FIFO.
  - To write a single non-zero length packet, there must be sufficient space in the data FIFO to accommodate the entire packet.
  - To write a zero length packet, the application cannot check the FIFO space.
2. If one of the above methods is used, when the application determines there is sufficient space to write transmit data packets, the application must first perform the corresponding write operation to the endpoint control register before writing data to the data FIFO. Typically, the application must perform a read-modify-write operation on the USBFS\_DIEPCTLx register to avoid modifying other contents in the register while setting the endpoint enable bit to 1.  
If sufficient space is available, the application can write multiple data packets for the same endpoint into the transmit FIFO. For periodic IN endpoints, the application can only write multiple data packets within a single frame at a time. The application will only write all packets to be sent in the next frame after the previous frame's communication transaction has completed.

### Set IN Endpoint NAK to 1

#### Internal Data Flow:

1. When the application sets the IN NAK of a specific endpoint to 1, the module will stop data transmission on that endpoint, regardless of whether data is available in the endpoint's transmit FIFO.
2. The asynchronous endpoint receives the IN token and replies with a NAK handshake response.
  - The synchronous endpoint receives an IN token and returns a zero length data packet.
3. The module triggers the INEPNE (IN endpoint NAK active) interrupt in USBFS\_DIEPINTx in response to the SNAK bit in USBFS\_DIEPCTLx.
4. When the application detects this interrupt, it will consider the endpoint to be in IN NAK mode. The application can clear this interrupt by setting the CNAK bit in USBFS\_DIEPCTLx to 1.

#### Application Programming Sequence:

1. To stop sending any data on a specific IN endpoint, the application must set the IN NAK bit to 1. To set this bit to 1, the following fields must be programmed.
  - SNAK=1 in USBFS\_DIEPCTLx
2. Wait for the INEPNE interrupt in USBFS\_DIEPINTx to trigger. This interrupt indicates the module has stopped transmitting data on the endpoint.
3. The module can send active IN data on the endpoint when the application sets the NAK bit to 1 but the "NAK Active" interrupt has not yet been triggered.
4. The application can temporarily mask this interrupt by writing to the INEPNEM bit in DIEPMSK.
  - In DIEPMSK, INEPNEM=0
5. To exit endpoint NAK mode, the application must clear the NAK status bit (NAKSTS) in USBFS\_DIEPCTLx. This operation also clears the INEPNE interrupt (in USBFS\_DIEPINTx).
  - In USBFS\_DIEPCTLx, CNAK=1
6. If the application has masked this interrupt, it must be unmasked as follows:
  - In DIEPMSK, INEPNEM=1

### Disable IN Endpoint

Use the following sequence to disable a previously enabled specific IN endpoint.

#### Application Programming Sequence:



1. The application must first stop writing data on the AHB before disabling the IN endpoint.
2. The application must set the endpoint to NAK mode.
  - SNAK=1 in USBFS\_DIEPCTLx
3. Wait for the INEPNE interrupt in USBFS\_DIEPINTx.
4. The following bits in the USBFS\_DIEPCTLx register of the endpoint that must be disabled must be set to 1.
  - EPDIS=1 in USBFS\_DIEPCTLx
  - SNAK=1 in USBFS\_DIEPCTLx
5. The EPDISD interrupt in USBFS\_DIEPINTx indicates that the block has completely disabled the specified endpoint. At the same time the interrupt is triggered, the module also clears the following bits:
  - In USBFS\_DIEPCTLx, EPENA=0
  - In USBFS\_DIEPCTLx, EPDIS=0
6. The application must read the USBFS\_DIEPTSIZx register for periodic IN EP to calculate the amount of data transmitted on the endpoint via USB.
7. The application must clear the data in the endpoint transmit FIFO by setting the following fields in the USBFS\_GRSTCTL register to 1:
  - TXFNUM (in USBFS\_GRSTCTL)=Transmit FIFO number of the endpoint
  - TXFFLSH (in USBFS\_GRSTCTL)=1

The application must poll the USBFS\_GRSTCTL register until the module clears the TXFFLSH bit, indicating the FIFO flush operation has ended. To send new data on this endpoint, the application can re-enable the endpoint later.

### Universal Non periodic IN Data Transmission

#### Application Requirements:

1. Before establishing an IN transmission, the application must ensure that each data packet comprising an IN transmission, can fit into a single buffer.
2. For IN transmission,s, the transmission size field in the endpoint transmission size register indicates the active data amount of the current transmission, which consists of multiple maximum packet sizes and a single short packet. The short data packet is sent at the end of transmission.
  - To send multiple packets of the maximum packet size and append a short packet at the end of transmission.  
 Transmission size [EPNUM]=x×MPSIZ[EPNUM] + sp  
 If (sp > 0), packet count[EPNUM]=x + 1.  
 Otherwise, packet count [EPNUM]=x
  - To send a single zero length data packet:  
 Transmission size[EPNUM]=0  
 Packet count[EPNUM]=1
  - To send multiple packets of the maximum packet size and append a zero length packet at the end of transmission, the application must split the transfer into two parts.  
 The first part transmits a packet with the maximum packet size, while the second part transmits only a zero length packet.  
 First transfer: transmission size[EPNUM]=x×MPSIZ[epnum]; Packet count=n;  
 Second transfer: transmission size[EPNUM]=0; Packet count=1;

3. After enabling an endpoint for data transmission, the module updates the transmission size register. At the IN transmission end, the application must read the transmission size register to determine how much data in the transmit FIFO has been sent out via USB.
4. The amount of data written to the transmit FIFO=Initial transmission size programmed by the application-Final transmission size after module update.
  - Amount of data sent via USB=(Initial packet count programmed by application-Final packet count after module update) ×MPSIZ[EPNUM]
  - Remaining data to be sent via USB=(Initial transmission size programmed by application-Data already sent via USB)

#### Internal Data Flow:

1. The application must set the transmission size and packet count fields in the specific endpoint's register, and enable the endpoint to transmit data.
2. The application must also write the required data to the transmit FIFO of this endpoint.
3. Each time the application writes a data packet to the transmit FIFO, the transmission size of this endpoint is automatically decremented by the size of the data packet. The application continuously fetches data from memory to write into the transmit FIFO until the transmission size of this endpoint becomes 0. After writing data to the FIFO, the "packet count in FIFO" counter increments (this is a 3-bit counter maintained internally by the module, with one counter per IN endpoint transmit FIFO. In the IN endpoint FIFO, the maximum number of data packets maintained by the module is always eight.) For zero length packets, each FIFO has an additional dedicated flag indicating no data is present in the FIFO.
4. When data is written to the transmit FIFO, the module will send out this data upon receiving an IN token. For each packet sent out and receiving an ACK handshake signal in response, the packet count of this endpoint will decrement by 1 until the packet count reaches 0. Upon timeout, the packet count does not decrement.
5. For zero length packets (indicated by the internal zero length flag), the module issues a zero length packet in response to the IN token and decrements the packet count field value.
6. If an IN token is received for an endpoint whose corresponding FIFO has no data and its packet count field is zero, the module will generate an "IN token received when TxFIFO is empty" (ITTXFE) interrupt for that endpoint (provided the NAK bit of the endpoint is not set to 1). The module returns a NAK handshake signal on this asynchronous endpoint.
7. The module internally resets the FIFO pointer to the beginning and does not generate a timeout interrupt.
8. When the transmission size is 0 and the packet count is 0, the transmission complete (XFRC) interrupt for this endpoint will be generated, and the endpoint enable will be cleared.

#### Application Programming Sequence:

1. Program the USBFS\_DIEPTSIZx register using the transmission size and corresponding packet count.
2. Program the USBFS\_DIEPCTLx register using endpoint features, and set the CNAK and EPENA (Endpoint enable) bits to 1.
3. When sending a non-zero length data packet, the application must poll the USBFS\_DTXFSTSx register (where x is the FIFO number associated with the endpoint) to determine whether there is sufficient space in the data FIFO. Before writing data, the application can also use the TXFE bit (in USBFS\_DIEPINTx).

#### Universal Periodic IN Data Transmission

This section describes typical periodic IN data transmission

**Application Requirements:**

1. The application requirements 1, 2, 3, and 4 for universal non-periodic IN data transmission also apply to periodic IN data transmission (with slight modification to requirement 2).
  - The application can only send a number of data packets with the maximum packet size or several packets with the maximum packet size, plus a short packet at the end of transmission.  
To send multiple packets of the maximum packet size followed by a short packet at the end of transmission, the following condition must be met:  
Transmission size [EPNUM] =  $x \times \text{MPSIZ}[\text{EPNUM}] + \text{sp}$   
(Where  $x$  is an integer greater than 0, and the range of  $\text{sp}$  is  $0 - \text{MPSIZ}[\text{EPNUM}] - 1$ )  
If ( $\text{sp} > 0$ ), packet count [EPNUM] =  $x + 1$   
Otherwise, packet count [EPNUM] =  $x$ ;  
MCNT [EPNUM] = packet count [EPNUM]
  - The application cannot send a zero length packet at the end of transmission. The application can independently send a zero length data packet.
  - To send a single zero length data packet:  
Transmission size [EPNUM] = 0  
Packet count [EPNUM] = 1  
MCNT [EPNUM] = packet count [EPNUM]
2. The application can only schedule one frame of data transmission at a time.
  - $(\text{MCNT} - 1) \times \text{MPSIZ} \leq \text{XFERSIZ} \leq \text{MCNT} \times \text{MPSIZ}$
  - $\text{PKTCNT} = \text{MCNT}$  (in USBFS\_DIEPTSIZE $_x$ )
  - If  $\text{XFERSIZ} < \text{MCNT} \times \text{MPSIZ}$ , the last transmitted packet will be a short packet.
  - Note: MCNT is in USBFS\_DIEPTSIZE $_x$ , MPSIZ is in USBFS\_DIEPCTL $_x$ , PKTCNT is in USBFS\_DIEPTSIZE $_x$ , XFERSIZ is in USBFS\_DIEPTSIZE $_x$ .
3. Before receiving the IN token, the application must write the complete data to be sent in the frame into the transmit FIFO. Upon receiving an IN token, the module will perform the operation for an empty transmit FIFO even if only 1 double word of data remains to be written for the frame to be transmitted. When the transmit FIFO is empty:
  - The synchronous endpoint will respond with a zero length data packet.
  - The interrupt endpoint will respond with a NAK handshake signal.

**Internal Data Flow:**

1. The application must set the transmission size and packet count fields in the specific endpoint's register, and enable the endpoint to transmit data.
2. The application must also write the required data to the transmit FIFO associated with this endpoint.
3. Each time the application writes a data packet to the transmit FIFO, the transmission size of this endpoint is automatically decremented by the size of the data packet. The application continuously fetches data from memory to write into the transmit FIFO until the transmission size of this endpoint becomes 0.
4. When the periodic endpoint receives an IN token, the module will start transmitting data from the FIFO (if data exists in the FIFO). If there is no complete data packet in the FIFO for the frame to be transmitted, the module will generate an "IN token received when TxFIFO is empty" interrupt for this endpoint.
  - The synchronous endpoint will respond with a zero length data packet.
  - The interrupt endpoint will respond with a NAK handshake signal.

5. The packet count for an endpoint is decremented by 1 when:
  - For synchronous endpoints, when transmitting a zero length or non-zero length data packet
  - For interrupt endpoints, decrement when sending the ACK handshake signal.
  - When both the transmission size and packet count are 0, a transmission completed interrupt will be generated for this endpoint, and the endpoint enable bit will be cleared.
6. Within the "Periodic Frame Interval" (controlled by the PFIVL bit in USBFS\_DCFG), the module generates an IISOIXFR interrupt in USBFS\_GINTSTS whenever it detects that data in any synchronous IN endpoint FIFO—which should be empty in the current frame—has not been fully transmitted.

#### Application Programming Sequence:

1. Program the USBFS\_DIEPCTLx register using endpoint features, and set the CNAK and EPENA bits to 1.
2. Write the data to be transmitted in the next frame into the transmit FIFO.
3. Hardware triggers the ITTXFE interrupt (in USBFS\_DIEPINTx) indicating the application has not written all required data to the transmit FIFO.
4. If the interrupt endpoint is already enabled before detecting the interrupt, the interrupt will be ignored. If the interrupt endpoint is not enabled, enable this endpoint so that data can be sent upon receiving the next IN token.
5. When the hardware triggers the XFRC interrupt (in USBFS\_DIEPINTx) and no ITTXFE interrupt is generated in USBFS\_DIEPINTx, it indicates the success of a synchronized IN transmission,. When reading the USBFS\_DIEPTSIZx register, if the transmission size=0 and packet count=0 are always obtained, it indicates all data has been transmitted via USB.
6. Setting the XFRC interrupt (in USBFS\_DIEPINTx) indicates the successful completion of an interrupt IN transmission, regardless of whether the ITTXFE interrupt (in USBFS\_DIEPINTx) is generated. When reading the USBFS\_DIEPTSIZx register, if the transmission size=0 and packet count=0 are always obtained, it indicates all data has been transmitted via USB.
7. In the USBFS\_GINTSTS register, if the incomplete synchronous IN transmission (IISOIXFR) interrupt is set without any of the aforementioned interrupts being generated, it indicates that the module did not receive at least one periodic IN token in the current frame.

#### Incomplete Synchronization IN Data Transmission

This section describes the operations that the application must perform for incomplete synchronous IN data transmission

##### Internal Data Flow:

1. The synchronization IN transmission is considered incomplete when any of the following conditions are met:
  - The module received a corrupted sync IN token on at least one sync IN endpoint. At this point, the application detects an incomplete synchronous IN transmission interrupt (IISOIXFR bit in USBFS\_GINTSTS).
  - The application writes data to the transmit FIFO too slowly, and receives an IN token before completing the full data write to the FIFO. At this point, the application detects the "IN token received when TxFIFO is empty" interrupt in USBFS\_DIEPINTx. The application can ignore this interrupt, as it will eventually generate an incomplete synchronous IN transmission interrupt (IISOIXFR bit in USBFS\_GINTSTS) at the end of the periodic frame. The module will respond to the received IN token by sending a zero length data packet via USB.
2. The application must stop writing data to the transmit FIFO as soon as possible.

3. The application must set the NAK bit and disable bit of the endpoint.
4. The module will disable this endpoint, clear the disable bit, and trigger the "Endpoint Disable" interrupt.

#### Application Programming Sequence:

1. The application can ignore the "IN token received when TxFIFO is empty" interrupt in USBFS\_DIEPINTx for any isochronous IN endpoint, as this will ultimately generate an incomplete synchronous IN transmission interrupt (in USBFS\_GINTSTS).
2. Hardware-triggered incomplete synchronous IN transmission interrupt (in USBFS\_GINTSTS) indicates there is at least one incomplete synchronous IN transmission, on a synchronous IN endpoint.
3. The application must read the "Endpoint Control" register of all synchronized IN endpoints to detect endpoints with incomplete IN data transmission
4. The application must stop writing data to the "Periodic Transmit FIFO" associated with these endpoints.
5. Program the following fields in the USBFS\_DIEPCTLx register to disable the endpoint:
  - SNAK=1 in USBFS\_DIEPCTLx
  - EPDIS=1 in USBFS\_DIEPCTLx
6. Hardware triggers the "Endpoint Disable" interrupt in USBFS\_DIEPINTx, indicating the module has disabled this endpoint.
  - At this point, the application must either empty the data in the associated transmit FIFO or overwrite the existing data in the FIFO by enabling a new transmission endpoint in the next frame. To refresh the data, the application must use the USBFS\_GRSTCTL register.

### Stop Asynchronous IN Endpoint

This section describes how the application can stop an asynchronous endpoint.

#### Application Programming Sequence:

1. Disable the IN endpoint to be stopped. Set the STALL bit to 1 simultaneously.
2. EPDIS=1 in USBFS\_DIEPCTLx (when the endpoint is enabled)
  - STALL=1 in USBFS\_DIEPCTLx
  - The priority of the STALL bit is always higher than that of the NAK bit.
3. A hardware-triggered "Endpoint Disable" interrupt (in USBFS\_DIEPINTx) notifies the application that the module has disabled the specified endpoint.
4. The application must empty the non-periodic or periodic transmit FIFO based on the endpoint type. For non-periodic endpoints, the application must re-enable another non-periodic endpoint without stopping to transmit data.
5. When the application is ready to end the STALL handshake signal for this endpoint, it must clear the STALL bit in USBFS\_DIEPCTLx.
6. If the application sets or clears the STALL state of an endpoint upon receiving a SetFeature.Endpoint Halt or ClearFeature.Endpoint Halt command from the host, the STALL bit must be set to 1 or cleared before the status phase transfer of the control endpoint.

#### Special Case: Stop Controlling the OUT Endpoint

During the data phase of a control transmission, if the number of IN/OUT tokens sent by the host exceeds the value specified in the SETUP packet, the module must respond with STALL to these excess IN/OUT tokens. In this scenario, the application must enable the ITTXFE interrupt of USBFS\_DIEPINTx and the OTEPDIS interrupt of USBFS\_DOEPINTx during the data phase of the control transmission (when the module has completed transmitting the data amount specified by the SETUP packet). Subsequently,

when the application receives this interrupt, it must set the STALL bit in the corresponding endpoint control register to 1 and clear this interrupt.

## 35.6 Register Description

The application controls the USBFS module by performing read and write operations on the control and status registers through the AHB slave interface. All registers in the USBFS module are 32-bit registers with 32-bit aligned addresses, and can only be accessed in 32-bit mode.

Control and status registers are categorized as follows:

- USBFS System Control Register
- Module Global Register
- Host Mode Register
- Device Mode Register
- Power and Clock Gating Control Registers
- Data FIFO(DFIFO) Access Register

The USBFS system control register has a different base address from other registers. This register is independent of the USBFS module and controls the relevant settings of the USBFS module.

Only the module global register, power and clock gating control registers, and data FIFO access registers can be accessed in both host and device modes. When the USBFS module operates in one mode (host or device), the application must not access registers in the alternate role mode, such as accessing device mode registers while in host mode. If an illegal access occurs, a mode mismatch interrupt will be generated and reflected in the module interrupt register USBFS.GINTSTS.NMIS bit. When the module switches from one role mode to another, the registers in the new operating mode must be reprogrammed to the power on reset state.

### Control Status Register Memory Mapping

The host and device mode registers occupy different addresses. All registers are implemented within the AHB clock domain.

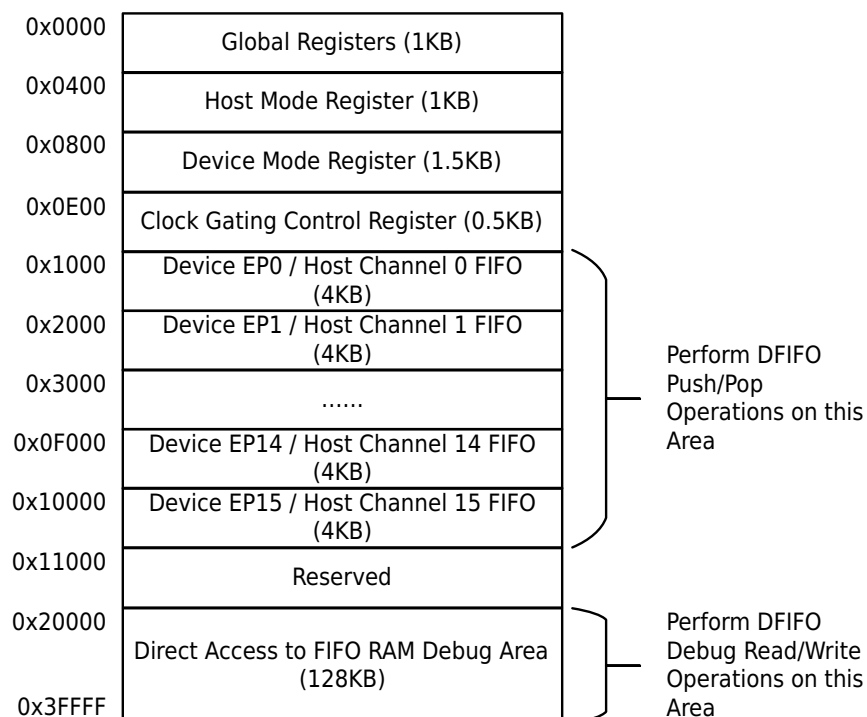


Figure 35-7 USBFS Control and Status Register Memory Mapping

### 35.6.1 Register List

USB system control register base address: 0x40055400

Table 35-3 USB System Control Register Offset

| Offset | Register     | Description                 | Width |
|--------|--------------|-----------------------------|-------|
| 0x00   | USB_SYCTLREG | USB System Control Register | 32    |

USBFS module register base address: 0x40080000

Table 35-4 USB2.0 Full-Speed Module USBFS Register Offset

| Offset      | Register        | Description                                             | Width |
|-------------|-----------------|---------------------------------------------------------|-------|
| 0x00        | USBFS_GVBUSCFG  | VBUS Control Register                                   | 32    |
| 0x08        | USBFS_GAHBCFG   | AHB Control Register                                    | 32    |
| 0x0C        | USBFS_GUSBCFG   | USB Configuration Register                              | 32    |
| 0x10        | USBFS_GRSTCTL   | Reset Register                                          | 32    |
| 0x14        | USBFS_GINTSTS   | Global Interrupt Status Register                        | 32    |
| 0x18        | USBFS_GINTMSK   | Global Interrupt Mask Register                          | 32    |
| 0x1C        | USBFS_GRXSTSR   | Receive Status Debug Read Register                      | 32    |
| 0x20        | USBFS_GRXSTSP   | Receive Status Read and Pop Register                    | 32    |
| 0x24        | USBFS_GRXFSIZ   | Receive FIFO Size Register                              | 32    |
| 0x28        | USBFS_HNPTXFSIZ | Host Non Periodic Transmit FIFO Size Register           | 32    |
| 0x28        | USBFS_DIEPTXF0  | Device Endpoint 0 Transmit FIFO Size Register           | 32    |
| 0x2C        | USBFS_HNPTXSTS  | Non-periodic Transmit FIFO/Queue Status Register        | 32    |
| 0x3C        | USBFS_CID       | Module ID Register                                      | 32    |
| 0x54        | USBFS_GLPMCFG   | LPM Configuration Register                              | 32    |
| 0x100       | USBFS_HPTXFSIZ  | Host Periodic Transmit FIFO Size Register               | 32    |
| 0x100+0x4*x | USBFS_DIEPTXFx  | Device IN Endpoint Transmit FIFO Size Register (x=1-15) | 32    |
| 0x400       | USBFS_HCFG      | Host Configuration Register                             | 32    |
| 0x404       | USBFS_HFIR      | Host Frame Time Interval Register                       | 32    |
| 0x408       | USBFS_HFNUM     | Host Frame Number/Frame Remaining Time Register         | 32    |
| 0x410       | USBFS_HPTXSTS   | Host Periodic Transmit FIFO/Queue Status Register       | 32    |
| 0x414       | USBFS_HAINT     | Host All Channels Interrupt Register                    | 32    |



| Offset       | Register         | Description                                           | Width |
|--------------|------------------|-------------------------------------------------------|-------|
| 0x418        | USBFS_HAINTMSK   | Host All Channels Interrupt Mask Register             | 32    |
| 0x440        | USBFS_HPRT       | Host Port Control and Status Register                 | 32    |
| 0x500+0x20*x | USBFS_HCCHARx    | Host Channel x Characteristic Register                | 32    |
| 0x508+0x20*x | USBFS_HCINTx     | Host Channel x Interrupt Register                     | 32    |
| 0x50C+0x20*x | USBFS_HCINTMSKx  | Host Channel x Interrupt Mask Register                | 32    |
| 0x510+0x20*x | USBFS_HCTSIZx    | Host Channel x Transfer Size Register                 | 32    |
| 0x514+0x20*x | USBFS_HCDMAX     | Host Channel x DMA Address Register                   | 32    |
| 0x800        | USBFS_DCFG       | Device Configuration Register                         | 32    |
| 0x804        | USBFS_DCTL       | Device Control Register                               | 32    |
| 0x808        | USBFS_DSTS       | Device Status Register                                | 32    |
| 0x810        | USBFS_DIEPMSK    | Device IN Endpoint General Interrupt Mask Register    | 32    |
| 0x814        | USBFS_DOEPMSK    | Device OUT Endpoint General Interrupt Mask Register   | 32    |
| 0x818        | USBFS_DAIN       | Device All Endpoints Interrupt Register               | 32    |
| 0x81C        | USBFS_DAINMSK    | Device All Endpoints Interrupt Mask Register          | 32    |
| 0x834        | USBFS_DIEPEMPMSK | Device IN Endpoint FIFO Empty Interrupt Mask Register | 32    |
| 0x900        | USBFS_DIEPCTL0   | Device IN Endpoint 0 Control Register                 | 32    |
| 0x900+0x20*x | USBFS_DIEPCTLx   | Device IN Endpoint x Control Register (x=1-15)        | 32    |
| 0x908+0x20*x | USBFS_DIEPINTx   | Device IN Endpoint x Interrupt Register               | 32    |
| 0x910        | USBFS_DIEPTSIZ0  | Device IN Endpoint 0 Transfer Size Register           | 32    |
| 0x910+0x20*x | USBFS_DIEPTSIZx  | Device IN Endpoint x Transfer Size Register (x=1-15)  | 32    |
| 0x914+0x20*x | USBFS_DIEPDMAx   | Device IN Endpoint x DMA Address Register             | 32    |
| 0x918+0x20*x | USBFS_DTXFSTSx   | Device IN Endpoint x Transmit FIFO Status Register    | 32    |
| 0xB00        | USBFS_DOEPCTLx   | Device OUT Endpoint 0 Control Register                | 32    |
| 0xB00+0x20*x | USBFS_DOEPCTLx   | Device OUT Endpoint x Control Register (x=1-15)       | 32    |
| 0xB08+0x20*x | USBFS_DOEPINTx   | Device OUT Endpoint x Interrupt Register              | 32    |
| 0xB10        | USBFS_DOEPSIZ0   | Device OUT Endpoint 0 Transfer Size Register          | 32    |

| Offset       | Register       | Description                                           | Width |
|--------------|----------------|-------------------------------------------------------|-------|
| 0xB10+0x20*x | USBFS_DOEPSIZx | Device OUT Endpoint x Transfer Size Register (x=1-15) | 32    |
| 0xB14+0x20*x | USBFS_DOEPDMAx | Device OUT Endpoint x DMA Address Register            | 32    |
| 0xE00        | USBFS_GCCTL    | Clock Gating Control Register                         | 32    |

**Note**

In the table, x represents there are multiple registers of the same type, with values ranging from 0 to 15. Special identifiers are subject to the description in the register row.

## 35.6.2 USB System Control Register

### 35.6.2.1 USB System Control Register (USB\_SYCTLREG)

| Offset | Bit Position |    |    |    |           |    |           |    |          |    |    |    |           |    |           |    |          |    |    |    |              |    |             |   |           |   |          |   |   |   |             |   |           |  |
|--------|--------------|----|----|----|-----------|----|-----------|----|----------|----|----|----|-----------|----|-----------|----|----------|----|----|----|--------------|----|-------------|---|-----------|---|----------|---|---|---|-------------|---|-----------|--|
| 0x00   | 31           | 30 | 29 | 28 | 27        | 26 | 25        | 24 | 23       | 22 | 21 | 20 | 19        | 18 | 17        | 16 | 15       | 14 | 13 | 12 | 11           | 10 | 9           | 8 | 7         | 6 | 5        | 4 | 3 | 2 | 1           | 0 |           |  |
| Reset  | 0x00000000   |    |    |    |           |    |           |    |          |    |    |    |           |    |           |    |          |    |    |    |              |    |             |   |           |   |          |   |   |   |             |   |           |  |
| Name   | Reserved     |    |    |    | USBHS_NFE |    | USBHS_NFS |    | Reserved |    |    |    | USBFS_NFE |    | USBFS_NFS |    | Reserved |    |    |    | USBHS_FSPHYE |    | USBHS_SOFEN |   | USBHS_DFB |   | Reserved |   |   |   | USBFS_SOFEN |   | USBFS_DFB |  |
| Access |              |    |    |    | RW        |    | RW        |    |          |    |    |    | RW        |    | RW        |    |          |    |    |    | RW           |    | RW          |   | RW        |   |          |   |   |   | RW          |   | RW        |  |

| Bit   | Name      | Function                        | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | Access |
|-------|-----------|---------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:27 | Reserved  | Reserved                        | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | -      |
| 26    | USBHS_NFE | USBHS filter enable register    | Used to control the switch of the USBHS on-chip full-speed PHY DP/DM analog filter in STOP mode <ul style="list-style-type: none"> <li>0b0: Analog filter off</li> <li>0b1: Analog filter on, filtering range please refer to the setting of USBHS_NFS</li> </ul>                                                                                                                                                                                                                   | RW     |
| 25:24 | USBHS_NFS | USBHS filter selection register | Used to control the filtering range of the USBHS on-chip full-speed PHY DP/DM analog filter in STOP mode <ul style="list-style-type: none"> <li>0b00: Filter width gear 1</li> <li>0b01: Filter width gear 2</li> <li>0b10: Filter width gear 3</li> <li>0b11: Filter width gear 4</li> </ul> For specific values of each gear, please refer to the "Electrical Characteristics" chapter in the "Datasheet" under "USB On-Chip Full-Speed PHY STOP Mode Filtering Characteristics." | RW     |
| 23:19 | Reserved  | Reserved                        | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | -      |
| 18    | USBFS_NFE | USBFS filter enable register    | Used to control the switch of the USBHS on-chip full-speed PHY DP/DM analog filter in STOP mode <ul style="list-style-type: none"> <li>0b0: Analog filter off</li> <li>0b1: Analog filter on, filtering range please refer to the setting of USBFS_NFS</li> </ul>                                                                                                                                                                                                                   | RW     |

| Bit   | Name         | Function                                                     | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | Access |
|-------|--------------|--------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 17:16 | USBFS_NFS    | USBFS filter selection register                              | Used to control the filtering range of the USBHS on-chip full-speed PHY DP/DM analog filter in STOP mode <ul style="list-style-type: none"> <li>● 0b00: Filter width gear 1</li> <li>● 0b01: Filter width gear 2</li> <li>● 0b10: Filter width gear 3</li> <li>● 0b11: Filter width gear 4</li> </ul> For specific values of each gear, please refer to the "Electrical Characteristics" chapter in the "Datasheet" under "USB On-Chip Full-Speed PHY STOP Mode Filtering Characteristics." | RW     |
| 15:11 | Reserved     | Reserved                                                     | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | -      |
| 10    | USBHS_FSPHYE | USBHS on-chip full-speed PHY enable bit                      | <ul style="list-style-type: none"> <li>● 0b0: USBHS on-chip full-speed PHY suspended</li> <li>● 0b1: USBHS on-chip full-speed PHY enabled</li> </ul>                                                                                                                                                                                                                                                                                                                                        | RW     |
| 9     | USBHS_SOFEN  | USBHS SOF pulse output enable bit                            | When the USBHS host transmits an SOF or the device successfully receives an SOF, the SOF pulse with a width of 16 HCLK clock periods is output from the PAD enable <ul style="list-style-type: none"> <li>● 0b0: SOF pulse not output</li> <li>● 0b1: SOF pulse output</li> </ul>                                                                                                                                                                                                           | RW     |
| 8     | USBHS_DFB    | USBHS VBUS/ID pin internal debounce filter bypass enable bit | <ul style="list-style-type: none"> <li>● 0b0: Module internal debounce filter active</li> <li>● 0b1: Bypass module internal debounce filter</li> </ul> <div>  <b>Note</b><br/>Active in both device and host modes. </div>                                                                                                                                                                               | RW     |
| 7:2   | Reserved     | Reserved                                                     | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | -      |
| 1     | USBFS_SOFEN  | USBFS SOF pulse output enable bit                            | When the USBFS host transmits an SOF or the device successfully receives an SOF, the SOF pulse with a width of 16 HCLK clock periods is output from the PAD enable <ul style="list-style-type: none"> <li>● 0b0: SOF pulse not output</li> <li>● 0b1: SOF pulse output</li> </ul> <div>  <b>Note</b><br/>Active in both device and host modes. </div>                                                    | RW     |
| 0     | USBFS_DFB    | USBFS VBUS/ID pin internal debounce filter bypass enable bit | <ul style="list-style-type: none"> <li>● 0b0: Module internal debounce filter active</li> <li>● 0b1: Bypass module internal debounce filter</li> </ul> <div>  <b>Note</b><br/>Active in both device and host modes. </div>                                                                                                                                                                               | RW     |

### 35.6.3 USBFS Global Register

These registers are available in both host mode and device mode, and do not require reprogramming when switching between these two modes.

#### 35.6.3.1 VBUS Control Register (USBFS\_GVBUSCFG)

This register can be used to set the VBUS value to ignore the state of the VBUS pin.

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |         |   |          |   |          |   |   |   |  |  |  |  |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---------|---|----------|---|----------|---|---|---|--|--|--|--|
| 0x00   | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7       | 6 | 5        | 4 | 3        | 2 | 1 | 0 |  |  |  |  |
| Reset  | 0x00XX0000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |         |   |          |   |          |   |   |   |  |  |  |  |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   | VBUSVAL |   | VBUSOVEN |   | Reserved |   |   |   |  |  |  |  |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   | RW      |   | RW       |   |          |   |   |   |  |  |  |  |

| Bit  | Name     | Function              | Description                                                                                                                                                                                                                                                                    | Access |
|------|----------|-----------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:8 | Reserved | Reserved              | -                                                                                                                                                                                                                                                                              | -      |
| 7    | VBUSVAL  | VBUS value            | Used to set the VBUS value of USBFS; when set to 1 and VBUSOVEN is set to 1, the USBFS is powered on.<br><br> <b>Note</b><br>Active only in device mode.                                    | RW     |
| 6    | VBUSOVEN | VBUS override enabled | Used to reflect the value set by VBUSVAL to the status of the USBFS CORE. The value of VBUSVAL is enable only when this bit is set to 1.<br><br> <b>Note</b><br>Active only in device mode. | RW     |
| 5:0  | Reserved | Reserved              | -                                                                                                                                                                                                                                                                              | -      |

### 35.6.3.2 AHB Control Register (USBFS\_GAHBCFG)

This register can be used to configure the module after power-on or when changing role modes, mainly containing AHB system-related configuration parameters.

The application must program this register before starting any AHB or USB transactions. Do not change this register after initial programming.

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |          |    |          |       |    |         |   |         |  |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|----------|----|----------|-------|----|---------|---|---------|--|
| 0x08   | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7        | 6  | 5        | 4     | 3  | 2       | 1 | 0       |  |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |          |    |          |       |    |         |   |         |  |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   | PTXFELVL |    | Reserved | DMAEN |    | HBSTLEN |   | GINTMSK |  |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   | RW       | RW |          | RW    | RW |         |   | RW      |  |

| Bit  | Name     | Function                    | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | Access |
|------|----------|-----------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:9 | Reserved | Reserved                    | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | -      |
| 8    | PTXFELVL | Periodic TxFIFO empty level | <p>Indicates when to trigger the periodic TxFIFO empty interrupt bit (PTXFE bit in USBFS_GINTSTS) in the module interrupt register.</p> <ul style="list-style-type: none"> <li>0b0: PTXFE (in USBFS_GINTSTS) interrupt indicates the periodic TxFIFO is in half empty status</li> <li>0b1: PTXFE (in USBFS_GINTSTS) interrupt indicates the periodic TxFIFO is completely empty</li> </ul> <div>  <b>Note</b><br/>           Accessible only in host mode.         </div> | RW     |
| 7    | TXFELVL  | Device TxFIFO empty level   | <p>In device mode, this bit indicates when to trigger the IN endpoint transmit FIFO empty interrupt (TXFE in USBFS_DIEPINTx).</p> <ul style="list-style-type: none"> <li>0b0: TXFE (in USBFS_DIEPINTx) interrupt indicates the IN endpoint TxFIFO is in half empty status</li> <li>0b1: TXFE (in USBFS_DIEPINTx) interrupt indicates the IN endpoint TxFIFO is completely empty</li> </ul> <div>  <b>Note</b><br/>           Active only in device mode.         </div>   | RW     |
| 6    | Reserved | Reserved                    | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | -      |
| 5    | DMAEN    | DMA enable                  | <ul style="list-style-type: none"> <li>0b0: The module operates in slave mode</li> <li>0b1: The module operates in DMA mode</li> </ul> <div>  <b>Note</b><br/>           Active in both device and host modes.         </div>                                                                                                                                                                                                                                             | RW     |

| Bit | Name    | Function                 | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                             | Access |
|-----|---------|--------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 4:1 | HBSTLEN | Burst length/<br>type    | <ul style="list-style-type: none"> <li>● 0b0000: Single</li> <li>● 0b0001: INCR</li> <li>● 0b0011: INCR4</li> <li>● 0b0101: INCR8</li> <li>● 0b0111: INCR16</li> <li>● Others: Setting prohibited</li> </ul> <div>  <b>Note</b><br/>Active in both device and host modes. </div>                                                                                                       | RW     |
| 0   | GINTMSK | Global interrupt<br>mask | <p>This bit is used to mask or unmask global interrupts. The interrupt status register is updated by the module, regardless of the setting of this bit.</p> <ul style="list-style-type: none"> <li>● 0b0: Mask application-triggered interrupts</li> <li>● 0b1: Unmask application-triggered interrupts</li> </ul> <div>  <b>Note</b><br/>Active in both device and host modes. </div> | RW     |

### 35.6.3.3 USB Configuration Register (USBFS\_GUSBCFG)

This register can be used to configure the module after power on or when changing role modes. It contains configuration parameters related to USB and USB-PHY.

The application must program this register before starting any AHB or USB transactions. Do not change this register after initial programming.

| Offset | Bit Position |       |       |          |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |      |    |          |   |        |          |   |       |   |   |   |   |   |
|--------|--------------|-------|-------|----------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|------|----|----------|---|--------|----------|---|-------|---|---|---|---|---|
| 0x0C   | 31           | 30    | 29    | 28       | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12   | 11 | 10       | 9 | 8      | 7        | 6 | 5     | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00001440   |       |       |          |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |      |    |          |   |        |          |   |       |   |   |   |   |   |
| Name   | Reserved     | FDMOD | FHMOD | Reserved |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | TRDT |    | Reserved |   | PHYSEL | Reserved |   | TOTAL |   |   |   |   |   |
| Access |              | RW    | RW    |          |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RW   |    |          |   | RW     |          |   | RW    |   |   |   |   |   |

| Bit   | Name     | Function          | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | Access |
|-------|----------|-------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31    | Reserved | Reserved          | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | -      |
| 30    | FDMOD    | Force device mode | <p>Writing 1 to this bit forces the module into device mode, ignoring the state of the USBFS_ID input pin.</p> <ul style="list-style-type: none"> <li>0b0: Normal mode, depends on the input status of the USBFS_ID pin.</li> <li>0b1: Forced device mode</li> </ul> <p>After forcing the bit to 1, the application must wait at least 25ms before the change takes effect.</p> <div>  <b>Note</b><br/>           Active in both device and host modes.         </div> | RW     |
| 29    | FHMOD    | Force host mode   | <p>Writing 1 to this bit forces the module into host mode, ignoring the state of the USBFS_ID Input Pin.</p> <ul style="list-style-type: none"> <li>0b0: Normal mode, depends on the input status of the USBFS_ID pin.</li> <li>0b1: Force host mode</li> </ul> <p>After forcing the bit to 1, the application must wait at least 25ms before the change takes effect.</p> <div>  <b>Note</b><br/>           Active in both device and host modes.         </div>      | RW     |
| 28:14 | Reserved | Reserved          | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | -      |



| Bit   | Name            | Function                             | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | Access |
|-------|-----------------|--------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 13:10 | TRDT            | USB turn-around time                 | <p>Set the turnaround time in units of PHY clock periods. To calculate the TRDT value, use the following formula:<br/> <math>TRDT = 4 \times AHB \text{ Clock} + 1 \text{ PHY Clock}</math><br/>           For example:</p> <ul style="list-style-type: none"> <li>● If the AHB Clock Frequency=84MHz (PHY Clock Frequency=48MHz), set TRDT to 9.</li> <li>● If the AHB Clock Frequency=48MHz (PHY Clock Frequency=48MHz), set TRDT to 5.</li> </ul> <div>  <b>Note</b><br/>           Active only in device mode.         </div>                                                                                      | RW     |
| 9:7   | <i>Reserved</i> | <i>Reserved</i>                      | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | -      |
| 6     | PHYSEL          | Full speed serial transceiver select | Read always returns 1, indicating the selection of On Chip Full Speed serial transceiver.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | RO     |
| 5:3   | <i>Reserved</i> | <i>Reserved</i>                      | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | -      |
| 2:0   | TOCAL           | FS timeout calibration               | <p>The additional latency introduced by the PHY includes the number of PHY clocks set by the application in this field, as well as the module's full-speed inter-packet timeout interval. The impact of delays introduced by different PHYs on the data line state varies. The standard USB timeout value for full-speed operation ranges from 16 to 18 bit times (inclusive). The application must program this field according to the enumeration speed. The number of bit times added per PHY clock is 0.25 bit times.</p> <div>  <b>Note</b><br/>           Active in both device and host modes.         </div> | RW     |

### 35.6.3.4 Reset Register (USBFS\_GRSTCTL)

The application resets various hardware features in the module through this register.

| Offset | Bit Position |        |          |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |        |   |   |         |   |         |   |          |       |   |       |  |       |
|--------|--------------|--------|----------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|--------|---|---|---------|---|---------|---|----------|-------|---|-------|--|-------|
| 0x10   | 31           | 30     | 29       | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10     | 9 | 8 | 7       | 6 | 5       | 4 | 3        | 2     | 1 | 0     |  |       |
| Reset  | 0x80000000   |        |          |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |        |   |   |         |   |         |   |          |       |   |       |  |       |
| Name   | AHBIDL       | DMAREQ | Reserved |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | TXFNUM |   |   | TXFFLSH |   | RXFFLSH |   | Reserved | FCRST |   | HSRST |  | CSRST |
| Access | RO           | RO     |          |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RW     |   |   | RW      |   | RW      |   |          | RW    |   | RW    |  | RW    |

| Bit   | Name            | Function           | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | Access |
|-------|-----------------|--------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31    | AHBIDL          | AHB master idle    | Indicates the AHB master state machine is in idle condition.<br><br><div>  <b>Note</b><br/>           Active in both device and host modes.         </div>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | RO     |
| 30    | DMAREQ          | DMA request signal | This bit indicates a DMA request is in progress. For debugging purposes.<br><br><div>  <b>Note</b><br/>           Active in both device and host modes.         </div>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | RO     |
| 29:11 | <i>Reserved</i> | <i>Reserved</i>    | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | -      |
| 10:6  | TXFNUM          | TxFIFO number      | FIFO number for FIFO refresh using the TxFIFO refresh bit. This field can only be modified after the module clears the TxFIFO refresh bit.<br><ul style="list-style-type: none"> <li>0b00000: Refresh non-periodic TxFIFO in host mode, refresh TxFIFO0 in device mode</li> <li>0b00001: Refresh periodic TxFIFO in host mode, refresh TXFIFO1 in device mode</li> <li>0b00010: Refresh TXFIFO2 in device mode</li> <li>0b00011: Refresh TXFIFO3 in device mode</li> <li>.....</li> <li>0b00101: Refresh TXFIFO15 in device mode</li> <li>0b10000: Refresh all transmit FIFOs in device mode or host mode</li> <li>Others: Setting prohibited</li> </ul> <div>  <b>Note</b><br/>           Active in both device and host modes.         </div> | RW     |

| Bit | Name            | Function                 | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | Access |
|-----|-----------------|--------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 5   | TXFFLSH         | TxFIFO flush             | <p>This bit selectively refreshes one or all transmit FIFOs, but the operation cannot be performed when the module is handling communication transactions.</p> <p>The application can only perform a write operation to this bit after confirming that the module is not currently executing read and write operations on the TxFIFO. Use the following register for confirmation:</p> <ul style="list-style-type: none"> <li>● Read: NAK active interrupt ensures the module is not currently performing a read operation on the FIFO.</li> <li>● Write: The AHBIDL bit in USBFS_GRSTCTL ensures the module is not currently performing any write operations to the FIFO.</li> </ul> <div>  <b>Note</b><br/> Active in both device and host modes. </div> | RW     |
| 4   | RXFFLSH         | RxFIFO flush             | <p>Applications can use this bit to refresh the entire RxFIFO, but must first ensure the module is not currently processing communication transactions. The application can only perform a write operation to this bit after confirming that the module is not currently executing read and write operations on the RxFIFO.</p> <p>The application must wait until this bit is cleared before performing other operations. Typically, an 8-clock period wait is required (based on the slowest of the PHY or AHB clock).</p> <div>  <b>Note</b><br/> Active in both device and host modes. </div>                                                                                                                                                        | RW     |
| 3   | <i>Reserved</i> | <i>Reserved</i>          | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | -      |
| 2   | FCRST           | Host frame counter reset | <p>When the application writes to this bit, the frame counter in the module is reset. After the frame counter is reset, the frame number of the next SOF sent by the module will be 0.</p> <div>  <b>Note</b><br/> Active in both device and host modes. </div>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | RW     |
| 1   | HSRST           | HCLK soft reset          | <p>The application uses this bit to refresh the control logic in the AHB clock domain. Reset only the AHB clock domain pipeline.</p> <p>FIFO is not refreshed via this bit.</p> <p>After terminating transactions on the AHB in accordance with the protocol, all state machines in the AHB clock domain are reset to the idle state. Clear the CSR control bits used by the AHB clock domain state machine.</p> <p>To clear this interrupt, the status mask bit generated by the AHB clock domain state machine</p>                                                                                                                                                                                                                                                                                                                        | RW     |

| Bit | Name  | Function        | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | Access |
|-----|-------|-----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
|     |       |                 | <p>and used to control the interrupt status must be cleared.</p> <p>Since the interrupt status bit is not cleared, the application can obtain the status of all module events that occurred after this bit was set to 1. This bit is self-clearing; the module will clear this bit after all necessary logic is reset. This process requires several clock periods, depending on the current state of the module.</p> <div>  <b>Note</b><br/>           Active in both device and host modes.         </div>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |        |
| 0   | CSRST | Core soft reset | <p>Reset the HCLK and PCLK domains as follows:<br/>Clear individual interrupt and all CSR register bits except:</p> <ul style="list-style-type: none"> <li>● GATEHCLK bit in USBFS_GCCTL</li> <li>● STPPCLK bit in USBFS_GCCTL</li> <li>● FSLSPCS bit in USBFS_HCFG</li> <li>● DSPD bit in USBFS_DCFG</li> </ul> <p>Reset all module state machines (excluding AHB slaves) to idle state and flush all transmit FIFOs and receive FIFOs.</p> <p>At the end of the last data phase in an AHB transmission, terminate all transactions on the AHB master as soon as possible. Terminate all USB transactions immediately.</p> <p>The application can perform a write operation to this bit at any time when a reset of the module is required. This bit is self-clearing. The block will clear this bit after all necessary logic is reset, which requires several clock periods depending on the current state of the module. Once this bit is cleared, software must wait for at least 3 PHY clock periods before accessing the PHY domain (synchronization delay). Additionally, the software must wait until bit 31 in this register is set to 1 (indicating the AHB master is idle) before it can start operation.</p> <p>Software reset is typically used in two scenarios: first, during software development, and second, after the user dynamically modifies the PHY selection bit in the aforementioned USB configuration register. When the user modifies the PHY, the corresponding clock will be selected for the PHY and used in the PHY domain. Once a new clock is selected, the PHY domain must be reset to ensure normal operation.</p> <div>  <b>Note</b><br/>           Active in both device and host modes.         </div> | RW     |

### 35.6.3.5 Global Interrupt Status Register (USBFS\_GINTSTS)

This register is used to interrupt the application via system-level events in the current mode (Device Mode or Host Mode). This register indicates the current mode.

The FIFO status interrupt is read-only; if software performs read and write operations on the FIFO while processing these interrupts, the FIFO interrupt flag will be automatically cleared.

Before enabling the interrupt bit, the application must clear the USBFS\_GINTSTS register during initialization to avoid generating any interrupts prior to initialization.

| Offset | Bit Position |          |         |         |        |       |       |         |          |           |               |         |      |      |          |      |         |         |        |         |       |          |          |          |        |       |     |          |      |      |    |   |
|--------|--------------|----------|---------|---------|--------|-------|-------|---------|----------|-----------|---------------|---------|------|------|----------|------|---------|---------|--------|---------|-------|----------|----------|----------|--------|-------|-----|----------|------|------|----|---|
| 0x14   | 31           | 30       | 29      | 28      | 27     | 26    | 25    | 24      | 23       | 22        | 21            | 20      | 19   | 18   | 17       | 16   | 15      | 14      | 13     | 12      | 11    | 10       | 9        | 8        | 7      | 6     | 5   | 4        | 3    | 2    | 1  | 0 |
| Reset  | 0x14000020   |          |         |         |        |       |       |         |          |           |               |         |      |      |          |      |         |         |        |         |       |          |          |          |        |       |     |          |      |      |    |   |
| Name   | WKUINT       | VBUSVINT | DISCINT | CIDSCHG | LPMINT | PTXFE | HCINT | HPRTINT | Reserved | DATAFSUSP | IPXFR/INCOMPI | ISOIXFR | OEPI | IEPI | Reserved | EOPF | ISOODRP | ENUMDNE | USBRST | USBSUSP | ESUSP | Reserved | GONAKEFF | GINAKEFF | NPTXFE | RXFNE | SOF | Reserved | MMIS | CMOD |    |   |
| Access | RW           | RW       | RW      | RW      | RW     | RO    | RO    | RO      |          | RW        | RW            | RW      | RO   | RO   |          | RW   | RW      | RW      | RW     | RW      | RW    |          |          | RO       | RO     | RO    | RO  | RW       |      | RW   | RO |   |

| Bit | Name     | Function                                            | Description                                                                                                                                                                                                                                                                                                                                                                                                                   | Access |
|-----|----------|-----------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31  | WKUINT   | Resume/<br>remote wakeup<br>detected inter-<br>rupt | In device mode, this interrupt is triggered when a resume signal is detected on the USB bus.<br>In host mode, this interrupt will be triggered when a remote wakeup is detected on the USB.<br>Writing 1 to this bit clears it via software.<br><br><div>  <b>Note</b><br/>           Active in both device and host modes.         </div> | RW     |
| 30  | VBUSVINT | VBUS valid in-<br>terrupt                           | In device mode, this interrupt is triggered when a low-to-high transition is detected on the USBFS_VBUS pin.<br>Writing 1 to this bit clears it via software.<br><br><div>  <b>Note</b><br/>           Active only in device mode.         </div>                                                                                          | RW     |
| 29  | DISCINT  | Disconnect de-<br>tected interrupt                  | Trigger this interrupt when a device disconnection is detected.<br>Writing 1 to this bit clears it via software.<br><br><div>  <b>Note</b><br/>           Active only in host mode.         </div>                                                                                                                                         | RW     |

| Bit | Name     | Function                        | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | Access |
|-----|----------|---------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 28  | CIDSCHG  | Connector ID status change      | <p>When the connector ID line status changes, the module sets this bit to 1.<br/>Writing 1 to this bit clears it via software.</p> <div>  <b>Note</b><br/>Active in both device and host modes. </div>                                                                                                                                                                                                                                                                                                                                | RW     |
| 27  | LPMINT   | LPM interrupt                   | <ul style="list-style-type: none"> <li>● Device mode: This bit is set to 1 when the device receives a correct LPM transmission and active response.</li> <li>● Host mode: This bit is set to 1 when the host receives a active response after sending LPM or when the preset retry count is exceeded.</li> </ul>                                                                                                                                                                                                                                                                                                       | RW     |
| 26  | PTXFE    | Periodic TxFIFO empty interrupt | <p>When the periodic transmit FIFO is half-empty or in full empty status, and there is space in the periodic request queue to write at least one entry, this interrupt will be triggered. Whether the FIFO is in a half empty state or full empty status is determined by the periodic TxFIFO empty level bit (PTXFELVL bit in USBFS_GAHBCFG register) in the USBFS_GAHBCFG register.</p> <div>  <b>Note</b><br/>Active only in host mode. </div>                                                                                    | RO     |
| 25  | HCINT    | Host channels interrupt         | <p>When the module sets this bit to 1, it indicates a pending interrupt in one channel of the block (in Host Mode). The application must read the host USBFS_HAINT register to determine the exact channel number where the interrupt occurred, then read the corresponding USBFS_HCINTx register to identify the specific cause of the interrupt. The application must first clear the corresponding status bit in the USBFS_HCINTx register before clearing that bit.</p> <div>  <b>Note</b><br/>Active only in host mode. </div> | RO     |
| 24  | HPRTINT  | Host port interrupt             | <p>When the module sets this bit to 1, it indicates a status change in the USBFS controller port under host mode. The application must read the USBFS_HPRT register to determine the exact event that triggered this interrupt. The application must first clear the corresponding status bit in the USBFS_HPRT register before clearing this bit.</p> <div>  <b>Note</b><br/>Active only in host mode. </div>                                                                                                                      | RO     |
| 23  | Reserved | Reserved                        | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | -      |

| Bit | Name      | Function             | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | Access |
|-----|-----------|----------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 22  | DATAFSUSP | Data fetch suspended | <p>This interrupt is only active in DMA mode. This interrupt indicates that the module has stopped fetching data for the IN endpoint due to unavailable TxFIFO space or request queue space. The application uses this interrupt in the endpoint mismatch algorithm. For example, after detecting an endpoint mismatch, the application will perform the following operations:</p> <ul style="list-style-type: none"> <li>● Set the global non-periodic IN NAK handshake signal to 1</li> <li>● Disable IN endpoint</li> <li>● Clear FIFO</li> <li>● Determine the token sequence based on the IN token sequence learning queue</li> <li>● Re-enable endpoint</li> <li>● Clear global non-periodic IN NAK handshake signal</li> </ul> <p>If the global non-periodic IN NAK has been cleared but the module has not yet retrieved data from the IN endpoint, and an IN token is received simultaneously, the module will generate an "IN token received while FIFO is empty" interrupt. Then USBFS sends a NAK response to the host. To avoid the above situation, the application can check the DATAFSUSP interrupt in USBFS_GINTSTS, which ensures the global NAK handshake signal is cleared only after the FIFO is full. Additionally, the application can mask the "Interrupt upon receiving IN token when FIFO is empty" while clearing the global IN NAK handshake signal.</p> <p>Writing 1 to this bit clears it via software.</p> <div>  <b>Note</b><br/>           Active only in device mode.         </div> | RW     |

| Bit | Name                        | Function                                                                                  | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | Access |
|-----|-----------------------------|-------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 21  | IPXFR/<br>INCOMPISO-<br>OUT | IPXFR: Incomplete periodic transfer<br>INCOMPISO-OUT: Incomplete isochronous OUT transfer | <ul style="list-style-type: none"> <li>● IPXFR:<br/>In host mode, if there are pending periodic transactions that are scheduled to complete during the current frame, the module will set this interrupt bit to 1.<br/>Writing 1 to this bit clears it via software.</li> </ul> <div>  <b>Note</b><br/>Active only in host mode.         </div> <ul style="list-style-type: none"> <li>● INCOMPISOOUT:<br/>In device mode, when the module sets this interrupt to 1, it indicates that at least one transfer on a synchronous OUT endpoint in the current frame remains incomplete. This interrupt is triggered along with the periodic frame end interrupt (EOPF) bit in this register.<br/>Writing 1 to this bit clears it via software.</li> </ul> <div>  <b>Note</b><br/>Active only in device mode.         </div> | RW     |
| 20  | IISOIXFR                    | Incomplete isochronous IN transfer                                                        | <p>When the module sets this interrupt to 1, it indicates that at least one transfer on a synchronous IN endpoint in the current frame remains incomplete. This interrupt is triggered along with the periodic frame end interrupt (EOPF) bit in this register.<br/>Writing 1 to this bit clears it via software.</p> <div>  <b>Note</b><br/>Active only in device mode.         </div>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | RW     |
| 19  | OEPINT                      | OUT endpoint interrupt                                                                    | <p>When the module sets this bit to 1, it indicates a pending interrupt on an OUT endpoint in the module (in device mode). The application must read the host USBFS_DAIN register to determine the exact OUT endpoint number where the interrupt occurred, then read the corresponding USBFS_DOEPINTx register to identify the specific cause of the interrupt. The application must first clear the corresponding status bit in the USBFS_DOEPINTx register before clearing this bit.</p> <div>  <b>Note</b><br/>Active only in device mode.         </div>                                                                                                                                                                                                                                                                                                                                           | RO     |



| Bit   | Name     | Function                                 | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | Access |
|-------|----------|------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 18    | IEPINT   | IN endpoint interrupt                    | <p>When the module sets this bit to 1, it indicates a pending interrupt on an IN endpoint within the module (in device mode). The application must read the host USBFS_DAIN register to determine the exact IN endpoint number where the interrupt occurred, then read the corresponding USBFS_DIEPINTx register to identify the specific cause of the interrupt. The application must first clear the corresponding status bit in the USBFS_DIEPINTx register before clearing this bit.</p> <div>  <b>Note</b><br/>           Active only in device mode.         </div> | RO     |
| 17:16 | Reserved | Reserved                                 | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | -      |
| 15    | EOPF     | End of periodic frame interrupt          | <p>Indicates that the current frame has reached the period specified by the periodic frame interval field (PFIVL bit in USBFS_DCFG) in the USBFS_DCFG register.</p> <p>Writing 1 to this bit clears it via software.</p> <div>  <b>Note</b><br/>           Active only in device mode.         </div>                                                                                                                                                                                                                                                                     | RW     |
| 14    | ISOODRP  | Isochronous OUT packet dropped interrupt | <p>If the module cannot write the Isochronous OUT data packet to the Rx FIFO due to insufficient space to accommodate the maximum data packet of the Isochronous OUT endpoint, the module will set this bit to 1.</p> <p>Writing 1 to this bit clears it via software.</p> <div>  <b>Note</b><br/>           Active only in device mode.         </div>                                                                                                                                                                                                                 | RW     |
| 13    | ENUMDNE  | Enumeration done interrupt               | <p>When the module sets this bit to 1, it indicates the speed enumeration is complete. The application must read the USBFS_DSTS register to obtain the enumeration speed.</p> <p>Writing 1 to this bit clears it via software.</p> <div>  <b>Note</b><br/>           Active only in device mode.         </div>                                                                                                                                                                                                                                                         | RW     |
| 12    | USB_RST  | USB reset interrupt                      | <p>When the module sets this bit to 1, it indicates a reset signal is detected on the USB.</p> <p>Writing 1 to this bit clears it via software.</p> <div>  <b>Note</b><br/>           Active only in device mode.         </div>                                                                                                                                                                                                                                                                                                                                        | RW     |

| Bit | Name            | Function                                       | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | Access |
|-----|-----------------|------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 11  | USBSUSP         | USB suspend interrupt                          | <p>When the module sets this bit to 1, it indicates that a suspend status has been detected on the USB. When the idle state on the USB bus persists for 3ms, the module enters a suspend state. Writing 1 to this bit clears it via software.</p> <div>  <b>Note</b><br/>           Active only in device mode.         </div>                                                                                                                                                                                                                                                                                                                                                                    | RW     |
| 10  | ESUSP           | Early suspend interrupt                        | <p>When the module sets this bit, it indicates that the USB idle state has been detected for 3ms.</p> <div>  <b>Note</b><br/>           Active only in device mode.         </div>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | RW     |
| 9:8 | <i>Reserved</i> | <i>Reserved</i>                                | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | -      |
| 7   | GONAKEFF        | Global OUT NAK effective interrupt             | <p>Indicates that the "Set global OUT NAK to 1" bit (SGONAK bit in USBFS_DCTL register) set by the application has taken effect in the module. By writing to the "Clear Global OUT NAK" bit (CGONAK bit in USBFS_DCTL register), this bit can be cleared.</p> <div>  <b>Note</b><br/>           Active only in device mode.         </div>                                                                                                                                                                                                                                                                                                                                                       | RO     |
| 6   | GINAKEFF        | Global IN non-periodic NAK effective interrupt | <p>Indicates that the "Set global non-periodic IN NAK to 1" bit (SGINAK bit in USBFS_DCTL) set by the application in the USBFS_DCTL register has taken effect in the module. In other words, the module has sampled the global IN NAK bit set by the application, and the result has taken effect. By clearing the "Clear global non-periodic IN NAK" bit (CGINAK bit in USBFS_DCTL) in the USBFS_DCTL register, this bit can be cleared. This interrupt does not necessarily indicate that a NAK handshake signal has been sent on the USB. The priority of the STALL bit is higher than that of the NAK bit.</p> <div>  <b>Note</b><br/>           Active only in device mode.         </div> | RO     |

| Bit | Name            | Function                            | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | Access |
|-----|-----------------|-------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 5   | NPTXFE          | Non-periodic TxFIFO empty interrupt | <p>When the non-periodic TxFIFO is half-empty or in full empty status, and there is at least space for one writable entry in the non-periodic transmission request queue, this interrupt will be triggered. Whether the FIFO is in a half empty state or full empty status is determined by the non-periodic TxFIFO empty level bit (TXFELVL bit in USBFS_GAHBCFG register) in the USBFS_GAHBCFG register.</p> <div>  <b>Note</b><br/>           Active only in host mode.         </div>                                                                                                                                                         | RO     |
| 4   | RXFNE           | RxFIFO non-empty interrupt          | <p>Indicates that at least one data packet is waiting to be read in the RxFIFO.</p> <div>  <b>Note</b><br/>           Active in both device and host modes.         </div>                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | RO     |
| 3   | SOF             | Start of frame interrupt            | <p>In host mode, when the module sets this bit to 1, it indicates that an SOF (FS) or Keep-Alive (LS) signal has been transmitted on the USB. The application must set this bit to 1 to clear the interrupt.</p> <p>In device mode, when the module sets this bit to 1, it indicates that an SOF token has been received on the USB. The application can obtain the current frame number by reading the device status register. This interrupt occurs only when the module operates in FS mode.</p> <p>Writing 1 to this bit clears it via software.</p> <div>  <b>Note</b><br/>           Active in both device and host modes.         </div> | RW     |
| 2   | <i>Reserved</i> | <i>Reserved</i>                     | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | -      |
| 1   | MMIS            | Mode mismatch interrupt             | <p>When the application attempts to access the following register, the module sets this bit to 1:</p> <ul style="list-style-type: none"> <li>● The module operates in device mode to access the host mode register.</li> <li>● The module operates in host mode to access the device mode register.</li> </ul> <p>Register access ends with an OKAY response on the AHB, but the access is internally ignored by the module and does not affect module operation.</p> <p>Writing 1 to this bit clears it via software.</p> <div>  <b>Note</b><br/>           Active in both device and host modes.         </div>                               | RW     |

| Bit | Name | Function                  | Description                                                                                                                                                                                                                                                                       | Access |
|-----|------|---------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 0   | CMOD | Current mode of operation | <p>Indicates the current mode.</p> <ul style="list-style-type: none"><li>● 0b0: Device mode</li><li>● 0b1: Host mode</li></ul> <div> <b>Note</b><br/>Active in both device and host modes.</div> | RO     |

### 35.6.3.6 Global Interrupt Mask Register (USBFS\_GINTMSK)

This register is used in conjunction with the global interrupt status register (USBFS\_GINTSTS) to interrupt the application. If an interrupt bit is masked, the corresponding interrupt will not be generated, but the associated register bit will still be set to 1.

| Offset | Bit Position |         |        |          |         |        |      |        |          |            |                       |          |       |       |          |       |          |          |         |          |        |          |           |           |         |        |      |          |       |          |   |   |
|--------|--------------|---------|--------|----------|---------|--------|------|--------|----------|------------|-----------------------|----------|-------|-------|----------|-------|----------|----------|---------|----------|--------|----------|-----------|-----------|---------|--------|------|----------|-------|----------|---|---|
| 0x18   | 31           | 30      | 29     | 28       | 27      | 26     | 25   | 24     | 23       | 22         | 21                    | 20       | 19    | 18    | 17       | 16    | 15       | 14       | 13      | 12       | 11     | 10       | 9         | 8         | 7       | 6      | 5    | 4        | 3     | 2        | 1 | 0 |
| Reset  | 0x00000000   |         |        |          |         |        |      |        |          |            |                       |          |       |       |          |       |          |          |         |          |        |          |           |           |         |        |      |          |       |          |   |   |
| Name   | WKUIM        | VBUSVIM | DISCIM | CIDSCHGM | LPMINTM | PTXFEM | HCIM | HPRTIM | Reserved | DATAFSUSPM | IPXFRM/INCOMPIISOOUTM | IISOXFRM | OEPIM | IEPIM | Reserved | EOPFM | ISOODRPM | ENUMDNEM | USBRSTM | USBSUSPM | ESUSPM | Reserved | GONAKEFFM | GINAKEFFM | NPTXFEM | RXFNEM | SOFM | Reserved | MMISM | Reserved |   |   |
| Access | RW           | RW      | RW     | RW       | RW      | RW     | RW   | RW     |          | RW         | RW                    | RW       | RW    | RW    |          | RW    | RW       | RW       | RW      | RW       | RW     |          | RW        | RW        | RW      | RW     | RW   |          |       | RW       |   |   |

| Bit | Name     | Function                                                 | Description                                                                                                                                                                                                                                                             | Access |
|-----|----------|----------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31  | WKUIM    | Resume/<br>remote wakeup<br>detected inter-<br>rupt mask | <ul style="list-style-type: none"> <li>0b0: Maskable interrupt</li> <li>0b1: Enable interrupt</li> </ul> <div>  <b>Note</b><br/>Active in both device and host modes.         </div> | RW     |
| 30  | VBUSVIM  | VBUS valid in-<br>terrupt mask                           | <ul style="list-style-type: none"> <li>0b0: Maskable interrupt</li> <li>0b1: Enable interrupt</li> </ul> <div>  <b>Note</b><br/>Active only in device mode.         </div>           | RW     |
| 29  | DISCIM   | Disconnect de-<br>tected interrupt<br>mask               | <ul style="list-style-type: none"> <li>0b0: Maskable interrupt</li> <li>0b1: Enable interrupt</li> </ul> <div>  <b>Note</b><br/>Active only in host mode.         </div>             | RW     |
| 28  | CIDSCHGM | Connector ID<br>status change<br>interrupt mask          | <ul style="list-style-type: none"> <li>0b0: Maskable interrupt</li> <li>0b1: Enable interrupt</li> </ul> <div>  <b>Note</b><br/>Active in both device and host modes.         </div> | RW     |
| 27  | LPMINTM  | LPM interrupt<br>mask                                    | <ul style="list-style-type: none"> <li>0b0: Maskable interrupt</li> <li>0b1: Enable interrupt</li> </ul> <div>  <b>Note</b><br/>Active in both device and host modes.         </div> | RW     |

| Bit | Name                      | Function                                                                                                                  | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | Access |
|-----|---------------------------|---------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 26  | PTXFEM                    | Periodic TxFIFO empty interrupt mask                                                                                      | <ul style="list-style-type: none"> <li>● 0b0: Maskable interrupt</li> <li>● 0b1: Enable interrupt</li> </ul> <div>  <b>Note</b><br/>Active only in host mode.         </div>                                                                                                                                                                                                                                                                                                                                                                                                                                  | RW     |
| 25  | HCIM                      | Host channels interrupt mask                                                                                              | <ul style="list-style-type: none"> <li>● 0b0: Maskable interrupt</li> <li>● 0b1: Enable interrupt</li> </ul> <div>  <b>Note</b><br/>Active only in host mode.         </div>                                                                                                                                                                                                                                                                                                                                                                                                                                  | RW     |
| 24  | HPRTIM                    | Host port interrupt mask                                                                                                  | <ul style="list-style-type: none"> <li>● 0b0: Maskable interrupt</li> <li>● 0b1: Enable interrupt</li> </ul> <div>  <b>Note</b><br/>Active only in host mode.         </div>                                                                                                                                                                                                                                                                                                                                                                                                                                  | RW     |
| 23  | <i>Reserved</i>           | <i>Reserved</i>                                                                                                           | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | -      |
| 22  | DATA-FSUSPM               | Data fetch suspended interrupt mask                                                                                       | <ul style="list-style-type: none"> <li>● 0b0: Maskable interrupt</li> <li>● 0b1: Enable interrupt</li> </ul> <div>  <b>Note</b><br/>Active only in device mode.         </div>                                                                                                                                                                                                                                                                                                                                                                                                                              | RW     |
| 21  | IPXFRM/<br>INCOMPI-SOOUTM | IPXFRM: Incomplete periodic transfer interrupt mask<br>INCOMPI-SOOUTM: Incomplete isochronous OUT transfer interrupt mask | <ul style="list-style-type: none"> <li>● IPXFRM:               <ul style="list-style-type: none"> <li>▶ 0b0: Maskable interrupt</li> <li>▶ 0b1: Enable interrupt</li> </ul> </li> </ul> <div>  <b>Note</b><br/>Active only in host mode.         </div> <ul style="list-style-type: none"> <li>● INCOMPISOOUTM:               <ul style="list-style-type: none"> <li>▶ 0b0: Maskable interrupt</li> <li>▶ 0b1: Enable interrupt</li> </ul> </li> </ul> <div>  <b>Note</b><br/>Active only in device mode.         </div> | RW     |
| 20  | IISOIXFRM                 | Incomplete isochronous IN transfer interrupt mask                                                                         | <ul style="list-style-type: none"> <li>● 0b0: Maskable interrupt</li> <li>● 0b1: Enable interrupt</li> </ul> <div>  <b>Note</b><br/>Active only in device mode.         </div>                                                                                                                                                                                                                                                                                                                                                                                                                              | RW     |
| 19  | OEPIM                     | OUT endpoint interrupt mask                                                                                               | <ul style="list-style-type: none"> <li>● 0b0: Maskable interrupt</li> <li>● 0b1: Enable interrupt</li> </ul> <div>  <b>Note</b><br/>Active only in device mode.         </div>                                                                                                                                                                                                                                                                                                                                                                                                                              | RW     |

| Bit   | Name      | Function                                      | Description                                                                                                                                                                                                                                                       | Access |
|-------|-----------|-----------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 18    | IEPIM     | IN endpoint interrupt mask                    | <ul style="list-style-type: none"> <li>● 0b0: Maskable interrupt</li> <li>● 0b1: Enable interrupt</li> </ul> <div>  <b>Note</b><br/>Active only in device mode.         </div>   | RW     |
| 17:16 | Reserved  | Reserved                                      | -                                                                                                                                                                                                                                                                 | -      |
| 15    | EOPFM     | End of periodic frame interrupt mask          | <ul style="list-style-type: none"> <li>● 0b0: Maskable interrupt</li> <li>● 0b1: Enable interrupt</li> </ul> <div>  <b>Note</b><br/>Active only in device mode.         </div>   | RW     |
| 14    | ISOODRPM  | Isochronous OUT packet dropped interrupt mask | <ul style="list-style-type: none"> <li>● 0b0: Maskable interrupt</li> <li>● 0b1: Enable interrupt</li> </ul> <div>  <b>Note</b><br/>Active only in device mode.         </div>   | RW     |
| 13    | ENUMDNEM  | Enumeration done interrupt mask               | <ul style="list-style-type: none"> <li>● 0b0: Maskable interrupt</li> <li>● 0b1: Enable interrupt</li> </ul> <div>  <b>Note</b><br/>Active only in device mode.         </div> | RW     |
| 12    | USBRSTM   | USB reset interrupt mask                      | <ul style="list-style-type: none"> <li>● 0b0: Maskable interrupt</li> <li>● 0b1: Enable interrupt</li> </ul> <div>  <b>Note</b><br/>Active only in device mode.         </div> | RW     |
| 11    | USBSUSPM  | USB suspend interrupt mask                    | <ul style="list-style-type: none"> <li>● 0b0: Maskable interrupt</li> <li>● 0b1: Enable interrupt</li> </ul> <div>  <b>Note</b><br/>Active only in device mode.         </div> | RW     |
| 10    | ESUSPM    | Early suspend interrupt mask                  | <ul style="list-style-type: none"> <li>● 0b0: Maskable interrupt</li> <li>● 0b1: Enable interrupt</li> </ul> <div>  <b>Note</b><br/>Active only in device mode.         </div> | RW     |
| 9:8   | Reserved  | Reserved                                      | -                                                                                                                                                                                                                                                                 | -      |
| 7     | GONAKEFFM | Global OUT NAK effective interrupt mask       | <ul style="list-style-type: none"> <li>● 0b0: Maskable interrupt</li> <li>● 0b1: Enable interrupt</li> </ul> <div>  <b>Note</b><br/>Active only in device mode.         </div> | RW     |

| Bit | Name            | Function                                            | Description                                                                                                                                                                                                                                                                 | Access |
|-----|-----------------|-----------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 6   | GINAKEFFM       | Global IN non-periodic NAK effective interrupt mask | <ul style="list-style-type: none"> <li>● 0b0: Maskable interrupt</li> <li>● 0b1: Enable interrupt</li> </ul> <div>  <b>Note</b><br/>Active only in device mode.         </div>             | RW     |
| 5   | NPTXFEM         | Non-periodic Tx FIFO empty interrupt mask           | <ul style="list-style-type: none"> <li>● 0b0: Maskable interrupt</li> <li>● 0b1: Enable interrupt</li> </ul> <div>  <b>Note</b><br/>Active only in host mode.         </div>               | RW     |
| 4   | RXFNEM          | Rx FIFO non-empty interrupt mask                    | <ul style="list-style-type: none"> <li>● 0b0: Maskable interrupt</li> <li>● 0b1: Enable interrupt</li> </ul> <div>  <b>Note</b><br/>Active in both device and host modes.         </div>   | RW     |
| 3   | SOFM            | Start of frame interrupt mask                       | <ul style="list-style-type: none"> <li>● 0b0: Maskable interrupt</li> <li>● 0b1: Enable interrupt</li> </ul> <div>  <b>Note</b><br/>Active in both device and host modes.         </div>  | RW     |
| 2   | <i>Reserved</i> | <i>Reserved</i>                                     | -                                                                                                                                                                                                                                                                           | -      |
| 1   | MMISM           | Mode mismatch interrupt mask                        | <ul style="list-style-type: none"> <li>● 0b0: Maskable interrupt</li> <li>● 0b1: Enable interrupt</li> </ul> <div>  <b>Note</b><br/>Active in both device and host modes.         </div> | RW     |
| 0   | <i>Reserved</i> | <i>Reserved</i>                                     | -                                                                                                                                                                                                                                                                           | -      |



### 35.6.3.7 Receive Status Debug Read/Status Read and Pop Register (USBFS\_GRXSTSR/USBFS\_GRXSTSP)

Reading the receive status debug read register returns the content at the top of the receive FIFO.

Reading the receive status read and pop register can additionally pop the data entry at the top of the RxFIFO. The interpretation of the receive status content differs between host mode and device mode.

When the receive FIFO is empty, the module ignores read or pop operations on this register and return the value 0x0000 0000. When the receive FIFO non-empty bit (RXFNE bit in USBFS\_GINTSTS) of the module interrupt register is set, the application must only pop the receive status FIFO.

The read offset: 0x1C

The pop offset: 0x20

#### Host Mode

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |        |    |      |    |      |    |    |    |    |    |   |   |   |   |             |   |   |   |   |   |  |  |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|--------|----|------|----|------|----|----|----|----|----|---|---|---|---|-------------|---|---|---|---|---|--|--|
| '      | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19     | 18 | 17   | 16 | 15   | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5           | 4 | 3 | 2 | 1 | 0 |  |  |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |        |    |      |    |      |    |    |    |    |    |   |   |   |   |             |   |   |   |   |   |  |  |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    | PKTSTS |    | DPID |    | BCNT |    |    |    |    |    |   |   |   |   | CHNUM_EPNUM |   |   |   |   |   |  |  |
| Access |              |    |    |    |    |    |    |    |    |    |    |    | RO     |    | RO   |    | RO   |    |    |    |    |    |   |   |   |   | RO          |   |   |   |   |   |  |  |

| Bit   | Name        | Function       | Description                                                                                                                                                                                                                                                                                                                                 | Access |
|-------|-------------|----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:21 | Reserved    | Reserved       | -                                                                                                                                                                                                                                                                                                                                           | -      |
| 20:17 | PKTSTS      | Packet status  | Indicates the status of the received data packet. <ul style="list-style-type: none"> <li>0b0010: IN packet received</li> <li>0b0011: IN transfer complete (triggers interrupt)</li> <li>0b0101: Data synchronization error (triggers interrupt)</li> <li>0b0111: Channel suspended (triggers interrupt)</li> <li>Other: Reserved</li> </ul> | RO     |
| 16:15 | DPID        | Data PID       | Indicates the data PID of the received packet <ul style="list-style-type: none"> <li>0b00: DATA0</li> <li>0b01: DATA2</li> <li>0b10: DATA1</li> <li>0b11: MDATA</li> </ul>                                                                                                                                                                  | RO     |
| 14:4  | BCNT        | Byte count     | Indicates the number of bytes in the received IN data packet.                                                                                                                                                                                                                                                                               | RO     |
| 3:0   | CHNUM_EPNUM | Channel number | Indicates the channel number to which the currently received data packet belongs.                                                                                                                                                                                                                                                           | RO     |

## Device Mode:

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |        |    |      |    |      |    |    |    |    |    |   |   |   |   |             |   |   |   |   |   |  |  |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|--------|----|------|----|------|----|----|----|----|----|---|---|---|---|-------------|---|---|---|---|---|--|--|
| '      | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19     | 18 | 17   | 16 | 15   | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5           | 4 | 3 | 2 | 1 | 0 |  |  |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |        |    |      |    |      |    |    |    |    |    |   |   |   |   |             |   |   |   |   |   |  |  |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    | PKTSTS |    | DPID |    | BCNT |    |    |    |    |    |   |   |   |   | CHNUM_EPNUM |   |   |   |   |   |  |  |
| Access |              |    |    |    |    |    |    |    |    |    |    |    | RO     |    | RO   |    | RO   |    |    |    |    |    |   |   |   |   | RO          |   |   |   |   |   |  |  |

| Bit   | Name        | Function       | Description                                                                                                                                                                                                                                                                                                                                                                          | Access |
|-------|-------------|----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:21 | Reserved    | Reserved       | -                                                                                                                                                                                                                                                                                                                                                                                    | -      |
| 20:17 | PKTSTS      | Packet status  | Indicates the status of the received data packet. <ul style="list-style-type: none"> <li>0b0001: Global OUT NAK (triggers interrupt).</li> <li>0b0010: OUT packet received</li> <li>0b0011: OUT transfer complete (triggers interrupt).</li> <li>0b0100: SETUP transaction complete (triggers interrupt).</li> <li>0b0110: SETUP packet received</li> <li>Other: Reserved</li> </ul> | RO     |
| 16:15 | DPID        | Data PID       | Indicates the data PID of the received OUT data packet. <ul style="list-style-type: none"> <li>0b00: DATA0</li> <li>0b01: DATA2</li> <li>0b10: DATA1</li> <li>0b11: MDATA</li> </ul>                                                                                                                                                                                                 | RO     |
| 14:4  | BCNT        | Byte count     | Indicates the number of bytes in the received data packet.                                                                                                                                                                                                                                                                                                                           | RO     |
| 3:0   | CHNUM_EPNUM | Channel number | Indicates the endpoint number to which the currently received data packet belongs.                                                                                                                                                                                                                                                                                                   | RO     |

### 35.6.3.8 Receive FIFO Size Register (USBFS\_GRXFSIZ)

This application can program the RAM size that must be allocated to the RxFIFO.

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |      |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|------|----|---|---|---|---|---|---|---|---|---|---|
| 0x24   | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11   | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000280   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |      |    |   |   |   |   |   |   |   |   |   |   |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RXFD |    |   |   |   |   |   |   |   |   |   |   |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |      |    |   |   |   |   |   |   |   |   |   |   |

| Bit   | Name     | Function     | Description                                                                                                                    | Access |
|-------|----------|--------------|--------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:11 | Reserved | Reserved     | -                                                                                                                              | -      |
| 10:0  | RXFD     | RxFIFO depth | In units of 32-bit words. The value range is from 16 to 256.<br>The power-on reset value is the maximum FIFO depth of Rx data. | RW     |

### 35.6.3.9 Host Non Periodic Transmit FIFO Size Register (USBFS\_HNPTXFSIZ) /Device Endpoint 0 Transmit FIFO Size Register (USBFS\_DIEPTXF0)

This application can program the size of RAM that must be allocated to the TxFIFO.

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |                |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----------------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0x28   | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15             | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x02800280   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |                |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | NPTXFD/TX0FD |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | NPTXFSA/TX0FSA |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access | RW           |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RW             |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit   | Name               | Function                                                                                           | Description                                                                                                                                                                                                                                                                           | Access |
|-------|--------------------|----------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:16 | NPTXFD/<br>TX0FD   | NPTXFD: Non-periodic TxFIFO depth<br>TX0FD: End-point 0 TxFIFO depth                               | <ul style="list-style-type: none"> <li>Host mode: NPTXFD, in units of 32-bit words. The value range is from 16 to 256.</li> <li>Device mode: TX0FD, in units of 32-bit words. The value range is from 16 to 256.</li> </ul>                                                           | RW     |
| 15:0  | NPTXFSA/<br>TX0FSA | NPTXFSA: Non-periodic transmit RAM start address<br>TX0FSA: End-point 0 transmit RAM start address | <ul style="list-style-type: none"> <li>Host mode: NPTXFSA, this field contains the starting address of the memory of the non-periodic send FIFO RAM.</li> <li>Device mode: TX0FSA, this field contains the starting address of the memory of the endpoint 0 send FIFO RAM.</li> </ul> | RW     |

### 35.6.3.10 Non periodic Transmit FIFO/Queue Status Register (USBFS\_HNPTXSTS)

This read-only register contains free space information for the non-periodic TxFIFO and non-periodic transmit request queue.

This register is enable only in host mode; inactive in device mode.

| Offset | Bit Position |          |    |    |    |    |    |    |    |          |    |    |    |    |    |    |    |          |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |
|--------|--------------|----------|----|----|----|----|----|----|----|----------|----|----|----|----|----|----|----|----------|----|----|----|----|---|---|---|---|---|---|---|---|---|---|--|
| 0x2C   | 31           | 30       | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22       | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14       | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |  |
| Reset  | 0x00080280   |          |    |    |    |    |    |    |    |          |    |    |    |    |    |    |    |          |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |
| Name   | Reserved     | NPTXQTOP |    |    |    |    |    |    |    | NPTQXSAV |    |    |    |    |    |    |    | NPTXFSAV |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |
| Access |              | RO       |    |    |    |    |    |    |    | RO       |    |    |    |    |    |    |    | RO       |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |

| Bit   | Name     | Function                                            | Description                                                                                                                                                                                                                                                                                                                                                                              | Access |
|-------|----------|-----------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31    | Reserved | Reserved                                            | -                                                                                                                                                                                                                                                                                                                                                                                        | -      |
| 30:24 | NPTXQTOP | Top of the non-periodic transmit request queue      | The entry currently being processed by the MAC in the non-periodic transmit request queue.<br>[30:27]: Channel/endpoint number<br>[26:25]:<br><ul style="list-style-type: none"> <li>0b00: IN/OUT token</li> <li>0b01: Zero-length transmit data packet</li> <li>0b10: Reserved</li> <li>0b11: Channel stop command</li> </ul> [24]: End (last entry of the selected channel/endpoint)   | RO     |
| 23:16 | NPTQXSAV | Non-periodic transmit request queue space available | Indicates the available free space size in the non-periodic transmit request queue, ranging from 0 to 8.<br>In host mode, this queue holds IN and OUT requests.<br><ul style="list-style-type: none"> <li>0: Non-periodic transmit request queue is full</li> <li>1: 1 positions available</li> <li>2: 2 positions available</li> <li>.....</li> <li>8: 8 positions available</li> </ul> | RO     |
| 15:0  | NPTXFSAV | Non-periodic TxFIFO space available                 | Indicates the available free space size in the non-periodic TxFIFO, in 32-bit word units, ranging from 0 to 256.<br><ul style="list-style-type: none"> <li>0: Non-periodic TxFIFO is full</li> <li>1: 1 word available</li> <li>2: 2 words available</li> <li>.....</li> <li>256: 256 words available</li> </ul>                                                                         | RO     |

35.6.3.11 Module ID Register (USBFS\_CID)

This register is a programmable user configuration ID register.

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0x3C   | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x12345678   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | PRODUCT_ID   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access | RW           |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit  | Name            | Function         | Description                                                  | Access |
|------|-----------------|------------------|--------------------------------------------------------------|--------|
| 31:0 | PROD-<br>UCT_ID | Product ID field | The ID field that can be programmed by the appli-<br>cation. | RW     |

### 35.6.3.12 LPM Configuration Register (USBFS\_GLPMCFG)

This register is a programmable LPM configuration register.

| Offset | Bit Position |    |        |    |            |    |         |    |         |    |          |    |         |    |        |    |        |    |        |    |          |    |        |   |         |   |      |   |        |   |       |   |
|--------|--------------|----|--------|----|------------|----|---------|----|---------|----|----------|----|---------|----|--------|----|--------|----|--------|----|----------|----|--------|---|---------|---|------|---|--------|---|-------|---|
| 0x54   | 31           | 30 | 29     | 28 | 27         | 26 | 25      | 24 | 23      | 22 | 21       | 20 | 19      | 18 | 17     | 16 | 15     | 14 | 13     | 12 | 11       | 10 | 9      | 8 | 7       | 6 | 5    | 4 | 3      | 2 | 1     | 0 |
| Reset  | 0x00000000   |    |        |    |            |    |         |    |         |    |          |    |         |    |        |    |        |    |        |    |          |    |        |   |         |   |      |   |        |   |       |   |
| Name   | Reserved     |    | ENBSEL |    | LPMRCNTSTS |    | SENDLPM |    | LPMRCNT |    | LPMCHIDX |    | L1RSMOK |    | SLPSTS |    | LPMRSP |    | L1DSEN |    | BSELTHRS |    | L1SSEN |   | REMWAKE |   | BSEL |   | LPMACK |   | LPMEN |   |
| Access |              |    | RW     |    | RO         |    | RW      |    | RW      |    | RW       |    | RO      |    | RO     |    | RO     |    | RW     |    | RW       |    | RW     |   | RW      |   | RW   |   | RW     |   | RW    |   |

| Bit   | Name       | Function                   | Description                                                                                                                                                                                                                                                                                                                                                                                                | Access |
|-------|------------|----------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:29 | Reserved   | Reserved                   | -                                                                                                                                                                                                                                                                                                                                                                                                          | -      |
| 28    | ENBSEL     | BSEL enable bit            | <ul style="list-style-type: none"> <li>0b0: Follow the USB 2.0 Link Power Management Addendum Engineering Change Notice to the USB 2.0 specification, July 16, 2007.</li> <li>0b1: Follow the Errata for USB 2.0 ECN Link Power Management (LPM) -7/2007</li> </ul>                                                                                                                                        | RW     |
| 27:25 | LPMRCNTSTS | LPM retry status           | Records the remaining number of LPM retries.<br><div>  <b>Note</b><br/>           Active only in host mode.         </div>                                                                                                                                                                                              | RO     |
| 24    | SENDLPM    | Send LPM transaction       | <ul style="list-style-type: none"> <li>0b0: Do not send LPM transaction.</li> <li>0b1: Send LPM transaction (EXT and LPM to-kens).</li> </ul> This bit is automatically cleared when the device returns a response (ACK, STALL, or NYET).<br><div>  <b>Note</b><br/>           Active only in host mode.         </div> | RW     |
| 23:21 | LPMRCNT    | LPM retry count            | When an ERROR response is received, the host retries the LPM transaction until a active response (ACK, STALL, or NYET) is received.<br><div>  <b>Note</b><br/>           Active only in host mode.         </div>                                                                                                       | RW     |
| 20:17 | LPMCHIDX   | LPM transmit channel index | The host channel index used for LPM transactions; the hardware automatically inserts the device address and endpoint information set for this channel.<br><div>  <b>Note</b><br/>           Active only in host mode.         </div>                                                                                    | RW     |

| Bit   | Name     | Function                    | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | Access |
|-------|----------|-----------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 16    | L1RSMOK  | L1 status bit               | <ul style="list-style-type: none"> <li>● 0b0: Current state cannot recover from L1.</li> <li>● 0b1: Current state can recover from L1.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | RO     |
| 15    | SLPSTS   | Sleep status bit            | <b>Device Mode:</b><br>When the device sends an ACK in response to an LPM transaction and the protocol-specified time has elapsed, this bit is set to 1, indicating entry into SLEEP mode.<br>When the bus state changes, disconnects, or a remote wakeup signal is sent, this bit is automatically cleared to 0, exiting SLEEP mode.<br><b>Host Mode:</b><br>The host sends an LPM transaction; after receiving an ACK response from the device, this bit is set to 1, indicating entry into SLEEP mode.<br>When the host receives a remote wakeup signal, initiates a wakeup, or initiates a reset, this bit is automatically cleared to 0, exiting SLEEP mode. <ul style="list-style-type: none"> <li>● 0b0: Not in L1</li> <li>● 0b1: In L1</li> </ul> | RO     |
| 14:13 | LPMRSP   | LPM response                | LPM response received by the host or sent by the device. <ul style="list-style-type: none"> <li>● 0b00: ERROR (no response).</li> <li>● 0b01: STALL</li> <li>● 0b10: NYET</li> <li>● 0b11: ACK</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | RO     |
| 12    | L1DSEN   | L1 deep sleep enable bit    | To minimize power consumption, the application should set this bit to 1.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | RW     |
| 11:8  | BSELTHRS | BSEL threshold setting      | <b>Device Mode:</b> When the device receives a BSEL value greater than or equal to the register setting, it enters L1.<br><b>Host Mode:</b> The time for the host to send a resume signal.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | RW     |
| 7     | L1SSEN   | L1 shallow sleep enable bit | To minimize power consumption, the application should set this bit to 1.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | RW     |
| 6     | REMWAKE  | bRemoteWake value           | bRemoteWake value sent by the host or received by the device.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | RW     |
| 5:2   | BSEL     | BSEL setting                | <b>Device Mode:</b> BSEL bmAttribute value, automatically updated after a active LPM transaction response.<br><b>Host Mode:</b> BSEL value sent by the host for LPM transactions, or resume signal length value. <ul style="list-style-type: none"> <li>● 0b0000: 125us</li> <li>● 0b0001: 150us</li> <li>● 0b0010: 200us</li> <li>● 0b0100: 400us</li> <li>● 0b0101: 500us</li> <li>● 0b0110: 1000us</li> </ul>                                                                                                                                                                                                                                                                                                                                           | RW     |



| Bit | Name   | Function       | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | Access |
|-----|--------|----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 5:2 | BSEL   | BSEL setting   | <ul style="list-style-type: none"> <li>● 0b0111: 2000us</li> <li>● 0b1000: 3000us</li> <li>● 0b1001: 4000us</li> <li>● 0b1010: 5000us</li> <li>● 0b1011: 6000us</li> <li>● 0b1100: 7000us</li> <li>● 0b1101: 8000us</li> <li>● 0b1110: 9000us</li> <li>● 0b1111: 10000us</li> <li>● Others: Setting prohibited</li> </ul>                                                                                                                                                                                                                                                                                                                                      | RW     |
| 1   | LPMACK | LPM response   | <ul style="list-style-type: none"> <li>● 0b0: NYET</li> <li>● 0b1: ACK</li> </ul> <p>Even if this bit is written as 1, an ACK response is not sent in the following scenarios (LPM transaction errors) :</p> <ul style="list-style-type: none"> <li>● PID/CRC error exists; in this scenario, an ERROR response is sent.</li> <li>● bLinkState is not equal to 0001b; in this scenario, a STALL response is sent.</li> <li>● Pending data transfer exists; in this scenario, a NYET response is sent.</li> </ul> <div>  <b>Note</b><br/> Active only in device mode. </div> | RW     |
| 0   | LPMEN  | LPM enable bit | <ul style="list-style-type: none"> <li>● 0b0: Disable</li> <li>● 0b1: Enable</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | RW     |

### 35.6.3.13 Host Periodic Transmit FIFO Size Register (USBFS\_HPTXFSIZ)

This application can program the RAM size that must be allocated to the periodic Tx FIFO.

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |       |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0x100  | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15    | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x02800500   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |       |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | PTXFD        |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | PTXSA |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access | RW           |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RW    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit   | Name  | Function                            | Description                                                                                                  | Access |
|-------|-------|-------------------------------------|--------------------------------------------------------------------------------------------------------------|--------|
| 31:16 | PTXFD | Host periodic Tx FIFO depth         | In units of 32-bit words, the minimum value is 16.                                                           | RW     |
| 15:0  | PTXSA | Host periodic Tx FIFO start address | The power-on reset value is the sum of the maximum Rx FIFO depth and the maximum non-periodic Tx FIFO depth. | RW     |

### 35.6.3.14 Device IN Endpoint Transmit FIFO Size Register (USBFS\_DIEPTXFx) (x=1-15)

This application can program the size that must be allocated to the device TxFIFO.

Offset:  $0x100 + 0x4 \times x$

| Offset | Bit Position           |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |          |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|------------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| -      | 31                     | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15       | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x02800500+0x280*(x-1) |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |          |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | INEPTXFD               |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | INEPTXSA |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access | RW                     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RW       |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit   | Name     | Function                                     | Description                                                                                                                                        | Access |
|-------|----------|----------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:16 | INEPTXFD | The TxFIFO depth of the device IN end-point. | In units of 32-bit words, the minimum value is 16.                                                                                                 | RW     |
| 15:0  | INEPTXSA | IN endpoint TxFIFOx RAM start address        | This field contains the starting address of the memory of the IN endpoint send FIFOx. The address must be aligned with the 32-bit memory location. | RW     |

### 35.6.4 USBFS Host Mode Register

The host mode register affects the module operation in host mode. In device mode, accessing the host mode register is prohibited as it yields undefined results.

#### 35.6.4.1 Host Configuration Register (USBFS\_HCFG)

This register configures the module after power-on. Do not modify this register after initializing the host.

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |       |   |    |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|-------|---|----|---|
| 0x400  | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3     | 2 | 1  | 0 |
| Reset  | 0x00000200   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |       |   |    |   |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   | FSLSS |   |    |   |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   | RW    |   | RW |   |

| Bit  | Name     | Function                                       | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | Access |
|------|----------|------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:3 | Reserved | Reserved                                       | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | -      |
| 2    | FSLSS    | Only supports FS and LS function control bits. | <p>The application uses this bit to control the enumeration speed of the module. Using this bit, the application can make the module operate as an FS host, even if the connected device supports HS communication. Do not change this field after initial programming.</p> <ul style="list-style-type: none"> <li>● 0b0: Setting prohibited</li> <li>● 0b1: Only FS/LS, even if the connected device can support HS</li> </ul>                                                                                                                                                                                                                                                        | RW     |
| 1:0  | FSLSPCS  | FS/LS PHY clock select                         | <p>When the module is in FS host mode:</p> <ul style="list-style-type: none"> <li>● 0b01: PHY clock runs at 48MHz</li> <li>● Others: Setting prohibited</li> </ul> <p>When the module is in LS host mode:</p> <ul style="list-style-type: none"> <li>● 0b01: PHY clock runs at 48MHz</li> <li>● 0b10: PHY clock runs at 6MHz</li> <li>● Others: Setting prohibited</li> </ul> <div>  <b>Notice</b><br/>           When a device is connected to the host, FSLSPCS must be set according to the speed of the connected device (a software reset is required after changing this bit).         </div> | RW     |

### 35.6.4.2 Host Frame Time Interval Register (USBFS\_HFIR)

This register stores the frame interval information set by the USBFS controller for the current speed of the connected device.

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |       |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0x404  | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15    | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x0000EA60   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |       |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | FRIVL |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RW    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit   | Name     | Function       | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | Access |
|-------|----------|----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:16 | Reserved | Reserved       | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | -      |
| 15:0  | FRIVL    | Frame interval | <p>The value programmed by the application in this field specifies the time interval between two consecutive SOF (FS) or Keep-Alive tokens (LS). This field contains the number of PHY clocks that constitute the desired frame interval. The application can only write values to this register after setting the port enable bit (PENA bit in USBFS_HPRT) of the host port control and status register to 1. If no value is programmed, the module calculates based on the PHY clock specified in the FS/LS PHY clock selection field (FSLSPCS in USBFS_HCFG) of the host configuration register. Do not change the value of this field after initial configuration. Set value=frame interval (ms) × (PHY clock frequency) - 1.</p> <div>  <b>Notice</b><br/>           When the application needs to change the frame interval time, it can modify the FRIVL bit.         </div> | RW     |

## 35.6.4.3 Host Frame Number/Frame Remaining Time Register (USBFS\_HFNUM)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |       |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0x408  | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15    | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00003FFF   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |       |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | FTREM        |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | FRNUM |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access | RO           |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RO    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit   | Name  | Function             | Description                                                                                                                                                                                                                                                                        | Access |
|-------|-------|----------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:16 | FTREM | Frame time remaining | Indicates the remaining time of the current frame (in units of PHY clock counts). This field decrements by 1 for each PHY clock that passes. When the value reaches zero, this field reloads the value from the frame interval register and the module sends a new SOF on the USB. | RO     |
| 15:0  | FRNUM | Frame number         | When a new SOF is sent on the USB, the value of this field increments by 1, and it clears to zero when it reaches 0x3FFF.                                                                                                                                                          | RO     |

### 35.6.4.4 Host Periodic Transmit FIFO/Queue Status Register (USBFS\_HPTXSTS)

This read-only register contains information about the free space in the periodic Tx FIFO and periodic transmit request queue.

| Offset | Bit Position |    |    |    |    |    |    |    |         |    |    |    |    |    |    |    |          |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|---------|----|----|----|----|----|----|----|----------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0x410  | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23      | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15       | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00080280   |    |    |    |    |    |    |    |         |    |    |    |    |    |    |    |          |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | PTXQTOP      |    |    |    |    |    |    |    | PTXQSAV |    |    |    |    |    |    |    | PTXFSAVL |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access | RO           |    |    |    |    |    |    |    | RO      |    |    |    |    |    |    |    | RO       |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit   | Name     | Function                                        | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | Access |
|-------|----------|-------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:24 | PTXQTOP  | Top of the periodic transmit request queue      | Indicates the item currently being processed by the MAC in the periodic Tx request queue. This register is used for debugging.<br>[31]: Even/odd frame<br><ul style="list-style-type: none"> <li>0b0: Transmit in even frames</li> <li>0b1: Transmit in odd frames</li> </ul> [30:27]: Channel/endpoint number<br>[26:25]: Type<br><ul style="list-style-type: none"> <li>0b00: Input/Output</li> <li>0b01: Zero-length data packet</li> <li>0b10: Reserved</li> <li>0b11: Disable channel command</li> </ul> [24]: Terminate (Last entry for the selected channel) | RO     |
| 23:16 | PTXQSAV  | Periodic transmit request queue space available | Indicates the number of free positions available for writing in the periodic transmit request queue, ranging from 0 to 8. This queue contains both IN and OUT requests.<br><ul style="list-style-type: none"> <li>0: The periodic transmit request queue is full</li> <li>1: 1 position available.</li> <li>.....</li> <li>8: 8 positions available.</li> </ul>                                                                                                                                                                                                     | RO     |
| 15:0  | PTXFSAVL | Periodic transmit data FIFO space available     | Indicates the number of free positions available for writing in the periodic Tx FIFO. In units of 32-bit words.<br><ul style="list-style-type: none"> <li>0: The periodic Tx FIFO is full</li> <li>1: 1 word available.</li> <li>.....</li> <li>n: n words are available (n is the value of the register USBFS_HPTXFSIZ.PTXFD).</li> </ul>                                                                                                                                                                                                                          | RO     |

### 35.6.4.5 Host All Channels Interrupt Register (USBFS\_HAINT)

When an event occurs on the channel, the host all-channel interrupt register interrupts the application using the host channel interrupt bit in the module interrupt register (the HCINT bit in USBFS\_GINTSTS). Each channel corresponds to one interrupt bit, with a maximum of 16 bits.

When the application clears the interrupt through the corresponding host channel x interrupt register, the bit in this register is also cleared.

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |       |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0x414  | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15    | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |       |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | HAINT |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RO    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit   | Name     | Function          | Description                                                                                                                                                                                                                                                                                                                                                                       | Access |
|-------|----------|-------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:16 | Reserved | Reserved          | -                                                                                                                                                                                                                                                                                                                                                                                 | -      |
| 15:0  | HAINT    | Channel interrupt | Corresponding to Channel 15 to Channel 0: <ul style="list-style-type: none"> <li>● 0b0: No interrupt was generated on this channel</li> <li>● 0b1: An interrupt was generated on this channel</li> </ul> <div>  <b>Note</b><br/>           Each bit corresponds to one channel.         </div> | RO     |

### 35.6.4.6 Host All Channels Interrupt Mask Register (USBFS\_HAINTMSK)

The host all-channel interrupt mask register is used in conjunction with the host all-channel interrupt register to interrupt the application when an event occurs on the channel.

Each channel corresponds to one interrupt mask bit, with a maximum of 16 bits.

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |        |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|--------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0x418  | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15     | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |        |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | HAINTM |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RW     |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit   | Name     | Function               | Description                                                                                                                                                                                                                                                                                                                | Access |
|-------|----------|------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:16 | Reserved | Reserved               | -                                                                                                                                                                                                                                                                                                                          | -      |
| 15:0  | HAINTM   | Channel interrupt mask | Corresponding to Channel 15 to Channel 0: <ul style="list-style-type: none"> <li>0b0: Maskable interrupt</li> <li>0b1: Enable interrupt</li> </ul> <div>  <b>Note</b><br/>           Each bit corresponds to one channel.         </div> | RW     |



### 35.6.4.7 Host Port Control and Status Register (USBFS\_HPRT)

This register is only available in host mode. Currently, the USBFS host supports only one port.

This register contains USB port-related information, such as USB reset, enable, suspend, resume, and connection status. The PENCHNG/PCDET bit in this register can trigger an application interrupt through the host port interrupt bit (HPRTINT bit in USBFS\_GINTST) in the module interrupt register. When a port interrupt occurs, the application must read this register and clear the bits that triggered the interrupt. The application must write 1 to this bit to clear the interrupt.

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |      |          |    |    |    |    |      |       |    |          |      |       |    |      |          |   |         |      |       |       |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|------|----------|----|----|----|----|------|-------|----|----------|------|-------|----|------|----------|---|---------|------|-------|-------|
| 0x440  | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18   | 17       | 16 | 15 | 14 | 13 | 12   | 11    | 10 | 9        | 8    | 7     | 6  | 5    | 4        | 3 | 2       | 1    | 0     |       |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |      |          |    |    |    |    |      |       |    |          |      |       |    |      |          |   |         |      |       |       |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    | PSPD | Reserved |    |    |    |    | PWPR | PLSTS |    | Reserved | PRST | PSUSP |    | PRES | Reserved |   | PENCHNG | PENA | PCDET | PCSTS |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    | RO   |          |    |    |    |    | RW   | RO    |    |          | RW   | RW    | RW |      |          |   | RW      | RW   | RW    | RO    |

| Bit   | Name     | Function         | Description                                                                                                                                                                                                                                                                                                                                                                                                             | Access |
|-------|----------|------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:19 | Reserved | Reserved         | -                                                                                                                                                                                                                                                                                                                                                                                                                       | -      |
| 18:17 | PSPD     | Port speed       | Indicates the speed of the device connected to this port.<br><ul style="list-style-type: none"> <li>0b01: Full speed</li> <li>0b10: Low speed</li> <li>0b00/0b11: Reserved</li> </ul>                                                                                                                                                                                                                                   | RO     |
| 16:13 | Reserved | Reserved         | -                                                                                                                                                                                                                                                                                                                                                                                                                       | -      |
| 12    | PWPR     | Port power       | The application uses this field to control the power supply of this port. Since the internal PHY of this USBFS does not have power-supply capability, setting this bit to 1 enables the external USB power chip via USBFS_DRVVBUS.<br><ul style="list-style-type: none"> <li>0b0: Power off</li> <li>0b1: Power-on</li> </ul>                                                                                           | RW     |
| 11:10 | PLSTS    | Port line status | Indicates the current logic level of the USB data line.<br>[11]: Logic level of USBFS_DM<br>[10]: Logic level of USBFS_DP                                                                                                                                                                                                                                                                                               | RO     |
| 9     | Reserved | Reserved         | -                                                                                                                                                                                                                                                                                                                                                                                                                       | -      |
| 8     | PRST     | Port reset       | When the application sets this bit to 1, it initiates a reset sequence on this port. The application must time the reset period and clear this bit after the reset sequence is completed.<br><ul style="list-style-type: none"> <li>0b0: Port is not in reset state</li> <li>0b1: Port is in reset state</li> </ul> The application must set this bit to 1 and hold it for at least 10ms to initiate reset on the port. | RW     |

| Bit | Name            | Function                   | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | Access |
|-----|-----------------|----------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 7   | PSUSP           | Port suspend               | <p>The application sets this bit to 1 to place this port in suspend mode. The module stops sending SOF only when this bit is set to 1. To stop the PHY clock, the application must set the port clock stop bit to 1, which enables the PHY's suspend input pin.</p> <p>The read value of this bit reflects the current suspend status of the port. Upon detecting a remote wake-up signal or when the application sets the port reset bit or port resume bit in this register to 1, the module can clear this bit. Alternatively, the module can also clear this bit when the application sets the resume/remote wake-up detection interrupt bit or disconnect detection interrupt bit (WKUINT or DISCINT in USBFS_GINTSTS, respectively) in the module interrupt register to 1.</p> <ul style="list-style-type: none"> <li>● 0b0: Port is not in suspend mode</li> <li>● 0b1: Port is in suspend mode</li> </ul> | RW     |
| 6   | PRES            | Port resume                | <p>The application sets this bit to 1 to drive a resume signal on this port. The module will continuously drive the resume signal until the application clears this bit.</p> <p>As indicated by the port resume/remote wake-up detection interrupt bit (WKUINT bit in USBFS_GINTSTS) in the module interrupt register, if the module detects a USB remote wakeup sequence, it will start driving the resume signal without application intervention; if the module detects a disconnection event, this bit will be cleared.</p> <p>The read value of this bit indicates whether the current module is driving the resume signal.</p> <ul style="list-style-type: none"> <li>● 0b0: Resume signal not driven</li> <li>● 0b1: Drive resume signal</li> </ul>                                                                                                                                                        | RW     |
| 5:4 | <i>Reserved</i> | <i>Reserved</i>            | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | -      |
| 3   | PENCHNG         | Port enable/disable change | <p>When the state of port enable bit 2 in this register changes, the module sets this bit to 1.</p> <p>Writing 1 to this bit clears it via software.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | RW     |
| 2   | PENA            | Port enable                | <p>After the port executes the reset sequence, it can only be enabled by the module, and can be disabled by clearing this bit through overcurrent conditions, disconnection events, or application commands. The application cannot set this bit to 1 by writing to the register. This bit can only be cleared to disable the port. Operations on this bit will not trigger any application interrupts.</p> <ul style="list-style-type: none"> <li>● 0b0: Port disabled</li> <li>● 0b1: Port enabled</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                   | RW     |

| Bit | Name  | Function              | Description                                                                                                                                                                                                                                                               | Access |
|-----|-------|-----------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 1   | PCDET | Port connect detected | When a device connection is detected, the module sets this bit to 1 to trigger an application interrupt using the host port interrupt bit (HPRTINT bit in USBFS_GINTSTS) in the module interrupt register. The application must set this bit to 1 to clear the interrupt. | RW     |
| 0   | PCSTS | Port connect status   | <ul style="list-style-type: none"><li>● 0b0: Port not connected to device</li><li>● 0b1: Port device connected</li></ul>                                                                                                                                                  | RO     |

## 35.6.4.8 Host Channel x Characteristics Register (USBFS\_HCCHARx) (x=0-15)

This register is used to set the host channel characteristics.

Offset: 0x500+0x20\*x

| Offset | Bit Position |       |        |     |    |    |    |    |    |    |    |          |       |       |          |       |       |    |    |    |       |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|-------|--------|-----|----|----|----|----|----|----|----|----------|-------|-------|----------|-------|-------|----|----|----|-------|----|---|---|---|---|---|---|---|---|---|---|
| -      | 31           | 30    | 29     | 28  | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20       | 19    | 18    | 17       | 16    | 15    | 14 | 13 | 12 | 11    | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |       |        |     |    |    |    |    |    |    |    |          |       |       |          |       |       |    |    |    |       |    |   |   |   |   |   |   |   |   |   |   |
| Name   | CHENA        | CHDIS | ODDFRM | DAD |    |    |    |    |    |    |    | Reserved | EPTYP | LSDEV | Reserved | EPDIR | EPNUM |    |    |    | MPSIZ |    |   |   |   |   |   |   |   |   |   |   |
| Access | RW           | RW    | RW     | RW  |    |    |    |    |    |    |    |          | RW    | RW    |          | RW    | RW    |    |    |    | RW    |    |   |   |   |   |   |   |   |   |   |   |

| Bit   | Name     | Function         | Description                                                                                                                                                                                                                                                                                                         | Access |
|-------|----------|------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31    | CHENA    | Channel enable   | This field is set to 1 by application software and cleared by USBFS host hardware.<br><ul style="list-style-type: none"> <li>0b0: Disable channel</li> <li>0b1: Enable channel</li> </ul>                                                                                                                           | RW     |
| 30    | CHDIS    | Channel disable  | The application sets this bit to 1 to stop sending/receiving data through the channel. The stop operation takes effect even if the transmission via this channel has not yet completed.<br>The application must wait for the channel disabled interrupt to confirm that the channel has been disabled.              | RW     |
| 29    | ODDFRM   | Odd frame        | This field is set or reset by the application to indicate whether the USBFS host must transmit an odd or even frame, respectively. This field is only applicable to periodic (synchronous and interrupt) transactions.<br><ul style="list-style-type: none"> <li>0b0: Even frame</li> <li>0b1: Odd frame</li> </ul> | RW     |
| 28:22 | DAD      | Device address   | This field is used to specify the particular device to communicate with the host.                                                                                                                                                                                                                                   | RW     |
| 21:20 | Reserved | Reserved         | -                                                                                                                                                                                                                                                                                                                   | -      |
| 19:18 | EPTYP    | Endpoint type    | Indicates the selected transport type.<br><ul style="list-style-type: none"> <li>0b00: Control</li> <li>0b01: Isochronous</li> <li>0b10: Bulk</li> <li>0b11: Interrupt</li> </ul>                                                                                                                                   | RW     |
| 17    | LSDEV    | Low-speed device | This field is set by the application to indicate that this channel is communicating with a low-speed device.                                                                                                                                                                                                        | RW     |
| 16    | Reserved | Reserved         | -                                                                                                                                                                                                                                                                                                                   | -      |

| Bit   | Name  | Function            | Description                                                                                                                                                             | Access |
|-------|-------|---------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 15    | EPDIR | Endpoint direction  | Indicates whether the direction of the communication transaction is input or output. <ul style="list-style-type: none"><li>● 0b0: Output</li><li>● 0b1: Input</li></ul> | RW     |
| 14:11 | EPNUM | Endpoint number     | Indicates the endpoint number of the USB device to communicate with this host channel.                                                                                  | RW     |
| 10:0  | MPSIZ | Maximum packet size | Indicates the maximum packet size for device endpoints communicating with this host channel.                                                                            | RW     |

### 35.6.4.9 Host Channel x Interrupt Register (USBFS\_HCINTx) (x=0-15)

This register indicates the state of the channel when USB and AHB-related events occur. When the host channel interrupt bit (HCINT bit in USBFS\_GINTSTS) in the module interrupt register is set to 1, the application must read this register. Before performing a read operation on the register, the application must first read the host all-channel interrupt (USBFS\_HAINT) register to obtain the accurate channel number for the host channel x interrupt register. The application must clear the corresponding bits in this register to clear the corresponding bits in the USBFS\_HAINT and USBFS\_GINTSTS registers.

Offset: 0x508+0x20\*x

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |       |       |       |       |          |     |     |       |        |     |      |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-------|-------|-------|-------|----------|-----|-----|-------|--------|-----|------|
| -      | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10    | 9     | 8     | 7     | 6        | 5   | 4   | 3     | 2      | 1   | 0    |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |       |       |       |       |          |     |     |       |        |     |      |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | DTERR | FRMOR | BBERR | TXERR | Reserved | ACK | NAK | STALL | AHBERR | CHH | XFRC |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RW    | RW    | RW    | RW    |          | RW  | RW  | RW    | RW     | RW  | RW   |

| Bit   | Name     | Function                           | Description                                                                                                                                                                                                                                             | Access |
|-------|----------|------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:11 | Reserved | Reserved                           | -                                                                                                                                                                                                                                                       | -      |
| 10    | DTERR    | Data synchronization error         | The application must clear this bit by writing 1.                                                                                                                                                                                                       | RW     |
| 9     | FRMOR    | Frame overflow error               | The application must clear this bit by writing 1.                                                                                                                                                                                                       | RW     |
| 8     | BBERR    | Crosstalk error                    | A typical cause of crosstalk events is that the endpoint sent a packet, but the packet length exceeded the maximum packet length of the endpoint. The application must clear this bit by writing 1.                                                     | RW     |
| 7     | TXERR    | Communication transaction error    | Indicates one of the following errors occurred on the USB: <ul style="list-style-type: none"> <li>● CRC check failed</li> <li>● Timeout</li> <li>● Bit stuffing error</li> <li>● Wrong EOP</li> </ul> The application must clear this bit by writing 1. | RW     |
| 6     | Reserved | Reserved                           | -                                                                                                                                                                                                                                                       | -      |
| 5     | ACK      | ACK responses received/transmitted | The application must clear this bit by writing 1.                                                                                                                                                                                                       | RW     |
| 4     | NAK      | NAK response received              | The application must clear this bit by writing 1.                                                                                                                                                                                                       | RW     |
| 3     | STALL    | STALL response received            | The application must clear this bit by writing 1.                                                                                                                                                                                                       | RW     |

| Bit | Name   | Function                | Description                                                                                                                                                                                                                                                                     | Access |
|-----|--------|-------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 2   | AHBERR | AHB error               | This error is generated only when in internal DMA mode and an AHB error occurs during an AHB read/write operation. The application can obtain the error address by reading the corresponding DMA channel address register.<br>The application must clear this bit by writing 1. | RW     |
| 1   | CHH    | Channel halted          | The transfer ended abnormally due to any USB transaction error or in response to the application's disable request.<br>The application must clear this bit by writing 1.                                                                                                        | RW     |
| 0   | XFRC   | Transmission completion | Transfer completed normally without any errors.<br>The application must clear this bit by writing 1.                                                                                                                                                                            | RW     |

### 35.6.4.10 Host Channel x Interrupt Mask Register (USBFS\_HCINTMSKx) (x=0-15)

This register is used to select the mask for the host channel interrupts.

Offset: 0x50C+0x20\*x

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |        |        |        |        |          |      |      |        |         |      |       |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|--------|--------|--------|--------|----------|------|------|--------|---------|------|-------|
| -      | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9      | 8      | 7      | 6      | 5        | 4    | 3    | 2      | 1       | 0    |       |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |        |        |        |        |          |      |      |        |         |      |       |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | DTERRM | FRMORM | BBERRM | TXERRM | Reserved | ACKM | NAKM | STALLM | AHBERRM | CHHM | XFCRM |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RW     | RW     | RW     | RW     |          | RW   | RW   | RW     | RW      | RW   | RW    |

| Bit   | Name     | Function                                                | Description                                                                                              | Access |
|-------|----------|---------------------------------------------------------|----------------------------------------------------------------------------------------------------------|--------|
| 31:11 | Reserved | Reserved                                                | -                                                                                                        | -      |
| 10    | DTERRM   | Data synchroni-<br>zation error<br>mask                 | <ul style="list-style-type: none"> <li>0b0: Maskable interrupt</li> <li>0b1: Enable interrupt</li> </ul> | RW     |
| 9     | FRMORM   | Frame overflow<br>mask                                  | <ul style="list-style-type: none"> <li>0b0: Maskable interrupt</li> <li>0b1: Enable interrupt</li> </ul> | RW     |
| 8     | BBERRM   | Crosstalk error<br>mask                                 | <ul style="list-style-type: none"> <li>0b0: Maskable interrupt</li> <li>0b1: Enable interrupt</li> </ul> | RW     |
| 7     | TXERRM   | Communication<br>transaction er-<br>ror mask            | <ul style="list-style-type: none"> <li>0b0: Maskable interrupt</li> <li>0b1: Enable interrupt</li> </ul> | RW     |
| 6     | Reserved | Reserved                                                | -                                                                                                        | -      |
| 5     | ACKM     | ACK response<br>receive/trans-<br>mit interrupt<br>mask | <ul style="list-style-type: none"> <li>0b0: Maskable interrupt</li> <li>0b1: Enable interrupt</li> </ul> | RW     |
| 4     | NAKM     | NAK response<br>receive inter-<br>rupt mask             | <ul style="list-style-type: none"> <li>0b0: Maskable interrupt</li> <li>0b1: Enable interrupt</li> </ul> | RW     |
| 3     | STALLM   | Receive STALL<br>response inter-<br>rupt mask           | <ul style="list-style-type: none"> <li>0b0: Maskable interrupt</li> <li>0b1: Enable interrupt</li> </ul> | RW     |
| 2     | AHBERRM  | AHBERR inter-<br>rupt mask                              | <ul style="list-style-type: none"> <li>0b0: Maskable interrupt</li> <li>0b1: Enable interrupt</li> </ul> | RW     |
| 1     | CHHM     | Channel stop<br>interrupt mask                          | <ul style="list-style-type: none"> <li>0b0: Maskable interrupt</li> <li>0b1: Enable interrupt</li> </ul> | RW     |
| 0     | XFCRM    | Transfer com-<br>pleted interrupt<br>mask               | <ul style="list-style-type: none"> <li>0b0: Maskable interrupt</li> <li>0b1: Enable interrupt</li> </ul> | RW     |



### 35.6.4.11 Host Channel x Transfer Size Register (USBFS\_HCTSIZx) (x=0-15)

This register is used to set the host channel transfer size and data PID.

Offset: 0x510+0x20\*x

| Offset | Bit Position |      |    |        |    |    |    |    |    |    |    |    |    |        |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|------|----|--------|----|----|----|----|----|----|----|----|----|--------|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| -      | 31           | 30   | 29 | 28     | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18     | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |      |    |        |    |    |    |    |    |    |    |    |    |        |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | Reserved     | DPID |    | PKTCNT |    |    |    |    |    |    |    |    |    | XFRSIZ |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|        |              |      |    |        |    |    |    |    |    |    |    |    |    |        |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access |              | RW   |    | RW     |    |    |    |    |    |    |    |    |    | RW     |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit   | Name     | Function      | Description                                                                                                                                                                                                                                                                                                                                | Access |
|-------|----------|---------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31    | Reserved | Reserved      | -                                                                                                                                                                                                                                                                                                                                          | -      |
| 30:29 | DPID     | Data PID      | The application sets the initial synchronization PID for data communication in this field.<br>The host retains the setting of this field during this transfer transaction.<br><ul style="list-style-type: none"> <li>● 0b00: DATA0</li> <li>● 0b01: DATA2</li> <li>● 0b10: DATA1</li> <li>● 0b11: MDATA/SETUP</li> </ul>                   | RW     |
| 28:19 | PKTCNT   | Packet count  | The application sets the number of data packets to be sent or received in this field.<br>The host decrements the count value each time a data packet is successfully sent or received. When this value reaches 0, the application is interrupted to indicate normal completion of the operation.                                           | RW     |
| 18:0  | XFRSIZ   | Transfer size | For OUT operations, this field is the number of data bytes sent by the host during the transfer.<br>For IN operations, this field is the buffer size reserved by the application for the transfer. For IN transactions (periodic and non-periodic), the application programs this field as an integer multiple of the maximum packet size. | RW     |

35.6.4.12 Host Channel x DMA Address Register (USBFS\_HCDMAx) (x=0-15)

This register is used to set the DMA address when the host is in DMA mode.

Offset: 0x514+0x20\*x

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| -      | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0xFFFFFFFF   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | DMAADDR      |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access | RW           |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit  | Name    | Function    | Description                                                                                                                                                                                                            | Access |
|------|---------|-------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:0 | DMAADDR | DMA address | This field stores the address of the memory used for DMA transfers when the host obtains data from the device endpoint or sends data to the device endpoint. This register increments at the end of each AHB transfer. | RW     |

### 35.6.5 USBFS Device Mode Register

The device mode register affects the module operation in device mode. In host mode, accessing the device mode register is prohibited as it yields undefined results.

#### 35.6.5.1 Device Configuration Register (USBFS\_DCFG)

This register configures the module into device mode after power-on, execution of certain control commands, or enumeration. Do not change this register after initial programming.

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |       |    |     |    |    |    |          |   |          |   |      |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-------|----|-----|----|----|----|----------|---|----------|---|------|---|---|---|---|---|
| 0x800  | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15    | 14 | 13  | 12 | 11 | 10 | 9        | 8 | 7        | 6 | 5    | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x08200000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |       |    |     |    |    |    |          |   |          |   |      |   |   |   |   |   |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | PFIVL |    | DAD |    |    |    | Reserved |   | NZLSOHSK |   | DSPD |   |   |   |   |   |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RW    |    | RW  |    |    |    |          |   | RW       |   | RW   |   |   |   |   |   |

| Bit   | Name     | Function                             | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | Access |
|-------|----------|--------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:13 | Reserved | Reserved                             | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | -      |
| 12:11 | PFIVL    | Periodic frame interval              | Indicates the point in time within a frame when the periodic frame interrupt must be used to notify the application. This function can be used to determine whether all synchronous communications for this frame are completed. <ul style="list-style-type: none"> <li>0b00: 80% frame interval</li> <li>0b01: 85% frame interval</li> <li>0b10: 90% frame interval</li> <li>0b11: 95% frame interval</li> </ul>                                                                                                                                                                                                                            | RW     |
| 10:4  | DAD      | Device address                       | The application must set this field according to the command parameters after executing each SetAddress control command.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | RW     |
| 3     | Reserved | Reserved                             | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | -      |
| 2     | NZLSOHSK | Non-zero-length status OUT handshake | During the OUT transaction of the control transfer status stage, when the module receives a non-zero-length packet, the application can use this field to select the handshake signal to be sent. <ul style="list-style-type: none"> <li>0b0: Send the received OUT packet (zero-length or non-zero-length) to the application and reply with a handshake signal based on the NAK and STALL bits of the endpoint in the device endpoint control register.</li> <li>0b1: When receiving a non-zero-length status OUT transaction, reply with a STALL handshake signal, and the received OUT packet is not sent to the application.</li> </ul> | RW     |

| Bit | Name | Function     | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | Access |
|-----|------|--------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 1:0 | DSPD | Device speed | <p>Indicates the speed at which the application requires the module to enumerate or the maximum speed supported by the application. However, the actual bus speed can only be determined after completing the chirp sequence, and this speed is based on the speed of the USB host connected to the module.</p> <ul style="list-style-type: none"><li>● 0b00: Setting prohibited</li><li>● 0b01: Setting prohibited</li><li>● 0b10: Setting prohibited</li><li>● 0b11: Full-speed (USB1.1 transceiver clock is 48MHz)</li></ul> | RW     |

## 35.6.5.2 Device Control Register (USBFS\_DCTL)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |          |    |        |   |        |   |        |   |        |   |          |   |        |  |        |  |      |  |        |  |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----------|----|--------|---|--------|---|--------|---|--------|---|----------|---|--------|--|--------|--|------|--|--------|--|
| 0x804  | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11       | 10 | 9      | 8 | 7      | 6 | 5      | 4 | 3      | 2 | 1        | 0 |        |  |        |  |      |  |        |  |
| Reset  | 0x00000002   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |          |    |        |   |        |   |        |   |        |   |          |   |        |  |        |  |      |  |        |  |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | POPRGDNE |    | CGONAK |   | SGONAK |   | CGINAK |   | SGINAK |   | Reserved |   | GONSTS |  | GINSTS |  | SDIS |  | RWUSIG |  |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RW       |    | RW     |   | RW     |   | RW     |   | RW     |   |          |   | RO     |  | RO     |  | RW   |  | RW     |  |

| Bit   | Name     | Function                  | Description                                                                                                                                                                                                                                                                                                                                                                       | Access |
|-------|----------|---------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:12 | Reserved | Reserved                  | -                                                                                                                                                                                                                                                                                                                                                                                 | -      |
| 11    | POPRGDNE | Power-on programming done | The application uses this bit to indicate that the register has completed programming after waking up from Power-down mode.                                                                                                                                                                                                                                                       | RW     |
| 10    | CGONAK   | Clear global OUT NAK      | Writing to this bit clears the global OUT NAK.                                                                                                                                                                                                                                                                                                                                    | RW     |
| 9     | SGONAK   | Set global OUT NAK        | Writing to this bit sets the global OUT NAK. The application uses this bit to send NAK handshake signals on all OUT endpoints. The application can only set this bit to 1 after confirming that the global OUT NAK active bit (GONAKEFF bit in USBFS_GINTSTS) in the module interrupt register has been cleared.                                                                  | RW     |
| 8     | CGINAK   | Clear global IN NAK       | Writing to this bit clears the global IN NAK.                                                                                                                                                                                                                                                                                                                                     | RW     |
| 7     | SGINAK   | Set global IN NAK         | Writing to this field sets the global non-periodic IN NAK to 1. The application uses this bit to make all non-periodic IN endpoints send NAK handshake signals. The application can only set this bit to 1 after confirming that the global IN NAK active bit (GINAKEFF bit in USBFS_GINTSTS) in the module interrupt register has been cleared.                                  | RW     |
| 6:4   | Reserved | Reserved                  | -                                                                                                                                                                                                                                                                                                                                                                                 | -      |
| 3     | GONSTS   | Global OUT NAK status     | <ul style="list-style-type: none"> <li>0b0: Handshake signals will be sent based on FIFO status and NAK/STALL bit settings.</li> <li>0b1: Do not receive data regardless of whether there is free space in the RxFIFO. For all received packets except SETUP transactions, respond with a NAK handshake signal. All OUT packets of synchronous type will be discarded.</li> </ul> | RO     |

| Bit | Name   | Function                 | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | Access |
|-----|--------|--------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 2   | GINSTS | Global IN NAK status     | <ul style="list-style-type: none"> <li>● 0b0: The handshake signal will be replied based on data availability in the transmit FIFO.</li> <li>● 0b1: Enable all non-periodic IN endpoints to respond with NAK handshake, regardless of data availability in the transmit FIFO.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | RO     |
| 1   | SDIS   | Soft disconnect          | <p>The application uses this bit to signal the USBFS module to perform a soft disconnect. When this bit is set to 1, the host will not detect the device as connected, and the device will not receive signals on the USB. The module remains disconnected until the application clears this bit.</p> <ul style="list-style-type: none"> <li>● 0b0: Normal operation. This bit is cleared after a soft disconnect, causing the host to receive a device connected event. After reconnecting the device, the USB host will restart device enumeration.</li> <li>● 0b1: Enable the host to receive the device disconnection event.</li> </ul> <p>At full speed, the minimum duration for soft disconnect is specified as follows:</p> <ul style="list-style-type: none"> <li>● Suspended state: the minimum time is 1ms+2.5μs</li> <li>● Idle state: 2.5μs</li> <li>● Non-idle or suspend status: 2.5μs</li> </ul> | RW     |
| 0   | RWUSIG | Remote wake-up signaling | <p>When the application sets this bit to 1, the module initiates a remote wake-up signal to wake the USB host. The application must set this bit to 1 to make the module exit the suspend status. According to the USB 2.0 specification, the application must clear this bit within 1ms to 15ms after setting it to 1.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | RW     |

### 35.6.5.3 Device Status Register (USBFS\_DSTS)

This register indicates the status of the module when USB-related events occur. When an interrupt occurs, the endpoint information of the interrupt must be read from the device all-segment endpoint interrupt register (USBFS\_DAINTE).

| Offset | Bit Position |    |    |    |    |    |    |    |       |    |       |    |    |    |    |    |    |    |    |    |    |    |          |   |   |   |      |         |   |         |   |   |
|--------|--------------|----|----|----|----|----|----|----|-------|----|-------|----|----|----|----|----|----|----|----|----|----|----|----------|---|---|---|------|---------|---|---------|---|---|
| 0x808  | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23    | 22 | 21    | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9        | 8 | 7 | 6 | 5    | 4       | 3 | 2       | 1 | 0 |
| Reset  | 0x00000002   |    |    |    |    |    |    |    |       |    |       |    |    |    |    |    |    |    |    |    |    |    |          |   |   |   |      |         |   |         |   |   |
| Name   | Reserved     |    |    |    |    |    |    |    | LNSTS |    | FNSOF |    |    |    |    |    |    |    |    |    |    |    | Reserved |   |   |   | EERR | ENUMSPD |   | SUSPSTS |   |   |
| Access |              |    |    |    |    |    |    |    | RO    |    | RO    |    |    |    |    |    |    |    |    |    |    |    |          |   |   |   | RO   | RO      |   | RO      |   |   |

| Bit   | Name     | Function                 | Description                                                                                                                                                                                                                                                                                                                                                                                                                                               | Access |
|-------|----------|--------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:24 | Reserved | Reserved                 | -                                                                                                                                                                                                                                                                                                                                                                                                                                                         | -      |
| 23:22 | LNSTS    | USB bus status           | [23]: Logical level of D+.<br>[22]: Logical level of D-.                                                                                                                                                                                                                                                                                                                                                                                                  | RO     |
| 21:8  | FNSOF    | Receive SOF frame number | -                                                                                                                                                                                                                                                                                                                                                                                                                                                         | RO     |
| 7:4   | Reserved | Reserved                 | -                                                                                                                                                                                                                                                                                                                                                                                                                                                         | -      |
| 3     | EERR     | Erratic error            | The module sets this bit to 1 to report any undefined errors.<br>Due to undefined errors, the USBFS controller enters a suspend state and generate an interrupt via the early suspend bit (ESUSP bit in USBFS_GINTSTS) in the USBFS_GINTSTS register. If the early suspend interrupt is triggered by an undefined error, the application can only perform a soft disconnect to resume communication.                                                      | RO     |
| 2:1   | ENUMSPD  | Enumerated speed         | Indicates the speed at which the USBFS controller is enumerated after detecting the speed via the chirp sequence.<br><ul style="list-style-type: none"> <li>0b00: Reserved</li> <li>0b01: Reserved</li> <li>0b10: Reserved</li> <li>0b11: Full-speed (PHY clock runs at 48MHz).</li> </ul>                                                                                                                                                                | RO     |
| 0     | SUSPSTS  | Suspend status           | In device mode, this bit is set to 1 whenever a suspend state is detected on the USB. When the idle state on the USB bus persists for 3ms, the module enters a suspend state. The module exits the suspend state under the following conditions: <ul style="list-style-type: none"> <li>Activity on the USB data lines.</li> <li>The application writes to the remote wakeup signal bit (RWUSIG bit in USBFS_DCTL) in the USBFS_DCTL register.</li> </ul> | RO     |

### 35.6.5.4 Device IN Endpoint General Interrupt Mask Register (USBFS\_DIEPMSK)

This register is used in conjunction with the USBFS\_DIEPINTx registers of all endpoints to generate interrupts on each IN endpoint. By performing a write operation on the corresponding bit of this register, the IN endpoint interrupt in the USBFS\_DIEPINTx register can be masked. By default, all status interrupts are masked.

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |      |          |    |    |   |   |   |        |        |          |     |          |      |      |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|------|----------|----|----|---|---|---|--------|--------|----------|-----|----------|------|------|
| 0x810  | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13   | 12       | 11 | 10 | 9 | 8 | 7 | 6      | 5      | 4        | 3   | 2        | 1    | 0    |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |      |          |    |    |   |   |   |        |        |          |     |          |      |      |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | NAKM | Reserved |    |    |   |   |   | INPNEM | INPNMM | TTXFEMSK | TOM | Reserved | EPDM | XFRM |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RW   |          |    |    |   |   |   | RW     | RW     | RW       | RW  |          | RW   | RW   |

| Bit   | Name     | Function                                              | Description                                                                                              | Access |
|-------|----------|-------------------------------------------------------|----------------------------------------------------------------------------------------------------------|--------|
| 31:14 | Reserved | Reserved                                              | -                                                                                                        | -      |
| 13    | NAKM     | NAK interrupt mask                                    | <ul style="list-style-type: none"> <li>0b0: Maskable interrupt</li> <li>0b1: Enable interrupt</li> </ul> | RW     |
| 12:7  | Reserved | Reserved                                              | -                                                                                                        | -      |
| 6     | INEPNEM  | IN endpoint NAK active interrupt mask                 | <ul style="list-style-type: none"> <li>0b0: Maskable interrupt</li> <li>0b1: Enable interrupt</li> </ul> | RW     |
| 5     | INEPNMM  | IN token interrupt mask received on EP mismatch       | <ul style="list-style-type: none"> <li>0b0: Maskable interrupt</li> <li>0b1: Enable interrupt</li> </ul> | RW     |
| 4     | TTXFEMSK | IN token interrupt mask received when TxFIFO is empty | <ul style="list-style-type: none"> <li>0b0: Maskable interrupt</li> <li>0b1: Enable interrupt</li> </ul> | RW     |
| 3     | TOM      | Timeout interrupt mask (Non synchronous endpoint)     | <ul style="list-style-type: none"> <li>0b0: Maskable interrupt</li> <li>0b1: Enable interrupt</li> </ul> | RW     |
| 2     | Reserved | Reserved                                              | -                                                                                                        | -      |
| 1     | EPDM     | Endpoint disabled interrupt mask                      | <ul style="list-style-type: none"> <li>0b0: Maskable interrupt</li> <li>0b1: Enable interrupt</li> </ul> | RW     |
| 0     | XFRM     | Transfer completed interrupt mask                     | <ul style="list-style-type: none"> <li>0b0: Maskable interrupt</li> <li>0b1: Enable interrupt</li> </ul> | RW     |



### 35.6.5.5 Device OUT Endpoint General Interrupt Mask Register (USBFS\_DOEPMSK)

This register is used in conjunction with the USBFS\_DOEPINTx registers of all endpoints to generate interrupts on each OUT endpoint. By performing a write operation on the corresponding bit of this register, the OUT endpoint interrupt in the USBFS\_DOEPINTx register can be masked. By default, all status interrupts are masked.

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |        |    |       |    |          |  |      |  |       |  |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|--------|----|-------|----|----------|--|------|--|-------|--|
| 0x814  | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3      | 2  | 1     | 0  |          |  |      |  |       |  |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |        |    |       |    |          |  |      |  |       |  |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   | OTEPDM |    | STUPM |    | Reserved |  | EPDM |  | XFCRM |  |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   | RW     | RW |       | RW | RW       |  |      |  |       |  |

| Bit  | Name     | Function                                                 | Description                                                                                                                                                                                                                                                                           | Access |
|------|----------|----------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:5 | Reserved | Reserved                                                 | -                                                                                                                                                                                                                                                                                     | -      |
| 4    | OTEPDM   | OUT token interrupt mask received when endpoint disabled | <ul style="list-style-type: none"> <li>0b0: Maskable interrupt</li> <li>0b1: Enable interrupt</li> </ul> <div>  <b>Note</b><br/>           Applicable only to control OUT endpoints.         </div> | RW     |
| 3    | STUPM    | SETUP phase done interrupt mask                          | <ul style="list-style-type: none"> <li>0b0: Maskable interrupt</li> <li>0b1: Enable interrupt</li> </ul> <div>  <b>Note</b><br/>           Applicable only to control endpoints.         </div>    | RW     |
| 2    | Reserved | Reserved                                                 | -                                                                                                                                                                                                                                                                                     | -      |
| 1    | EPDM     | Endpoint disabled interrupt mask                         | <ul style="list-style-type: none"> <li>0b0: Maskable interrupt</li> <li>0b1: Enable interrupt</li> </ul>                                                                                                                                                                              | RW     |
| 0    | XFRM     | Transfer completed interrupt mask                        | <ul style="list-style-type: none"> <li>0b0: Maskable interrupt</li> <li>0b1: Enable interrupt</li> </ul>                                                                                                                                                                              | RW     |

### 35.6.5.6 Device All Endpoints Interrupt Register (USBFS\_DAINR)

When a active event occurs on an endpoint, the USBFS\_DAINR register interrupts the application via the device OUT endpoint interrupt bit or device IN endpoint interrupt bit (OEPINT or IEPINT bits in USBFS\_GINTSTS, respectively) in the USBFS\_GINTSTS register. Each endpoint corresponds to one interrupt bit, with a maximum of 16 interrupt bits for both OUT and IN endpoints. Bidirectional endpoints use the corresponding IN and OUT interrupt bits. When the application sets the bit to 1 and clears the bit in the corresponding device endpoint x interrupt register (USBFS\_DIEPINTx/USBFS\_DOEPINTx), the corresponding bit in this register is also set to 1 and cleared.

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |        |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |  |  |  |  |  |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|--------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|--|--|--|--|--|--|
| 0x818  | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15     | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |  |  |  |  |  |  |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |        |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |  |  |  |  |  |
| Name   | OEPINT       |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | IEPINT |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |  |  |  |  |  |
| Access | RW           |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RW     |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |  |  |  |  |  |

| Bit   | Name   | Function                    | Description                                                                                                                                                                                                                                                                                                                            | Access |
|-------|--------|-----------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:16 | OEPINT | OUT endpoint interrupt bits | Corresponding to OUT endpoints 15 to 0:<br>● 0b0: No interrupt has occurred for this endpoint<br>● 0b1: Interrupt has occurred for this endpoint<br><div>  <b>Note</b><br/>           Each bit corresponds to an OUT endpoint.         </div>       | RW     |
| 15:0  | IEPINT | IN endpoint interrupt bits  | Corresponding to IN Endpoint 15 to Endpoint 0:<br>● 0b0: No interrupt has occurred for this endpoint<br>● 0b1: Interrupt has occurred for this endpoint<br><div>  <b>Note</b><br/>           Each bit corresponds to an IN endpoint.         </div> | RW     |

### 35.6.5.7 Device All Endpoints Interrupt Mask Register (USBFS\_DAINTRMSK)

The USBFS\_DAINTRMSK register is used in conjunction with the device endpoint interrupt register to interrupt the application when an event occurs on the device endpoint. However, the corresponding bit in the USBFS\_DAINTR register is still set to 1 for this interrupt.

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |         |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |  |  |  |  |  |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|--|--|--|--|--|--|
| 0x81C  | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15      | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |  |  |  |  |  |  |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |         |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |  |  |  |  |  |
| Name   | OEPINTM      |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | IEPINTM |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |  |  |  |  |  |
| Access | RW           |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RW      |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |  |  |  |  |  |

| Bit   | Name    | Function                         | Description                                                                                                                                                                                                                                                                   | Access |
|-------|---------|----------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:16 | OEPINTM | OUT endpoint interrupt mask bits | Corresponding to OUT endpoints 15 to 0:<br>● 0b0: Maskable interrupt<br>● 0b1: Enable interrupt<br><div>  <b>Note</b><br/>           Each bit corresponds to an OUT endpoint.         </div> | RW     |
| 15:0  | IEPINTM | IN endpoint interrupt mask bits  | Corresponding to IN endpoints 15 to 0:<br>● 0b0: Maskable interrupt<br>● 0b1: Enable interrupt<br><div>  <b>Note</b><br/>           Each bit corresponds to an IN endpoint.         </div> | RW     |

### 35.6.5.8 Device IN Endpoint FIFO Empty Interrupt Mask Register (USBFS\_DIEPMPMSK)

This register is used to control the generation of IN endpoint FIFO empty interrupts.

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |           |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |  |  |  |  |  |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-----------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|--|--|--|--|--|--|
| 0x834  | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15        | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |  |  |  |  |  |  |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |           |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |  |  |  |  |  |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | INEPTXFEM |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |  |  |  |  |  |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RW        |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |  |  |  |  |  |

| Bit   | Name     | Function | Description | Access |
|-------|----------|----------|-------------|--------|
| 31:16 | Reserved | Reserved | -           | -      |

| Bit  | Name      | Function                                | Description                                                                                                                                                                                                                                                                                                                                                                                              | Access |
|------|-----------|-----------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 15:0 | INEPTXFEM | IN EP Tx FIFO empty interrupt mask bits | <p>These bits are used as the mask bits for USBFS_DIEPINTx.</p> <p>Corresponding to IN endpoint 15 to endpoint 0:</p> <ul style="list-style-type: none"><li>● 0b0: Maskable interrupt</li><li>● 0b1: Enable interrupt</li></ul> <div> <b>Note</b><br/>Each bit corresponds to a TXFE interrupt of an IN endpoint.</div> | RW     |

### 35.6.5.9 Device IN Endpoint 0 Control Register (USBFS\_DIEPCTL0)

This register is used to control control transfer endpoint 0.

| Offset | Bit Position |       |          |    |      |      |        |    |    |       |          |       |        |          |        |          |    |    |    |    |    |    |   |   |   |       |   |   |   |   |   |   |
|--------|--------------|-------|----------|----|------|------|--------|----|----|-------|----------|-------|--------|----------|--------|----------|----|----|----|----|----|----|---|---|---|-------|---|---|---|---|---|---|
| 0x900  | 31           | 30    | 29       | 28 | 27   | 26   | 25     | 24 | 23 | 22    | 21       | 20    | 19     | 18       | 17     | 16       | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6     | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00008000   |       |          |    |      |      |        |    |    |       |          |       |        |          |        |          |    |    |    |    |    |    |   |   |   |       |   |   |   |   |   |   |
| Name   | EPENA        | EPDIS | Reserved |    | SNAK | CNAK | TXFNUM |    |    | STALL | Reserved | EPTYP | NAKSTS | Reserved | USBAEP | Reserved |    |    |    |    |    |    |   |   |   | MPSIZ |   |   |   |   |   |   |
| Access | RW           | RW    |          |    | RW   | RW   | RW     |    |    | RW    |          | RO    | RO     |          | RO     |          |    |    |    |    |    |    |   |   |   | RW    |   |   |   |   |   |   |

| Bit   | Name     | Function         | Description                                                                                                                                                                                                                                                                                                                                                                                                                                         | Access |
|-------|----------|------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31    | EPENA    | Endpoint enable  | The application sets this bit to 1 to initiate a data transfer on endpoint 0.<br>The module clears this bit before any of the following interrupts are triggered on this endpoint: <ul style="list-style-type: none"> <li>● SETUP phase done</li> <li>● Endpoint disable</li> <li>● Transmission completion</li> </ul>                                                                                                                              | RW     |
| 30    | EPDIS    | Endpoint disable | The application can set this bit to 1 even before the transfer on this endpoint is completed to stop data transmission on the endpoint. The application must wait until the endpoint disable interrupt occurs before considering the endpoint as disabled. The module clears this bit before the endpoint disable interrupt bit is set to 1. The application can set this bit to 1 only after the endpoint enable bit of this endpoint is set to 1. | RW     |
| 29:28 | Reserved | Reserved         | -                                                                                                                                                                                                                                                                                                                                                                                                                                                   | -      |
| 27    | STALL    | Set NAK          | Writing to this bit sets the NAK bit of the endpoint to 1.<br>Through this bit, the application can control the sending of NAK handshake on the endpoint. The module may also set this bit to 1 when the endpoint receives a SETUP packet.                                                                                                                                                                                                          | RW     |
| 26    | CNAK     | Clear NAK        | Writing to this bit clears the NAK bit of the endpoint.                                                                                                                                                                                                                                                                                                                                                                                             | RW     |
| 25:22 | TXFNUM   | TxFIFO number    | This value is set to the FIFO number allocated to IN endpoint 0. Only TX-FIFO0 can be used.                                                                                                                                                                                                                                                                                                                                                         | RW     |
| 21    | STALL    | STALL handshake  | The application can only set this bit to 1. When the endpoint receives a SETUP token, the module clears this bit. If the NAK bit, global IN NAK, or global OUT NAK and this bit are all set to 1, the STALL bit takes precedence.                                                                                                                                                                                                                   | RW     |
| 20    | Reserved | Reserved         | -                                                                                                                                                                                                                                                                                                                                                                                                                                                   | -      |

| Bit   | Name            | Function            | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                              | Access |
|-------|-----------------|---------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 19:18 | EPTYP           | Endpoint type       | Hardware setting is '0b00', indicating a control-type endpoint.                                                                                                                                                                                                                                                                                                                                                                                                          | RO     |
| 17    | NAKSTS          | NAK status          | <ul style="list-style-type: none"> <li>● 0b0: The module replies with a non-NAK handshake based on FIFO status</li> <li>● 0b1: The module replies with a NAK handshake on this endpoint</li> </ul> <p>When this bit is set to 1 (either by the application or the module), the module stops sending data even if there is still data available in the Tx FIFO. Regardless of how this bit is set, the module always responds to SETUP packets with an ACK handshake.</p> | RO     |
| 16    | <i>Reserved</i> | <i>Reserved</i>     | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | -      |
| 15    | USBAEP          | USB active endpoint | This bit is always set to 1, indicating that control endpoint 0 is always active in all configurations and interfaces.                                                                                                                                                                                                                                                                                                                                                   | RO     |
| 14:2  | <i>Reserved</i> | <i>Reserved</i>     | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | -      |
| 1:0   | MPSIZ           | Maximum packet size | <p>The application must program this field to the maximum packet size of the current logical endpoint.</p> <ul style="list-style-type: none"> <li>● 0b00: 64 bytes</li> <li>● 0b01: 32 bytes</li> <li>● 0b10: 16 bytes</li> <li>● 0b11: 8 bytes</li> </ul>                                                                                                                                                                                                               | RW     |

### 35.6.5.10 Device IN Endpoint x Control Register (USBFS\_DIEPCTLx) (x=1-15)

Applications use this register to control the behavior of each logical endpoint (excluding endpoint 0).

Offset: 0x900+0x20\*x

| Offset | Bit Position |       |         |                |      |      |        |    |    |    |       |          |       |        |            |        |          |    |    |    |       |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|-------|---------|----------------|------|------|--------|----|----|----|-------|----------|-------|--------|------------|--------|----------|----|----|----|-------|----|---|---|---|---|---|---|---|---|---|---|
| -      | 31           | 30    | 29      | 28             | 27   | 26   | 25     | 24 | 23 | 22 | 21    | 20       | 19    | 18     | 17         | 16     | 15       | 14 | 13 | 12 | 11    | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |       |         |                |      |      |        |    |    |    |       |          |       |        |            |        |          |    |    |    |       |    |   |   |   |   |   |   |   |   |   |   |
| Name   | EPENA        | EPDIS | SODDFRM | SD0PID/SEVNFRM | SNAK | CNAK | TXFNUM |    |    |    | STALL | Reserved | EPTYP | NAKSTS | EONUM/DPID | USBAEP | Reserved |    |    |    | MPSIZ |    |   |   |   |   |   |   |   |   |   |   |
| Access | RW           | RW    | RW      | RW             | RW   | RW   | RW     |    |    |    | RW    |          | RO    | RO     | RO         | RO     |          |    |    |    | RW    |    |   |   |   |   |   |   |   |   |   |   |

| Bit | Name           | Function                     | Description                                                                                                                                                                                                                                                                                                                                                                                                                                         | Access |
|-----|----------------|------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31  | EPENA          | Endpoint enable              | The application sets this bit to 1 to initiate a data transfer on endpoint. The module clears this bit before any of the following interrupts are triggered on this endpoint: <ul style="list-style-type: none"> <li>● Endpoint disabled</li> <li>● Transmission completion</li> </ul>                                                                                                                                                              | RW     |
| 30  | EPDIS          | Endpoint disable             | The application can set this bit to 1 even before the transfer on this endpoint is completed to stop data transmission on the endpoint. The application must wait until the endpoint disable interrupt occurs before considering the endpoint as disabled. The module clears this bit before the endpoint disable interrupt bit is set to 1. The application can set this bit to 1 only after the endpoint enable bit of this endpoint is set to 1. | RW     |
| 29  | SODDFRM        | Set odd frame                | Applies only to synchronous IN and OUT endpoints.<br>Writing to this field sets the even/odd frame (EONUM) field to odd frame.                                                                                                                                                                                                                                                                                                                      | RW     |
| 28  | SD0PID/SEVNFRM | Set DATA0 PID/Set even frame | <b>SD0PID:</b> Applicable only to interrupt/bulk IN endpoints. Writing to this field sets the endpoint data PID (DPID) field in this register to DATA0.<br><b>SEVNFRM:</b> Applicable only to synchronous IN endpoints. Writing to this field sets the even/odd frame (EONUM) field to even frame.                                                                                                                                                  | RW     |
| 27  | SNAK           | Set NAK                      | Writing to this bit sets the NAK bit of the endpoint to 1.<br>Through this bit, the application can control the sending of NAK handshake on the endpoint. The module can also set this bit of the IN endpoint to 1 when a transmission completed interrupt occurs or after a SETUP is received on the endpoint.                                                                                                                                     | RW     |

| Bit   | Name            | Function                         | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | Access |
|-------|-----------------|----------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 26    | CNAK            | Clear NAK                        | Writing to this bit clears the NAK bit of the end-point.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | RW     |
| 25:22 | TXFNUM          | TxFIFO number                    | These bits are used to specify the FIFO number associated with this endpoint. A separate FIFO number must be assigned for each active IN end-point.<br>This field is only active for IN endpoints.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | RW     |
| 21    | STALL           | STALL hand-shake                 | The application sets this bit to 1, causing the device to respond with STALL to all tokens from the USB host. If the NAK bit, global IN NAK, or global OUT NAK is set to 1 at the same time as this bit, the STALL bit takes precedence. Only the application program can clear this bit, not the module.                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | RW     |
| 20    | <i>Reserved</i> | <i>Reserved</i>                  | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | -      |
| 19:18 | EPTYP           | Endpoint type                    | <ul style="list-style-type: none"> <li>● 0b00: Control</li> <li>● 0b01: Isochronous</li> <li>● 0b10: Bulk</li> <li>● 0b11: Interrupt</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | RO     |
| 17    | NAKSTS          | NAK status                       | <ul style="list-style-type: none"> <li>● 0b0: The module replies with a non-NAK handshake based on FIFO status</li> <li>● 0b1: The module replies with a NAK handshake on this endpoint</li> </ul> <p>When the application or module sets this bit to 1:</p> <ul style="list-style-type: none"> <li>● For non-synchronous IN endpoints: The module will stop transmitting any data through the IN endpoint even if there is available data in the TxFIFO.</li> <li>● For synchronous IN endpoints: The module transmits a zero-length data packet even when there is available data in the TxFIFO.</li> </ul> <p>Regardless of how this bit is set, the module always responds to SETUP packets with an ACK handshake.</p>                                                     | RO     |
| 16    | EONUM/<br>DPID  | Even/Odd frame/Endpoint data PID | <p><b>EONUM:</b><br/>Applies only to synchronous IN endpoints. Indicates the frame number where the module sends/receives synchronous data for this endpoint. The application must program the even/odd frame numbers through the SEVNFRM and SODDFRM fields in this register to enable this endpoint to transmit/receive synchronous data.</p> <ul style="list-style-type: none"> <li>● 0b0: Even frame</li> <li>● 0b1: Odd frame</li> </ul> <p><b>DPID:</b><br/>Applies to interrupt/bulk IN endpoints only. Contains the PID of the data packet to be received or transmitted on this endpoint. After the endpoint is activated, the application must program the PID of the first data packet to be received or transmitted on this endpoint. The application programs</p> | RO     |



| Bit   | Name     | Function            | Description                                                                                                                                                                                                                                                                                                                              | Access |
|-------|----------|---------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
|       |          |                     | the DATA0 or DATA1 PID using the SD0PID register field.<br><ul style="list-style-type: none"> <li>● 0b0: DATA0</li> <li>● 0b1: DATA1</li> </ul>                                                                                                                                                                                          |        |
| 15    | USBAEP   | USB active endpoint | Indicates whether this endpoint is active in the current configuration and interface. Upon detecting a USB reset, the module clears this bit for all endpoints (except endpoint 0). Upon receiving the SetConfiguration and SetInterface commands, the application must program the endpoint register accordingly and set this bit to 1. | RO     |
| 14:11 | Reserved | Reserved            | -                                                                                                                                                                                                                                                                                                                                        | -      |
| 10:0  | MPSIZ    | Maximum packet size | The application must program this field to the maximum packet size of the current logical endpoint. This value is in bytes.                                                                                                                                                                                                              | RW     |

### 35.6.5.11 Device IN Endpoint x Interrupt Register (USBFS\_DIEPINTx) (x=0-15)

This register indicates the status of the endpoint when USB and AHB related events occur. When the IN endpoint interrupt bit (IEPINT bit in USBFS\_GINTSTS) in the module interrupt register is set to 1, the application must read this register. Before the application can read this register, it must first read the device all endpoints interrupt (USBFS\_DAINTE) register to obtain the correct endpoint number of the device endpoint x interrupt register. The application must clear the corresponding bit in this register in order to clear the corresponding bits in the USBFS\_DAINTE and USBFS\_GINTSTS registers.

Offset: 0x908+0x20\*x

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |      |        |          |       |     |          |        |      |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|------|--------|----------|-------|-----|----------|--------|------|
| '      | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7    | 6      | 5        | 4     | 3   | 2        | 1      | 0    |
| Reset  | 0x00000080   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |      |        |          |       |     |          |        |      |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   | TXFE | INEPNE | Reserved | TTXFE | TOC | Reserved | EPDISD | XFRC |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   | RO   | RW     |          | RW    | RW  |          | RW     | RW   |

| Bit  | Name     | Function                               | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | Access |
|------|----------|----------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:8 | Reserved | Reserved                               | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | -      |
| 7    | TXFE     | Transmit FIFO empty                    | This interrupt is set when the TxFIFO of this endpoint is in half empty or full empty status. Whether the TxFIFO is in half empty or full empty status is determined by the TxFIFO empty level bit (TXFELVL bit in USBFS_GAHBCFG) in the USBFS_GAHBCFG register.                                                                                                                                                                                                                                                                                                                    | RO     |
| 6    | INEPNE   | IN endpoint NAK effective              | This bit can be cleared when the application clears the IN endpoint NAK by writing to the CNAK bit in USBFS_DIEPCTLx.<br>This interrupt indicates that the module has sampled the NAK set to 1 (by the application or the module), and the result has taken effect. This interrupt indicates that the IN endpoint NAK bit set to 1 by the application has taken effect in the module.<br>This interrupt does not guarantee that a NAK handshake signal was sent on the USB.<br>The STALL bit has priority over the NAK bit.<br>Writing 1 to this bit by software can also clear it. | RW     |
| 5    | Reserved | Reserved                               | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | -      |
| 4    | TTXFE    | IN token received when TxFIFO is empty | Applies only to non-periodic IN endpoints. When the TxFIFO (periodic/non-periodic) corresponding to this endpoint is empty and an IN token is received, an interrupt is generated. Write 1 to clear by software.                                                                                                                                                                                                                                                                                                                                                                    | RW     |

| Bit | Name            | Function                     | Description                                                                                                                                                             | Access |
|-----|-----------------|------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 3   | TOC             | Timeout condition            | Applicable only to control IN endpoints. Indicates that this endpoint has timed out in responding to the most recently received IN token. Write 1 to clear by software. | RW     |
| 2   | <i>Reserved</i> | <i>Reserved</i>              | -                                                                                                                                                                       | -      |
| 1   | EPDISD          | Endpoint disabled interrupt  | This bit indicates that the endpoint has been disabled by the application. Write 1 to clear by software.                                                                | RW     |
| 0   | XFRC            | Transfer completed interrupt | This field indicates that the transmission set up on this endpoint has been completed on both the USB and AHB. Write 1 to clear by software.                            | RW     |

### 35.6.5.12 Device IN Endpoint 0 Transfer Size Register (USBFS\_DIEPTSIZ0)

The application must modify this register before enabling endpoint 0. After enabling endpoint 0 by setting the endpoint enable bit (EPENA in USBFS\_DIEPCTL0) in the device control endpoint 0 control register, the module modifies this register. The application can only read this register after the module clears the endpoint enable bit.

Non-zero endpoints use registers for endpoints 1-15.

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |        |          |    |    |    |    |    |    |    |    |   |   |   |        |   |   |   |   |   |   |  |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|--------|----------|----|----|----|----|----|----|----|----|---|---|---|--------|---|---|---|---|---|---|--|
| 0x910  | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19     | 18       | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6      | 5 | 4 | 3 | 2 | 1 | 0 |  |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |        |          |    |    |    |    |    |    |    |    |   |   |   |        |   |   |   |   |   |   |  |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    | PKTCNT | Reserved |    |    |    |    |    |    |    |    |   |   |   | XFRSIZ |   |   |   |   |   |   |  |
| Access |              |    |    |    |    |    |    |    |    |    |    |    | RW     |          |    |    |    |    |    |    |    |    |   |   |   | RW     |   |   |   |   |   |   |  |

| Bit   | Name     | Function      | Description                                                                                                                                                                                                                                                                                                                                                                                         | Access |
|-------|----------|---------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:21 | Reserved | Reserved      | -                                                                                                                                                                                                                                                                                                                                                                                                   | -      |
| 20:19 | PKTCNT   | Packet count  | Indicates the number of data packets included in one data transfer for endpoint 0.<br>This field decrements each time a data packet (maximum size or short packet) is read from the TxFIFO.                                                                                                                                                                                                         | RW     |
| 18:7  | Reserved | Reserved      | -                                                                                                                                                                                                                                                                                                                                                                                                   | -      |
| 6:0   | XFRSIZ   | Transfer size | Indicates the amount of data included in one data transfer for endpoint 0, in bytes. The module only interrupts the application after these data have been transmitted. The transfer size can be set to the maximum packet size of the endpoint to interrupt at the end of each packet.<br>Each time a data packet from external memory is written to the TxFIFO, the module decrements this field. | RW     |

### 35.6.5.13 Device IN Endpoint x Transfer Size Register (USBFS\_DIEPTSIZx) (x=1-15)

The application must modify this register before enabling the endpoint. After the endpoint is enabled via the endpoint enable bit (EPENA bit in USBFS\_DIEPCTLx) in the USBFS\_DIEPCTLx register, the module modifies this register. The application can only read this register after the module clears the endpoint enable bit.

Offset: 0x910+0x20\*x

| Offset | Bit Position |      |    |        |    |    |    |    |    |    |    |    |    |        |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|------|----|--------|----|----|----|----|----|----|----|----|----|--------|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| -      | 31           | 30   | 29 | 28     | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18     | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |      |    |        |    |    |    |    |    |    |    |    |    |        |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | Reserved     | MCNT |    | PKTCNT |    |    |    |    |    |    |    |    |    | XFRSIZ |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access |              | RW   |    | RW     |    |    |    |    |    |    |    |    |    | RW     |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit   | Name     | Function      | Description                                                                                                                                                                                                                                                                                                                                                                                                            | Access |
|-------|----------|---------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31    | Reserved | Reserved      | -                                                                                                                                                                                                                                                                                                                                                                                                                      | -      |
| 30:29 | MCNT     | Multi count   | For periodic IN endpoints, this field indicates the number of packets that must be sent per frame on the USB. The module uses this field to calculate the data PID for synchronous IN endpoints. <ul style="list-style-type: none"> <li>● 0b00: Setting prohibited</li> <li>● 0b01: 1 packet</li> <li>● 0b10: 2 packets</li> <li>● 0b11: 3 packets</li> </ul>                                                          | RW     |
| 28:19 | PKTCNT   | Packet count  | Indicates the number of data packets included in one data transfer on this endpoint. This field decrements each time a data packet (maximum size or short packet) is read from the TxFIFO.                                                                                                                                                                                                                             | RW     |
| 18:0  | XFRSIZ   | Transfer size | This field contains the amount of data included in one data transfer of the current endpoint, in bytes. The module only interrupts the application after these data have been transmitted. The transfer size can be set to the maximum packet size of the endpoint to interrupt at the end of each packet.<br>Each time a data packet from external memory is written to the TxFIFO, the module decrements this field. | RW     |

35.6.5.14 Device IN Endpoint x DMA Address Register (USBFS\_DIEPDMAx) (x=0-15)

This register is used to set the DMA address when the device endpoint is in DMA mode.

Offset: 0x914+0x20\*x

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| -      | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0xFFFFFFFF   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | DMAADDR      |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access | RW           |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit  | Name    | Function    | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | Access |
|------|---------|-------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:0 | DMAADDR | DMA address | <div>This bit contains the starting address of the external storage area when data is stored on the endpoint using DMA.</div> <div> <b>Notice</b><br/>For control endpoints, the storage area pointed to by this field is also used to store control OUT data packets and SETUP transaction data packets. When three or more SETUP data packets are received consecutively, the SETUP data packets in the memory are overwritten. This register increments each time an AHB transfer is performed. The application must set a double-word aligned address.</div> | RW     |

## 35.6.5.15 Device IN Endpoint x Transmit FIFO Status Register (USBFS\_DTXFSTSx) (x=0-15)

Offset: 0x918+0x20\*x

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |           |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-----------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| -      | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15        | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000280   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |           |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | INEPTFSAV |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RO        |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit   | Name      | Function                           | Description                                                                                                                                                                                                                                                                                                                                           | Access |
|-------|-----------|------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:16 | Reserved  | Reserved                           | -                                                                                                                                                                                                                                                                                                                                                     | -      |
| 15:0  | INEPTFSAV | IN endpoint TxFIFO space available | Indicates the available free space size in the end-point TxFIFO. In 32-bit words: <ul style="list-style-type: none"> <li>● 0: The endpoint TxFIFO is full.</li> <li>● 1: 1 word available.</li> <li>● 2: 2 words available.</li> <li>● .....</li> <li>● n: n words are available (n is the value of the register USBFS_DIEPTFXx.INEPTXFD).</li> </ul> | RO     |

## 35.6.5.16 Device Control OUT Endpoint 0 Control Register (USBFS\_DOEPCTL0)

This register is used to control control transfer endpoint 0.

| Offset | Bit Position |       |          |    |      |      |          |    |    |       |      |       |        |          |        |          |
|--------|--------------|-------|----------|----|------|------|----------|----|----|-------|------|-------|--------|----------|--------|----------|
| 0xB00  | 31           | 30    | 29       | 28 | 27   | 26   | 25       | 24 | 23 | 22    | 21   | 20    | 19     | 18       | 17     | 16       |
| Reset  | 0x00008000   |       |          |    |      |      |          |    |    |       |      |       |        |          |        |          |
| Name   | EPENA        | EPDIS | Reserved |    | SNAK | CNAK | Reserved |    |    | STALL | SNPM | EPTYP | NAKSTS | Reserved | USBAEP | Reserved |
| Access | RW           | RW    |          |    | RW   | RW   |          |    |    | RW    | RW   | RO    | RO     |          | RO     |          |

| Bit   | Name     | Function         | Description                                                                                                                                                                                                                                                                                                            | Access |
|-------|----------|------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31    | EPENA    | Endpoint enable  | The application sets this bit to 1 to initiate a data transfer on endpoint 0.<br>The module clears this bit before any of the following interrupts are triggered on this endpoint: <ul style="list-style-type: none"> <li>● SETUP phase done</li> <li>● Endpoint disable</li> <li>● Transmission completion</li> </ul> | RW     |
| 30    | EPDIS    | Endpoint disable | The application cannot disable control OUT endpoint 0.                                                                                                                                                                                                                                                                 | RW     |
| 29:28 | Reserved | Reserved         | -                                                                                                                                                                                                                                                                                                                      | -      |
| 27    | SNAK     | Set NAK          | Writing to this bit sets the NAK bit of the endpoint to 1.<br>Through this bit, the application can control the sending of NAK handshake on the endpoint. The module may also set this bit to 1 when the endpoint receives a SETUP packet.                                                                             | RW     |
| 26    | CNAK     | Clear NAK        | Writing to this bit clears the NAK bit of the endpoint.                                                                                                                                                                                                                                                                | RW     |
| 25:22 | Reserved | Reserved         | -                                                                                                                                                                                                                                                                                                                      | -      |
| 21    | STALL    | STALL handshake  | The application can only set this bit to 1 when the endpoint receives a SETUP token, and the module clears it. If the NAK bit, global OUT NAK, and this bit are all set to 1, the STALL bit takes precedence. Regardless of how this bit is set, the module always responds to SETUP packets with an ACK handshake.    | RW     |
| 20    | SNPM     | Snoop mode       | This bit is used to configure the endpoint in snoop mode. In snoop mode, the module does not check whether the OUT packet is correct before transferring it to the application storage area.                                                                                                                           | RW     |
| 19:18 | EPTYP    | Endpoint type    | Hardware setting is '0b00', indicating a control-type endpoint.                                                                                                                                                                                                                                                        | RO     |



| Bit  | Name            | Function            | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | Access |
|------|-----------------|---------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 17   | NAKSTS          | NAK status          | <ul style="list-style-type: none"> <li>● 0b0: The module replies with a non-NAK handshake based on FIFO status</li> <li>● 0b1: The module replies with a NAK handshake on this endpoint</li> </ul> <p>When the application or module sets this bit to 1, the module stops receiving data even if there is space in the RxFIFO to accommodate the received packet. Regardless of how this bit is set, the module always responds to SETUP packets with an ACK handshake.</p> | RO     |
| 16   | <i>Reserved</i> | <i>Reserved</i>     | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | -      |
| 15   | USBAEP          | USB active endpoint | This bit is always set to 1, indicating that control endpoint 0 is always active in all configurations and interfaces.                                                                                                                                                                                                                                                                                                                                                      | RO     |
| 14:2 | <i>Reserved</i> | <i>Reserved</i>     | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | -      |
| 1:0  | MPSIZ           | Maximum packet size | <p>The maximum packet size of control OUT endpoint 0 is the same as the value programmed in control IN endpoint 0.</p> <ul style="list-style-type: none"> <li>● 0b00: 64 bytes</li> <li>● 0b01: 32 bytes</li> <li>● 0b10: 16 bytes</li> <li>● 0b11: 8 bytes</li> </ul>                                                                                                                                                                                                      | RW     |

### 35.6.5.17 Device OUT Endpoint x Control Register (USBFS\_DOEPCTLx) (x=1-15)

Applications use this register to control the behavior of each logical endpoint (excluding endpoint 0).

Offset: 0xB00+0x20\*x

| Offset | Bit Position |       |                |                |      |      |          |    |    |    |    |       |      |       |        |            |    |        |          |    |    |    |   |       |   |   |   |   |   |   |   |   |  |
|--------|--------------|-------|----------------|----------------|------|------|----------|----|----|----|----|-------|------|-------|--------|------------|----|--------|----------|----|----|----|---|-------|---|---|---|---|---|---|---|---|--|
| -      | 31           | 30    | 29             | 28             | 27   | 26   | 25       | 24 | 23 | 22 | 21 | 20    | 19   | 18    | 17     | 16         | 15 | 14     | 13       | 12 | 11 | 10 | 9 | 8     | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |  |
| Reset  | 0x00000000   |       |                |                |      |      |          |    |    |    |    |       |      |       |        |            |    |        |          |    |    |    |   |       |   |   |   |   |   |   |   |   |  |
| Name   | EPENA        | EPDIS | SDIPID/SODDFRM | SDOPID/SEVNFRM | SNAK | CNAK | Reserved |    |    |    |    | STALL | SNPM | EPTYP | NAKSTS | EONUM/DPID |    | USBAEP | Reserved |    |    |    |   | MPSIZ |   |   |   |   |   |   |   |   |  |
| Access | RW           | RW    | RW             | RW             | RW   | RW   |          |    |    |    |    | RW    | RW   | RO    | RO     | RO         | RO |        |          |    |    |    |   | RW    |   |   |   |   |   |   |   |   |  |

| Bit | Name           | Function                     | Description                                                                                                                                                                                                                                                                                                                                                                                                                                           | Access |
|-----|----------------|------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31  | EPENA          | Endpoint enable              | The application sets this bit to 1 to initiate a data transfer on endpoint 0.<br>The module clears this bit before any of the following interrupts are triggered on this endpoint: <ul style="list-style-type: none"> <li>● Endpoint disable</li> <li>● Transmission completion</li> </ul>                                                                                                                                                            | RW     |
| 30  | EPDIS          | Endpoint disable             | An application can set this bit to 1 to stop data sending/receiving on an endpoint even before the transfer is complete on that endpoint. The application must wait until the endpoint disable interrupt occurs before considering the endpoint as disabled. The module clears this bit before the endpoint disable interrupt bit is set to 1. The application can set this bit to 1 only after the endpoint enable bit of this endpoint is set to 1. | RW     |
| 29  | SD1PID/SODDFRM | Set DATA1 PID/set odd frame  | <b>SD1PID:</b> Applicable only to interrupt/bulk OUT endpoints. Writing to this field sets the endpoint data PID (DPID) field in this register to DATA1.<br><b>SODDFRM:</b> Applicable only to synchronous OUT endpoints. Writing to this field sets the even/odd frame (EONUM) field to odd frame.                                                                                                                                                   | RW     |
| 28  | SD0PID/SEVNFRM | Set DATA0 PID/Set even frame | <b>SD0PID:</b> Applicable only to interrupt/bulk OUT endpoints. Writing to this field sets the endpoint data PID (DPID) field in this register to DATA0.<br><b>SEVNFRM:</b> Applicable only to synchronous OUT endpoints. Writing to this field sets the even/odd frame (EONUM) field to even frame.                                                                                                                                                  | RW     |

| Bit   | Name            | Function                         | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | Access |
|-------|-----------------|----------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 27    | SNAK            | Set NAK                          | Writing to this bit sets the NAK bit of the endpoint to 1.<br>Through this bit, the application can control the sending of NAK handshake on the endpoint. The module can also set this bit of the OUT endpoint to 1 when a transmission completed interrupt occurs or after a SETUP is received on the endpoint.                                                                                                                                                                                                                                                                                                                                                               | RW     |
| 26    | CNAK            | Clear NAK                        | Writing to this bit clears the NAK bit of the endpoint.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | RW     |
| 25:22 | <i>Reserved</i> | <i>Reserved</i>                  | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | -      |
| 21    | STALL           | STALL handshake                  | The application can only set this bit to 1 when the endpoint receives a SETUP token, and the module clears it. If the NAK bit, global OUT NAK, and this bit are all set to 1, the STALL bit takes precedence. Only the application program can clear this bit, not the module.                                                                                                                                                                                                                                                                                                                                                                                                 | RW     |
| 20    | SNPM            | Snoop mode                       | This bit is used to configure the endpoint in snoop mode. In snoop mode, the module no longer checks the correctness of the receive data.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | RW     |
| 19:18 | EPTYP           | Endpoint type                    | <ul style="list-style-type: none"> <li>● 0b00: Control</li> <li>● 0b01: Isochronous</li> <li>● 0b10: Bulk</li> <li>● 0b11: Interrupt</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | RO     |
| 17    | NAKSTS          | NAK status                       | <ul style="list-style-type: none"> <li>● 0b0: The module replies with a non-NAK handshake based on FIFO status</li> <li>● 0b1: The module replies with a NAK handshake on this endpoint</li> </ul> <p>When the application or module sets this bit to 1, the module will stop receiving any data on the OUT endpoint even if there is space available in the RxFIFO to accommodate the incoming packet. Regardless of how this bit is set, the module always responds to SETUP packets with an ACK handshake.</p>                                                                                                                                                              | RO     |
| 16    | EONUM/<br>DPID  | Even/Odd frame/Endpoint data PID | <p><b>EONUM:</b><br/>Applies only to synchronous OUT endpoints. Indicates the frame number where the module sends/receives synchronous data for this endpoint. The application must program the even/odd frame numbers through the SEVNFRM and SODDFRM fields in this register to enable this endpoint to transmit/receive synchronous data.</p> <ul style="list-style-type: none"> <li>● 0b0: Even frame</li> <li>● 0b1: Odd frame</li> </ul> <p><b>DPID:</b><br/>Applies to interrupt/bulk OUT endpoints only. Contains the PID of the data packet to be received or transmitted on this endpoint. After the endpoint is activated, the application must program the PID</p> | RO     |

| Bit   | Name     | Function            | Description                                                                                                                                                                                                                                                                                                                              | Access |
|-------|----------|---------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
|       |          |                     | of the first data packet to be received or transmitted on this endpoint. The application programs the DATA0 or DATA1 PID using the SD0PID register field.<br><ul style="list-style-type: none"> <li>● 0b0: DATA0</li> <li>● 0b1: DATA1</li> </ul>                                                                                        |        |
| 15    | USBAEP   | USB active endpoint | Indicates whether this endpoint is active in the current configuration and interface. Upon detecting a USB reset, the module clears this bit for all endpoints (except endpoint 0). Upon receiving the SetConfiguration and SetInterface commands, the application must program the endpoint register accordingly and set this bit to 1. | RO     |
| 14:11 | Reserved | Reserved            | -                                                                                                                                                                                                                                                                                                                                        | -      |
| 10:0  | MPSIZ    | Maximum packet size | The application must program this field to the maximum packet size of the current logical endpoint. This value is in bytes.                                                                                                                                                                                                              | RW     |

### 35.6.5.18 Device OUT Endpoint x Interrupt Register (USBFS\_DOEPINTx) (x=0-15)

This register indicates the status of the endpoint when USB and AHB related events occur. When the OUT endpoint interrupt bit (OEPINT bit in USBFS\_GINTSTS) in the USBFS\_GINTSTS register is set to 1, the application must read this register. Before the application can read this register, it must first read the USBFS\_DAINTE register to obtain the correct endpoint number of the USBFS\_DOEPINTx register. The application must clear the corresponding bit in this register in order to clear the corresponding bits in the USBFS\_DAINTE and USBFS\_GINTSTS registers.

Offset: 0xB08+0x20\*x

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |         |          |         |      |          |        |      |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---------|----------|---------|------|----------|--------|------|---|
| '      | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7       | 6        | 5       | 4    | 3        | 2      | 1    | 0 |
| Reset  | 0x00000080   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |         |          |         |      |          |        |      |   |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   | B2BSTUP | Reserved | OTEPDIS | STUP | Reserved | EPDISD | XFRC |   |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   | RW      |          | RW      | RW   |          | RW     | RW   |   |

| Bit  | Name     | Function                                  | Description                                                                                                                                                                                                                                                                                                             | Access |
|------|----------|-------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:7 | Reserved | Reserved                                  | -                                                                                                                                                                                                                                                                                                                       | -      |
| 6    | B2BSTUP  | Received consecutive SETUP packet         | Applicable only to control OUT endpoints. This bit indicates that the endpoint has received more than three consecutive SETUP data packets. Writing 1 to this bit by software can also clear it.                                                                                                                        | RW     |
| 5    | Reserved | Reserved                                  | -                                                                                                                                                                                                                                                                                                                       | -      |
| 4    | OTEPDIS  | OUT token received when endpoint disabled | Applicable only to control OUT endpoints. Indicates that an OUT token was received before the endpoint was enabled, thus generating an interrupt. Write 1 to clear by software.                                                                                                                                         | RW     |
| 3    | STUP     | SETUP phase done                          | Applicable only to control OUT endpoints. Indicates that the SETUP phase of the control endpoint has been completed and no more consecutive SETUP data packets is received in the current control transfer. On this interrupt, the application can decode the received SETUP data packet. Write 1 to clear by software. | RW     |
| 2    | Reserved | Reserved                                  | -                                                                                                                                                                                                                                                                                                                       | -      |
| 1    | EPDISD   | Endpoint disabled interrupt               | This bit indicates that the endpoint has been disabled by the application. Write 1 to clear by software.                                                                                                                                                                                                                | RW     |
| 0    | XFRC     | Transfer completed interrupt              | This field indicates that the transmission set up on this endpoint has been completed on both the USB and AHB. Write 1 to clear by software.                                                                                                                                                                            | RW     |

### 35.6.5.19 Device OUT Endpoint 0 Transfer Size Register (USBFS\_DOEPTSIZ0)

The application must modify this register before enabling endpoint 0. After enabling endpoint 0 by setting the endpoint enable bit (EPENA in USBFS\_DIEPCTL0) in the device control endpoint 0 control register, the module modifies this register. The application can only read this register after the module clears the endpoint enable bit.

Non-zero endpoints use registers for endpoints 1-15.

| Offset | Bit Position |         |          |    |    |    |    |    |    |    |    |    |        |          |    |    |    |    |    |    |    |    |   |        |   |   |   |   |   |   |   |   |  |
|--------|--------------|---------|----------|----|----|----|----|----|----|----|----|----|--------|----------|----|----|----|----|----|----|----|----|---|--------|---|---|---|---|---|---|---|---|--|
| 0xB10  | 31           | 30      | 29       | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19     | 18       | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8      | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |  |
| Reset  | 0x00000000   |         |          |    |    |    |    |    |    |    |    |    |        |          |    |    |    |    |    |    |    |    |   |        |   |   |   |   |   |   |   |   |  |
| Name   | Reserved     | STUPCNT | Reserved |    |    |    |    |    |    |    |    |    | PKTCNT | Reserved |    |    |    |    |    |    |    |    |   | XFRSIZ |   |   |   |   |   |   |   |   |  |
| Access |              | RW      |          |    |    |    |    |    |    |    |    |    | RW     |          |    |    |    |    |    |    |    |    |   | RW     |   |   |   |   |   |   |   |   |  |

| Bit   | Name     | Function           | Description                                                                                                                                                                                                                                                                                                                                                                                             | Access |
|-------|----------|--------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31    | Reserved | Reserved           | -                                                                                                                                                                                                                                                                                                                                                                                                       | -      |
| 30:29 | STUPCNT  | SETUP packet count | This field specifies the number of SETUP data packets the endpoint can receive consecutively. <ul style="list-style-type: none"> <li>● 0b00: Setting prohibited</li> <li>● 0b01: 1 packet</li> <li>● 0b10: 2 packets</li> <li>● 0b11: 3 packets</li> </ul>                                                                                                                                              | RW     |
| 28:20 | Reserved | Reserved           | -                                                                                                                                                                                                                                                                                                                                                                                                       | -      |
| 19    | PKTCNT   | Packet count       | The number of data packets that should be received in one transfer.<br>Before enabling the endpoint, software sets this bit; after the transfer starts, this field value automatically decrements each time a data packet is received.                                                                                                                                                                  | RW     |
| 18:7  | Reserved | Reserved           | -                                                                                                                                                                                                                                                                                                                                                                                                       | -      |
| 6:0   | XFRSIZ   | Transfer size      | Indicates the amount of data included in one data transfer for endpoint 0, in bytes. The module only interrupts the application after these data have been transmitted. The transfer size can be set to the maximum packet size of the endpoint to interrupt at the end of each packet.<br>Each time a packet is read from the RxFIFO and written to external memory, the module decrements this field. | RW     |

### 35.6.5.20 Device OUT Endpoint x Transfer Size Register (USBFS\_DOEPTSIZE<sub>x</sub>) (x=1-15)

The application must modify this register before enabling the endpoint. After enabling the endpoint through the endpoint enable bit (EPENA bit in USBFS\_DOEPTCTL<sub>x</sub>) in the USBFS\_DOEPTCTL<sub>x</sub> register, the module modifies this register. The application can only read this register after the module clears the endpoint enable bit.

Offset: 0xB10+0x20\*x

| Offset | Bit Position |                |    |        |    |    |    |    |    |    |    |    |    |    |        |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----------------|----|--------|----|----|----|----|----|----|----|----|----|----|--------|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| -      | 31           | 30             | 29 | 28     | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17     | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |                |    |        |    |    |    |    |    |    |    |    |    |    |        |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | Reserved     | RXDPID/STUPCNT |    | PKTCNT |    |    |    |    |    |    |    |    |    |    | XFRSIZ |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access |              | RW             | RW |        |    |    |    |    |    |    |    |    |    | RW |        |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit   | Name               | Function                                   | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | Access |
|-------|--------------------|--------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31    | Reserved           | Reserved                                   | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | -      |
| 30:29 | RXDPID/<br>STUPCNT | Received data<br>PID/SETUP<br>packet count | <b>RXDPID:</b> Only applicable to synchronous OUT endpoints. This is the PID of the last packet received by this endpoint. <ul style="list-style-type: none"> <li>● 0b00: DATA0</li> <li>● 0b01: DATA2</li> <li>● 0b10: DATA1</li> <li>● 0b11: MDATA</li> </ul> <b>STUPCNT:</b> Only applicable to control OUT endpoints. This field specifies the number of SETUP data packets the endpoint can receive consecutively. <ul style="list-style-type: none"> <li>● 0b00: Setting prohibited</li> <li>● 0b01: 1 packet</li> <li>● 0b10: 2 packets</li> <li>● 0b11: 3 packets</li> </ul> | RW     |
| 28:19 | PKTCNT             | Packet count                               | Indicates the number of data packets included in one data transfer on this endpoint. This field decrements each time a packet (maximum size or short packet) is written to the RxFIFO.                                                                                                                                                                                                                                                                                                                                                                                               | RW     |

| Bit  | Name   | Function      | Description                                                                                                                                                                                                                                                                                                                                                                                                                           | Access |
|------|--------|---------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 18:0 | XFRSIZ | Transfer size | <p>This field contains the amount of data included in one data transfer of the current endpoint, in bytes. The module only interrupts the application after these data have been transmitted. The transfer size can be set to the maximum packet size of the endpoint to interrupt at the end of each packet.</p> <p>Each time a packet is read from the RxFIFO and written to external memory, the module decrements this field.</p> | RW     |



### 35.6.5.21 Device OUT Endpoint x DMA Address Register (USBFS\_DOEPDMAx) (x=0-15)

This register is used to set the DMA address when the device endpoint is in DMA mode.

Offset: 0xB14+0x20\*x

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| -      | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0xFFFFFFFF   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | DMAADDR      |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access | RW           |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit  | Name    | Function    | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | Access |
|------|---------|-------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:0 | DMAADDR | DMA address | <p>This bit contains the starting address of the external storage area when data is sent on the endpoint using DMA.</p> <div>  <b>Notice</b><br/>           For control endpoints, the storage area pointed to by this field is also used to store control OUT data packets and SETUP transaction data packets. When three or more SETUP data packets are received consecutively, the SETUP data packets in the memory are overwritten. This register increments each time an AHB transfer is performed. The application must set a double-word aligned address.         </div> | RW     |

### 35.6.6 USBFS Clock Gating Control Register

Control HCLK and PHY clock by gating the clock control register to reduce power consumption.

#### 35.6.6.1 Clock Gating Control Register (USBFS\_GCCTL)

This register is available in both host mode and device mode.

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |      |   |          |   |         |   |          |   |          |  |         |  |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|------|---|----------|---|---------|---|----------|---|----------|--|---------|--|
| 0xE00  | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7    | 6 | 5        | 4 | 3       | 2 | 1        | 0 |          |  |         |  |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |      |   |          |   |         |   |          |   |          |  |         |  |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   | SUSP |   | PHYSLEEP |   | ENL1GTG |   | Reserved |   | GATEHCLK |  | STPPCLK |  |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   | RO   |   | RO       |   | RW      |   |          |   | RW       |  | RW      |  |

| Bit  | Name     | Function                                     | Description                                                                                                                                                                                                                                                                                             | Access |
|------|----------|----------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:8 | Reserved | Reserved                                     | -                                                                                                                                                                                                                                                                                                       | -      |
| 7    | SUSP     | DeepSleep (L1 suspended) status bit          | <ul style="list-style-type: none"> <li>0b0: Not in DeepSleep state.</li> <li>0b1: In DeepSleep state.</li> </ul>                                                                                                                                                                                        | RO     |
| 6    | PHYSLEEP | PHY SLEEP mode status bit (PHY in sleep).    | <ul style="list-style-type: none"> <li>0b0: PHY is not in Sleep state.</li> <li>0b1: PHY is in Sleep state.</li> </ul>                                                                                                                                                                                  | RO     |
| 5    | ENL1GTG  | L1 clock gating (Enable sleep clock gating). | This bit is used for clock gating control in L1 mode.                                                                                                                                                                                                                                                   | RW     |
| 4:2  | Reserved | Reserved                                     | -                                                                                                                                                                                                                                                                                                       | -      |
| 1    | GATEHCLK | Gate HCLK                                    | When USB communication is suspended or the session is disabled, the application sets this bit to 1 to stop the clock to all modules except the AHB bus slave interface, host interface, and wake-up logic. When USB communication resumes or a new session is started, the application clears this bit. | RW     |
| 0    | STPPCLK  | Stop PHY clock                               | When USB communication is suspended, the session is disabled, or the device is disconnected, the application sets this bit to 1 to stop the PH clock. The application program clears this bit when USB communication resumes.                                                                           | RW     |

## 36 CAN FD Controller (CAN FD)

### 36.1 Introduction

CAN (Controller Area Network) bus is a bus standard that enables communication between microprocessors or devices without a host.

The CAN FD controllers (CAN1/CAN2) comply with the CAN bus protocol 2.0A and 2.0B and are upward compatible with CAN-FD.

The CAN bus controller can handle data transmission and reception on the bus. In this product, the CAN FD controller has 16 filter groups. Filters are used to select messages for the application to receive.

In the CAN FD controller, applications can send data to the bus through 1 high-priority Primary Transmit Buffer (PTB) and 3 Secondary Transmit Buffers (STB), with the transmission sequence determined by the transmit scheduler. Bus data is received through 8 Receive Buffers (RB). The 3 STBs and 8 RBs can be understood as a 3-stage FIFO and an 8-stage FIFO, both fully controlled by hardware.

CAN FD bus controllers can also support Time-trigger communication.

### 36.2 Features

- Fully supports CAN2.0A/CAN2.0B/CAN FD protocols
- CAN2.0 supports a maximum communication baud rate of 1Mbit/s
- Supports baud rate prescaler of 1 to 1/256 for flexible baud rate configuration
- 8 receive buffers
  - ▶ FIFO mode
  - ▶ Messages stored will not be overwritten by erroneous or unreceived data
- 1 high-priority PTB
- 3 STB
  - ▶ FIFO Mode
  - ▶ Priority arbitration method
- 16 independent filter groups
  - ▶ Support 11-bit standard ID and 29-bit extended ID
  - ▶ Programmable ID CODE bits and MASK bits
- Both PTB and STB support single send mode
- Supports silent mode
- Supports loopback mode
- Supports capturing transmission error types and locating bit arbitration lost bits
- Programmable error warning value
- Supports ISO11898-4 defined time-trigger CAN and receive timestamps

## 36.3 Functional Description

### 36.3.1 System Block Diagram

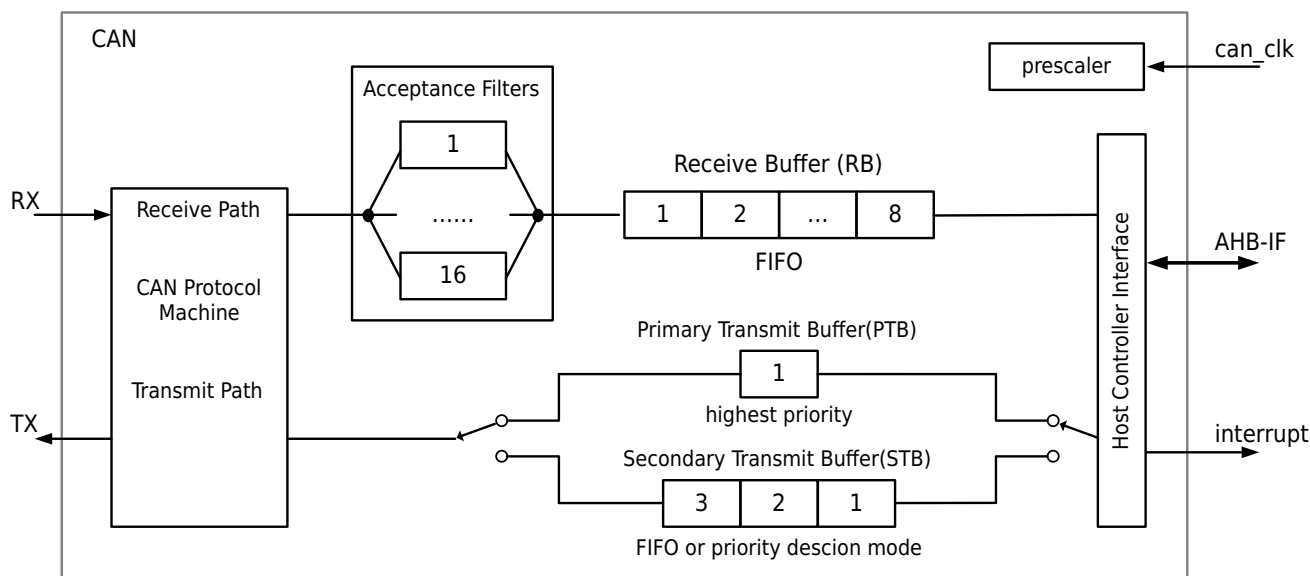


Figure 36-1 CAN FD System Block Diagram

### 36.3.2 Pin Description

Table 36-1 CAN Pin Description

| Pin name                | Direction | Description                                                                                            |
|-------------------------|-----------|--------------------------------------------------------------------------------------------------------|
| CANx_RX (x=1-2)         | Input     | CAN receive data signal                                                                                |
| CANx_TX (x=1-2)         | Output    | CAN sends data signal                                                                                  |
| CANx_TST_SAMPLE (x=1-2) | Output    | For observation only, sample bit set (communication clock width of one cycle after the sampling point) |
| CANx_TST_CLOCK (x=1-2)  | Output    | For observation only, baud rate (communication clock width of one cycle before the start of a bit)     |

### 36.3.3 Operating Mode

#### 36.3.3.1 Action Mode

Before using the CAN FD function, CAN FD function enable must be configured. Please refer to the description of the CAN global control register in the "Register Description" chapter.

The CAN FD controller has two operating modes: reset mode (CAN\_CFG\_STAT.RESET=1) and action mode (CAN\_CFG\_STAT.RESET=0). When initializing the module, first set the registers that can only be operated in reset mode (see the Register Description section for details) in reset mode, then exit reset mode and operate the remaining registers in action mode.

#### 36.3.3.2 Loopback Mode

CAN\_CTRL supports the following two loopback modes:

- Internal loopback

- External loopback

Both loopback modes can receive data frames sent by themselves, mainly used for testing purposes.

Internal loopback mode: Internally connects the receive data line to the transmit data line within the module, and transmit data is not output. In internal loopback mode, the node generates a self-acknowledge signal to avoid ACK errors.

External loopback mode maintains the connection with the transceiver, so the transmitted data can still appear on the CAN bus. With the help of the transceiver, CAN can receive its own transmitted data. In external loopback mode, the self-acknowledge signal can be generated or not by setting the SACK bit in the RCTRL register. When SACK=0, no self-acknowledge signal is generated; when SACK=1, a self-acknowledge signal is generated.

In external loopback mode with SACK=0, the following two situations may occur:

- Other nodes also receive the data frame sent by this node and send an ACK signal. In this case, the node can successfully send and receive data.
- If no other node returns an ACK signal, an ACK error will occur, causing the data to be retransmitted and the error counter to be incremented. In this case, it is recommended to use single send mode.

When returning from loopback mode to normal mode, in addition to clearing the mode bit, it is also necessary to perform a software reset of CAN\_CTRL.

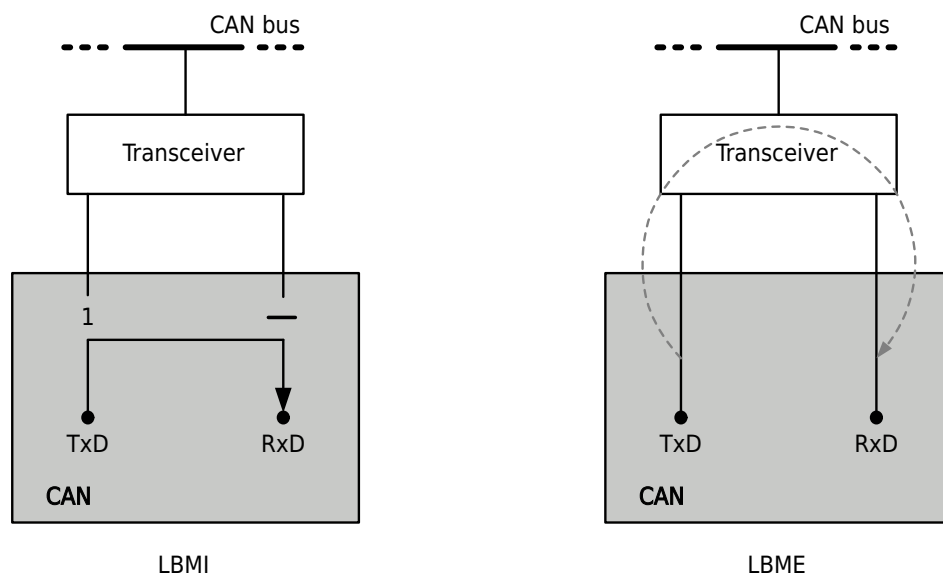


Figure 36-2 CAN FD LBMI and LBME Schematic Diagram

### 36.3.3.3 Silent Mode

Silent mode can be used to monitor CAN network data. In silent mode, data can be received from the CAN bus, but no data is sent to the bus. Set TCMD.LOM to 1 to enter silent mode, and clear it to exit silent mode.

External loopback mode can be combined with silent mode to form external loopback silent mode. In this mode, CAN can be considered a silent receiver but can send data when necessary. In external loopback silent mode, frames containing self-acknowledge signals are allowed to be sent, but the node will not generate error flags or overload frames.

### 36.3.4 Baud Rate Setting

The CAN communication clock (can\_clk) is used to generate the baud rate, which can be selected from system clock division or an independent clock (see the "Working Clock Specifications" section in the "Clock Module Unit (CMU)" chapter). It is specified by the CMU\_CANCKCFGR register. The clock selection must meet the condition that PCLK0 (CAN control logic clock) is 1.5 times or more of the CAN communication clock.

The diagram below shows the CAN bit time definition. The upper part above the dashed line represents the bit time specified by the CAN protocol, and the lower part below the dashed line represents the bit time defined by the CAN-CTRL. Where, segment1 and segment2 can be set through the SBT and FBT registers. The SBT and FBT registers can only be set when CAN\_CFG\_STAT.RESET=1, i.e., during CAN software reset. SBT register is used for the arbitration segment of CAN2.0 and CAN FD, and the FBT register is used for the CAN FD data segment.

For FD communication, it is recommended to select a communication clock of 20MHz/40MHz/80MHz.

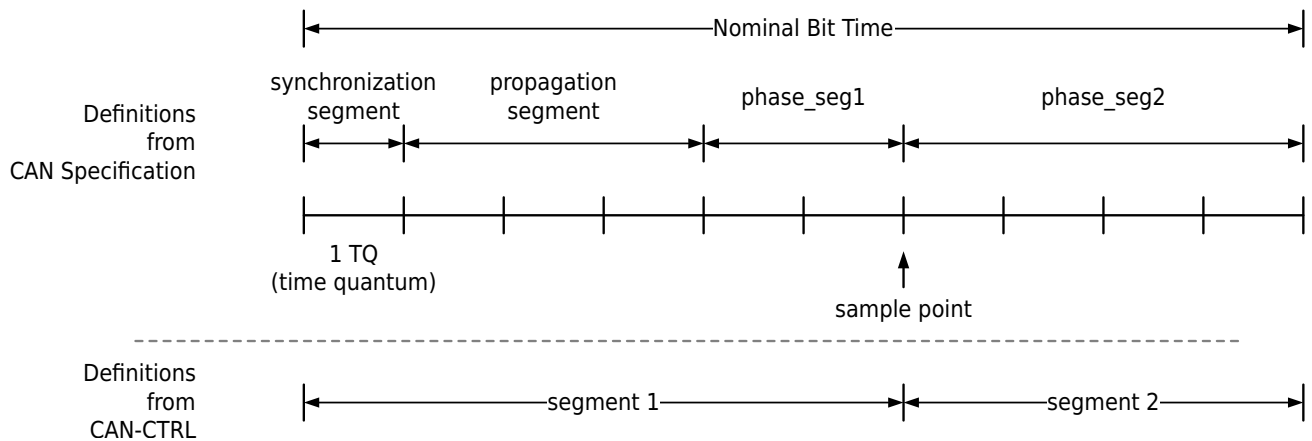


Figure 36-3 CAN Bit Time Definition Diagram

Please refer to the following formula for the TQ calculation method, where PRESC is set through the S\_PRESC bit in the SBT register or the F\_PRESC bit in the FBT register.  $f_{can\_clk}$  is the CAN communication clock frequency.

$$S\_TQ = \frac{S\_PRESC + 1}{f_{can\_clk}}$$

$$F\_TQ = \frac{F\_PRESC + 1}{f_{can\_clk}}$$

The bit timing calculation method should refer to the following formula, where S\_segment1 and S\_segment2 are configured through the S\_SEG\_1 bit and S\_SEG\_2 bit of the SBT register, and F\_segment1 and F\_segment2 are configured through the F\_SEG\_1 bit and F\_SEG\_2 bit of the FBT register.

$$Slow\_BT = t_{s\_segment1} + t_{s\_segment2} = ((S\_SEG\_1 + 2) + (S\_SEG\_2 + 1)) \times S\_TQ$$

$$Fast\_BT = t_{f\_segment1} + t_{f\_segment2} = ((F\_SEG\_1 + 2) + (F\_SEG\_2 + 1)) \times F\_TQ$$

Table 36-2 CAN Bit Time Setting Rules

| Bit         | Predetermined Area                                               | Rule                             |
|-------------|------------------------------------------------------------------|----------------------------------|
| SBT.S_SEG_1 | [0..63] CAN2.0 bits (slow)<br>[0..63] CAN FD nominal bits (slow) | SEG_1 ≥ SEG_2 + 1<br>SEG_2 ≥ SJW |
| SBT.S_SEG_2 | [0..7] CAN2.0 bits (slow)<br>[0..31] CAN FD nominal bits (slow)  |                                  |
| SBT.S_SJW   | [0..15] CAN2.0 bits (slow)<br>[0..15] CAN FD nominal bits (slow) |                                  |
| FBT.F_SEG_1 | [0..15] CAN FD data bits (fast)                                  | SEG_1 ≥ SEG_2<br>SEG_2 ≥ SJW     |
| FBT.F_SEG_2 | [0..7] CAN FD data bits (fast)                                   |                                  |
| FBT.F_SJW   | [0..7] CAN FD data bits (fast)                                   |                                  |

The following is a recommendation for CAN FD baud rate settings for reference only.

- PSP: Primary Sample Point
- SSP: Secondary Sample Point
- Seg 1: Segment 1
- Seg 2: Segment 2
- TDC: Transmitter Delay Compensation

Table 36-3 Baud Rate Setting Recommendations for 20MHz Communication Clock

| Bit Rate [Mbit/s]  | PSP [%] | SSP [%] | Prescaler | Bit Time [TQ] | Seg 1 [TQ] | Seg 2 [TQ] | SJW [TQ] | TDC [CAN Communication Clock] |
|--------------------|---------|---------|-----------|---------------|------------|------------|----------|-------------------------------|
| 0.25 (Arbitration) | 80      | -       | 1         | 80            | 64         | 16         | 16       | -                             |
| 0.5 (Arbitration)  | 80      | -       | 1         | 40            | 32         | 8          | 8        | -                             |
| 0.5                | 80      | Disable | 1         | 40            | 32         | 8          | 8        | -                             |
| 1                  | 80      | 80      | 1         | 20            | 16         | 4          | 4        | 16                            |
| 2                  | 80      | 80      | 1         | 10            | 8          | 2          | 2        | 8                             |
| 4                  | 80      | 80      | 1         | 5             | 4          | 1          | 1        | 4                             |
| 5                  | 75      | 75      | 1         | 4             | 3          | 1          | 1        | 3                             |

Table 36-4 Baud Rate Setting Recommendations for 40MHz Communication Clock

| Bit Rate [Mbit/s]  | PSP [%] | SSP [%] | Prescaler | Bit Time [TQ] | Seg 1 [TQ] | Seg 2 [TQ] | SJW [TQ] | TDC [CAN Communication Clock] |
|--------------------|---------|---------|-----------|---------------|------------|------------|----------|-------------------------------|
| 0.25 (Arbitration) | 80      | -       | 2         | 80            | 64         | 16         | 16       | -                             |
| 0.5 (Arbitration)  | 80      | -       | 1         | 80            | 64         | 16         | 16       | -                             |
| 0.5                | 80      | Disable | 2         | 40            | 32         | 8          | 8        | -                             |
| 1                  | 80      | 80      | 1         | 40            | 32         | 8          | 8        | 32                            |
| 2                  | 80      | 80      | 1         | 20            | 16         | 4          | 4        | 16                            |
| 4                  | 80      | 80      | 1         | 10            | 8          | 2          | 2        | 8                             |

| Bit Rate<br>[Mbit/s] | PSP<br>[%] | SSP<br>[%] | Prescaler | Bit Time<br>[TQ] | Seg 1<br>[TQ] | Seg 2<br>[TQ] | SJW<br>[TQ] | TDC<br>[CAN Communica-<br>tion Clock] |
|----------------------|------------|------------|-----------|------------------|---------------|---------------|-------------|---------------------------------------|
| 5                    | 75         | 75         | 1         | 8                | 6             | 2             | 2           | 6                                     |
| 8                    | 80         | 80         | 1         | 5                | 4             | 1             | 1           | 4                                     |

Table 36-5 Baud Rate Setting Recommendations for 80MHz Communication Clock

| Bit Rate<br>[Mbit/s]  | PSP<br>[%] | SSP<br>[%]   | Prescaler | Bit Time<br>[TQ] | Seg 1<br>[TQ] | Seg 2<br>[TQ] | SJW<br>[TQ] | TDC<br>[CAN Communica-<br>tion Clock] |
|-----------------------|------------|--------------|-----------|------------------|---------------|---------------|-------------|---------------------------------------|
| 0.25<br>(Arbitration) | 80         | -            | 4         | 80               | 64            | 16            | 16          | -                                     |
| 0.5<br>(Arbitration)  | 80         | -            | 2         | 80               | 64            | 16            | 16          | -                                     |
| 0.5                   | 80         | Disa-<br>ble | 4         | 40               | 32            | 8             | 8           | -                                     |
| 1                     | 80         | 80           | 2         | 40               | 32            | 8             | 8           | 64                                    |
| 2                     | 80         | 80           | 2         | 20               | 16            | 4             | 4           | 32                                    |
| 4                     | 80         | 80           | 1         | 20               | 16            | 4             | 4           | 16                                    |
| 5                     | 75         | 75           | 1         | 16               | 12            | 4             | 4           | 12                                    |
| 8                     | 80         | 80           | 1         | 10               | 8             | 2             | 2           | 8                                     |

## 36.3.5 Transmission Processing

### 36.3.5.1 Transmit Buffer

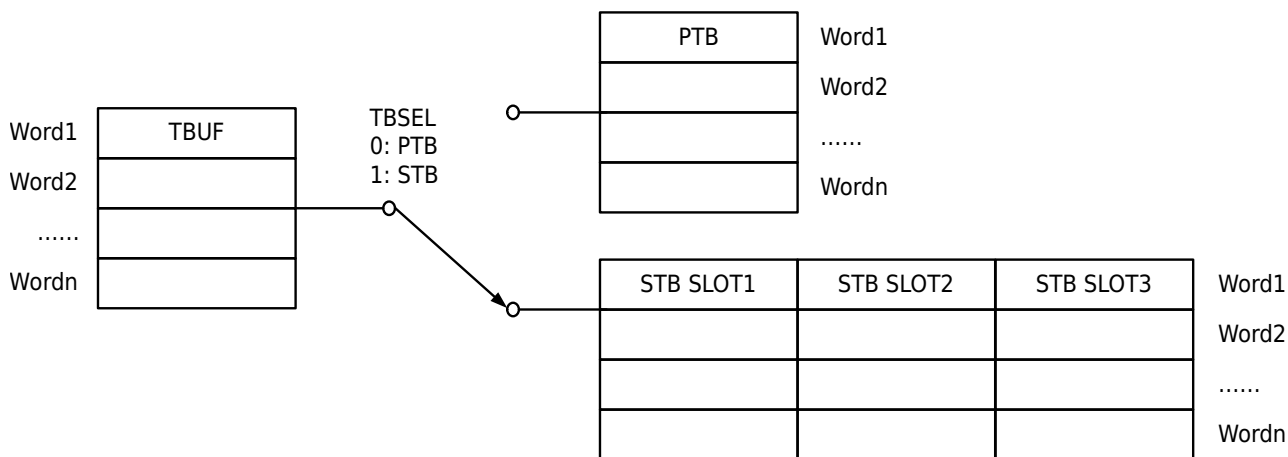
CAN\_CTRL provides two transmit buffers for sending data: the Primary Transmit Buffer (PTB) and the Secondary Transmit Buffer (STB). PTB has the highest priority but can only buffer one frame of data. STB has a lower priority than PTB but can buffer 3 frames of data, and the 3 frames in STB can operate in FIFO mode or priority arbitration mode.

The 3 frames in STB can be set to send all at once by setting the TSALL bit in the TCMD register to 1. In FIFO mode, the data written first is sent first. In priority mode, data with a smaller ID is sent first.

Data in PTB has the highest priority, so PTB transmission can delay STB transmission, but STB that has won arbitration and started transmission cannot be delayed by PTB transmission.

PTB and STB can be accessed through the TBUF register. Select PTB or STB by setting the TBSEL bit in the TCMD register. TBSEL=0 selects PTB; TBSEL=1 selects STB. Select the next SLOT in STB by setting the TSNEXT bit in the TCTRL register. The correspondence is shown in the diagram below:





Set TSNEXT bit to mark a slot filled and to jump to the next message slot

Figure 36-4 CAN FD TBUF Register Write Transmit Buffer and Schematic Diagram

### 36.3.5.2 Data Transmission

Before starting transmission, ensure that the data to be sent in PTB or STB has been filled, and then start PTB or STB transmission. Data refilling is not allowed during transmission.

The steps for setting up data transmission are as follows:

- Step1.** Set TBSEL to select the transmit BUF from PTB and STB;
- Step2.** Write the data to be sent through the TBUF register;
- Step3.** If STB is selected, set TSNEXT=1 to complete the loading of all STB SLOTS;
- Step4.** Send enable;
  - Use TPE for PTB transmission.
  - Use TSALL or TSONE for STB transmission.
- Step5.** Send complete status acknowledgment:
  - Use TPIF for PTB transmission completion, and TPIE to enable TPIF;
  - Use TSIF for STB transmission completion when using TSONE, and TSIE to enable TSIF;
  - Use TSIF for STB transmission completion when using TSALL. In this case, TSIF is set after all STB SLOT data to be sent is sent, and TSIE is used to enable TSIF.

### 36.3.5.3 Single Data Transmission

When automatic retransmission is not required, single send mode can be set through registers. The CFG\_STAT.TPSS bit is used for single send mode setting for PTB, and the CFG\_STAT.TSSS bit is used for STB. When data is successfully sent, the actions in single send mode are the same as those in normal send mode. However, when data is not successfully sent, the following results occur:

- TPIF is set (TPIE=1), and the corresponding BUF SLOT data is cleared.
- When there is an error in transmission, KOER is updated, and BEIF is set (BEIE=1).
- In case of arbitration lost, ALIF is set (ALIE=1).

In single send mode, TPIF alone cannot be relied upon to determine transmission completion. It is necessary to use BEIF and ALIF together to determine whether transmission is complete.

### 36.3.5.4 Cancel Data Transmission

Data transmission that has been requested but not yet executed can be canceled through TPA or TSA. The following situations may occur when canceling data transmission:

- During arbitration:

- ▶ If the node loses arbitration, data transmission is canceled.
- ▶ If the node wins arbitration, data transmission continues.
- Data sending:
  - ▶ Data is successfully sent and ACK is received. The corresponding flags and statuses are set normally. Data transmission is not canceled;
  - ▶ Data is successfully sent but no ACK is received. Data transmission is canceled, and the error counter is incremented.
  - ▶ For data transmission set by TSALL=1, the STB SLOT data being sent is sent normally, and the STB SLOT that has not started sending is canceled.

The results of canceling data transmission are as follows:

- TPA releases PTB and sets TPE=0.
- TSA releases one STB SLOT or all STB SLOTS depending on whether TSONE or TSALL is enabled for transmission.

### 36.3.6 Reception Processing

#### 36.3.6.1 Receive Buffer

CAN\_CTRL provides 8 SLOT receive buffers for storing received data. These 8 SLOT receive buffers operate in FIFO mode. RB SLOT received data can be read through the RBUF register. The earliest received data is always read first, and the read RB SLOT is released by setting RREL=1 in the RCTRL register, pointing to the next RB SLOT.

The diagram below shows the schematic for reading RB SLOT through RBUF.

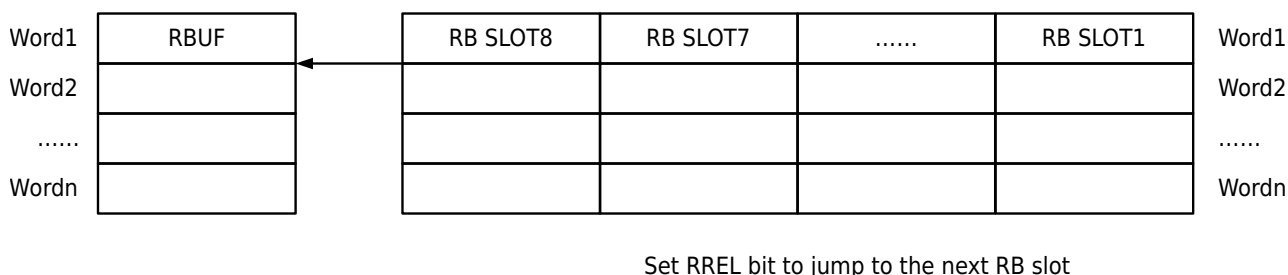


Figure 36-5 CAN RBUF Register Read Receive Buffer Schematic Diagram

#### 36.3.6.2 Receive Filter Register Set

CAN\_CTRL provides 16 groups of 32-bit filters for filtering received data to reduce CPU load. The filters can support standard format 11-bit ID or extended format 29-bit ID. Each filter group has a 32-bit ID CODE register and a 32-bit ID MASK register. The ID CODE register is used to compare the received CAN ID, and the ID MASK register is used to select the CAN ID bits for comparison. When the corresponding ID MASK bit is 1, the ID CODE of that bit is not compared.

Received data is received if it passes through any one of the 16 filter groups. The received data is stored in RB; otherwise, it is not received or stored.

Each filter group is enabled or disabled through the ACFEN register. ID CODE and ID MASK are set through the SELMASK bit in the ACFCTRL register. When SELMASK=0, it points to ID CODE; when SELMASK=1, it points to ID MASK. Filters are selected through the ACFADR bit in the ACFCTRL register. ID CODE and ID MASK are accessed through the ACF register and can only be set when CFG\_STAT.RESET=1, i.e., during CAN software reset. Refer to the diagram below for the method of accessing filter groups through the CAN ACF register.

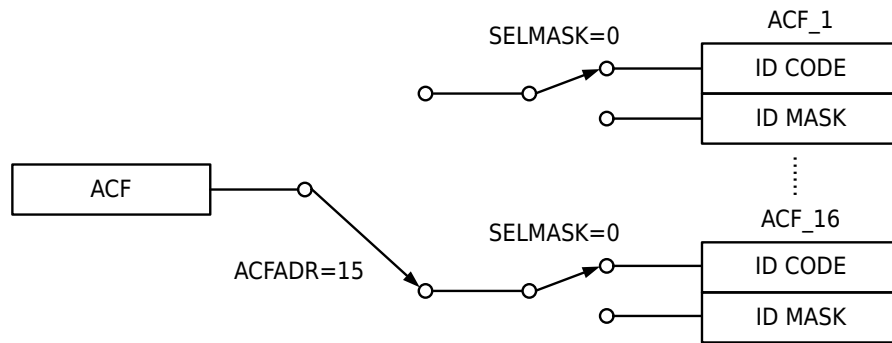


Figure 36-6 CAN ACF Register Access Filter Group Schematic Diagram

### 36.3.6.3 Data Reception

The receive filter groups can filter out unwanted received data, reducing the occurrence of interrupts and RB reads, thereby reducing CPU load. The steps for setting up received data are as follows:

- Step1.** Set filter groups.
- Step2.** Set RFIE, RAFIE, and AFWL.
- Step3.** Wait for RFIF or RAFIF.
- Step4.** Read the earliest received data from the RB FIFO through RBUF.
- Step5.** Set RREL=1 to select the next RB SLOT.
- Step6.** Repeat steps 4 and 5 until it is confirmed that RB is empty through RSTAT.

### 36.3.7 Error Handling

CAN\_CTRL can not only automatically handle some errors, such as automatically retransmitting data or discarding received frames containing errors, but also report errors to the CPU through interrupts.

CAN nodes have the following 3 error states:

- Error Active: The node automatically sends an active error flag when an error is detected.
- Error Passive: The node automatically sends a passive error flag when an error is detected.
- Bus Off: In this state, the node no longer affects the entire CAN network.

CAN\_CTRL provides two counters, TECNT and RECNT, for counting errors. The TECNT and RECNT counters are incremented or decremented according to the rules specified in the CAN protocol. Additionally, a programmable CAN error warning LIMIT register is provided to generate error interrupts to notify the CPU.

There are the following five types of errors during CAN communication, which can be identified through the KOER bit in the EALCAP register.

- Bit error
- Wrong form
- Stuff Error
- Acknowledgment error
- CRC error

### 36.3.8 BUS OFF

When the number of transmission errors exceeds 255, the CAN node automatically enters Bus Off state and no longer participates in CAN communication until it returns to Error Active state. The Bus Off state of the CAN node can be confirmed by checking the CFG\_STAT.BUSOFF bit. An EIF interrupt is generated when BUSOFF is set.

There are the following two methods for CAN to recover from Bus Off state to Error Active state:

- Power-on reset
- Received 128 consecutive 11-bit stealth bit sequences (recovery sequence)

In Bus Off state, the TECNT value remains unchanged, and RECNT is used to count the recovery sequence. After recovery from Bus Off state, TECNT and RECNT are reset to 0.

### 36.3.9 Arbitration lost Position Capture

CAN\_CTRL can precisely capture the arbitration lost bit position and reflect it in the ALC register. The ALC register stores the bit position of the most recent arbitration lost bit. If the node wins arbitration, the ALC bit is not updated.

**The ALC Value Is Defined As Follows:**

After the SOF bit, the ALC for the first ID data bit is 0, the ALC for the second ID data bit is 1, and so on. Since arbitration only occurs in the arbitration field, the maximum value of ALC is 31. For example, in an arbitration between a standard format remote frame and an extended frame, if the extended frame fails at the IDE bit, then ALC=12.

### 36.3.10 Time-Triggered TTCAN

CAN\_CTRL provides partial (level 1) hardware support for the time-triggered communication method specified in ISO11898-4. This chapter introduces the TTCAN function from the following five sections.

#### 36.3.10.1 TBUF Behavior in TTCAN Mode

##### TTTBM=1

When TTTBM=1, PTB and STB SLOTS together form TB SLOT, and the transmit BUF is specified through the TBPTR register. TBPTR=0 points to PTB, TBPTR=1 points to STB SLOT1, and so on. The host can mark transmit BUF SLOTS through the TBE and TBF registers. At this time, the TBSEL and TSNEXT registers are meaningless and can be ignored.

When TTTBM=1, PTB does not have any special attributes and is the same as STB SLOT. The transmission completion flag also uses TSIF.

In TTCAN mode, the transmit BUF does not have FIFO mode or priority arbitration mode, and only one selected SLOT can transmit data.

In TTCAN mode, transmission start requires a time-triggered method. TPE, TSONE, TSALL, TPSS, and TPA are fixed to 0 and ignored.

##### TTTBM=0

When TTTBM=0, event-driven communication and receive timestamp functions are used in combination. In this mode, the functions of PTB and STB are the same as when TTEN=0. Therefore, PTB always has the highest priority, and STB can operate in FIFO mode or priority arbitration mode.

#### 36.3.10.2 TTCAN Function Description

After power-on, the Time Master needs to be initialized according to the ISO 11898-4 protocol. In a CAN network, there can be up to eight potential Time Masters. Each Time Master has its own reference message ID (last three bits of the ID). These potential Time Masters send their respective reference messages based on their priorities.

After TTEN=1, the 16-bit counter starts working. When a reference message is successfully received or the Time Master successfully sends a reference message, the CAN controller copies Sync\_Mark to Ref\_Mark, and Ref\_Mark sets the cycle time to 0. Successfully receiving a reference message sets the RIF

flag, and successfully sending a reference message sets the TPIF flag or TSIF flag. At this time, the host needs to prepare the trigger conditions for the next action.

The trigger condition can be a receive trigger. This trigger only triggers an interrupt and can be used to detect when an expected message has not been received.

The trigger condition can also be a transmit trigger. This trigger starts transmitting data in the TBUF SLOT specified by the TTPTR register. If the selected TBUF SLOT is marked as empty, transmission does not start, but the interrupt flag is set.

### 36.3.10.3 TTCAN Timing

CAN\_CTRL supports ISO 11898-4 level 1. It includes a 16-bit counter that operates under the bit time defined by S\_PRESC, S\_SEG\_1, S\_SEG\_2. If TTEN=1, there is an additional prescaler T\_PRESC.

At the SOF of a frame of data, the counter value is Sync\_Mark. If the frame is a reference message, Sync\_Mark is copied to Ref\_Mark. The cycle time is equal to the counter value minus Ref\_Mark. This time is used as a timestamp for received messages or a trigger time baseline for transmitted messages.

### 36.3.10.4 TTCAN Trigger Mode

The TTYPE register defines the TTCAN trigger method, the TTPTR register specifies the transmit SLOT, and TT\_TRIG specifies the cycle time of the trigger.

The following 5 trigger methods are included:

- Immediate Trigger
- Time Trigger
- Single Transmit Trigger
- Transmit Start Trigger
- Transmit Stop Trigger

Except for the immediate trigger method, all triggers use the TTIF flag. When TTTBM=1, only the time-triggered method is supported.

#### Immediate Trigger

The trigger is started by writing the high bit of TT\_TRIG (the value written does not matter). In this mode, the data in the TBUF SLOT selected by TTPTR is immediately sent. TTIF is not set.

#### Time Trigger

The time-triggered method only generates an interrupt by setting the TTIF flag and has no other functions. If a node expects to receive expected data within a specific time window, the time-triggered method can be used. If the TT\_TRIG value is less than the actual cycle time, TEIF is set with no other action.

#### Single Transmit Trigger

The single transmit trigger method is used to send data within an execution time window. At this time, the TSSS bit is ignored.

The maximum of 16 cycle time ticks specified by ISO11898-4 is set through the TEW bit, with a setting range of 1-16. If data is not started to be sent within the specified transmit enable time window, the frame is discarded. The corresponding transmit BUF SLOT is marked as empty, and AIF is set. The data in the corresponding transmit BUF is not overwritten because it can be retransmitted by setting TPF.

If the TT\_TRIG value is less than the actual cycle time, TEIF is set with no other action.

### Transmit Start Trigger

The transmit start trigger method is used for arbitration within a time window. TSSS is used to determine whether to use auto-retransmit or single send mode. If the message specified by the TTPTR register is not successfully sent, the transmit stop trigger can be used to stop the transmission.

If the TT\_TRIG value is less than the actual cycle time, TEIF is set with no other action.

### Transmit Stop Trigger

The transmit stop trigger method is used to stop a transmission that has already started through the transmit start trigger method. If the transmission is stopped, the transmit frame is discarded, AIF is set, and the selected TBUF SLOT is marked as empty. However, the data in the TBUF SLOT is not overwritten and can be retransmitted by setting TPF.

If the TT\_TRIG value is less than the actual cycle time, TEIF is set and the stop is executed.

#### 36.3.10.5 TTCAN Trigger Gate Time

The TTCAN trigger watchdog time function is similar to the watchdog function and is used when TTTBM=1. It is used to watchdog the time since the last successful reception of a reference message. The reference message can be received within the cycle time or after an event. The application should set an appropriate watchdog time based on specific circumstances.

If the cycle count equals TT\_WTRIG, WTIF is set. The watchdog trigger is turned off by writing 0 to WTIE.

If TT\_WTRIG is less than the actual cycle time, TEIF is set.

### 36.3.11 TDC and RDC

During CAN FD communication, data delay may exceed one bit time, so compensation is required. TDC (Transmitter Delay Compensation) and RDC (Receiver Delay Compensation) are used for data delay compensation. TDC requires software control, while RDC is automatically enabled without software control. TDC uses a secondary sample point (SSP) to compensate for data delay. As shown in the following figure:

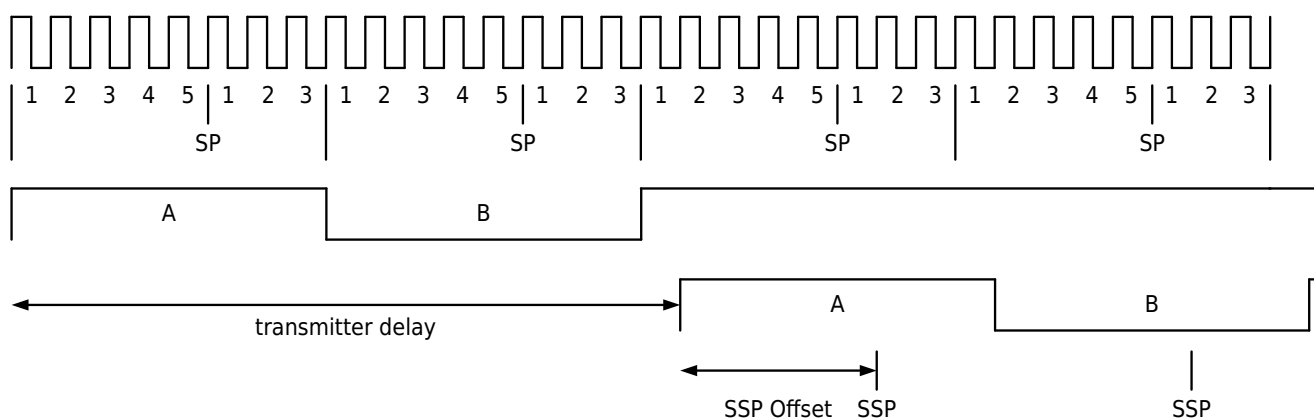


Figure 36-7 TDC Functional Block Diagram

When TDC is enabled, the controller can automatically determine the transmitter delay, and the SSP Offset can be set by configuring the TDC.SSPOFF register. It is recommended to set the SSP Offset to the same value as the  $t_{F\_segment1}$  setting.

### 36.3.12 Upward Compatibility with CAN-FD Function

When CAN FD function is disabled, CAN-CTRL automatically ignores CAN FD frames even when received in a network that includes CAN FD, without returning an ACK. It will send or receive the next CAN 2.0B frame once the bus is idle.

### 36.3.13 RAM ECC Function

Each CAN module contains a RAM memory that is used as the transmit buffer (TBUF), receive buffer (RBUF), and receive filter (ACF) for the CAN module.

In CAN RAM, the transmit buffer (TBUF) starts at address 0x00000000. Each TBUF is 72 bytes in size, arranged in order from low to high addresses as PTB, STB1-STB3. The receive buffer (RBUF) starts at address 0x00000120. Each RBUF is 72 bytes in size, arranged in order from low to high addresses as RBUF1-RBUF8. The receive filter (ACF) starts at address 0x000003A8. Each ACF is 8 bytes in size, arranged in order from low to high addresses as ACF1-ACF16.

Table 36-6 CAN RAM Address Mapping

| CAN RAM Element | CAN RAM Address Range |
|-----------------|-----------------------|
| PTB             | 0x00000000-0x00000047 |
| STB             | 0x00000048-0x0000011F |
| RBUF            | 0x00000120-0x000003A7 |
| ACF             | 0x000003A8-0x00000427 |

RAM is equipped with ECC checking, which involves correcting 1-bit errors and checking for 2-bit errors. ECC checking has an error injection function. Based on the settings of CAN\_RAM\_EIEN register and CAN\_RAM\_EIBIT0/CAN\_RAM\_EIBIT1 registers, the hardware circuitry will invert the data read from RAM bit by bit to achieve error injection. By setting CAN\_RAM\_EIBIT0/CAN\_RAM\_EIBIT1 registers, 1-bit or multi-bit errors can be induced. Since ECC checking is a single error correction and double error detection code, errors exceeding 2-bit may not be detected. The CAN\_RAM\_ECCERRADDR register records the access address when a 1-bit or 2-bit ECC error occurs in CAN RAM (as shown in the table above).

### 36.3.14 Software Reset Function

Software reset function is achieved by setting the CFG\_STAT.RESET bit to 1. The reset scope of the software reset function is shown in the table below.

Table 36-7 Software Reset Range Table

| Register Bit Name | Software Reset | Remark                              | Register Bit Name | Software Reset | Remark                                                |
|-------------------|----------------|-------------------------------------|-------------------|----------------|-------------------------------------------------------|
| ACFADR            | No             | -                                   | F_SEG_1           | Yes            | Writable only during software reset                   |
| ACODE             | No             | Writable only during software reset | F_SEG_2           | Yes            | Writable only during software reset                   |
| AE_x              | No             | -                                   | F_SJW             | Yes            | Writable only during software reset                   |
| AFWL              | No             | -                                   | KOER              | Yes            | -                                                     |
| AIF               | Yes            | -                                   | LBME              | Yes            | -                                                     |
| ALC               | Yes            | -                                   | LBMI              | Yes            | -                                                     |
| ALIE              | No             | -                                   | RACTIVE           | Yes            | Immediately stops reception and does not generate ACK |
| ALIF              | Yes            | -                                   | RAFIE             | No             | -                                                     |

| Register Bit Name | Software Reset | Remark                              | Register Bit Name | Software Reset | Remark                                     |
|-------------------|----------------|-------------------------------------|-------------------|----------------|--------------------------------------------|
| AMASK             | No             | Writable only during software reset | RAFIF             | Yes            | -                                          |
| BEIE              | No             | -                                   | RBALL             | Yes            | -                                          |
| BEIF              | Yes            | -                                   | RBUF              | Yes            | RB is marked as empty, value indeterminate |
| BUSOFF            | No             | Cleared by writing 1                | RECNT             | No             | Cleared by writing 1 to BUSOFF             |
| EIEF              | No             | -                                   | REF_ID            | No             | -                                          |
| EIF               | No             | -                                   | REF_IDE           | No             | -                                          |
| EPASS             | No             | -                                   | RFIE              | No             | -                                          |
| EPIE              | No             | -                                   | RFIF              | Yes            | -                                          |
| EPIF              | Yes            | -                                   | RIE               | No             | -                                          |
| EWARN             | No             | -                                   | RIF               | Yes            | -                                          |
| EWL               | Yes            | -                                   | ROIE              | No             | -                                          |
| FD_ISO            | No             | Writable only during software reset | ROIF              | Yes            | -                                          |
| F_FRESC           | No             | Writable only during software reset | ROM               | No             | -                                          |
| ROV               | Yes            | -                                   | TSMODE            | No             | -                                          |
| RREL              | Yes            | -                                   | TSNEXT            | Yes            | -                                          |
| RSTAT             | Yes            | -                                   | TSONE             | Yes            | -                                          |
| SACK              | Yes            | -                                   | TPIE              | No             | -                                          |
| SELMASK           | No             | -                                   | TPIF              | Yes            | -                                          |
| S_PRESC           | No             | Writable only during software reset | TPSS              | Yes            | -                                          |
| S_SEG_1           | No             | Writable only during software reset | TSFF              | Yes            | All STB SLOTS are marked as empty          |
| S_SEG_2           | No             | Writable only during software reset | TSIE              | No             | -                                          |
| S_SJW             | No             | Writable only during software reset | TSIF              | Yes            | -                                          |
| SSPOFF            | Yes            | -                                   | TSSS              | Yes            | -                                          |
| TACTIVE           | Yes            | Immediately stops transmission      | TSSTAT            | Yes            | All STB SLOTS are marked as empty          |
| TBE               | Yes            | -                                   | TTEN              | Yes            | -                                          |
| TBF               | No             | -                                   | TTIF              | Yes            | -                                          |
| TBPTR             | No             | -                                   | TTIE              | No             | -                                          |



| Register Bit Name | Software Reset | Remark                                  | Register Bit Name | Software Reset | Remark |
|-------------------|----------------|-----------------------------------------|-------------------|----------------|--------|
| TBSEL             | Yes            | -                                       | TTPTR             | No             | -      |
| TBUF              | Yes            | STB is marked as empty, pointing to PTB | TTTBM             | No             | -      |
| TDCEN             | Yes            | -                                       | TTYPE             | No             | -      |
| TECNT             | No             | Can be cleared by BUS-OFF=1             | TT_TRIG           | No             | -      |
| TEIF              | Yes            | -                                       | TT_WTRIG          | No             | -      |
| TPA               | Yes            | -                                       | T_PRESC           | No             | -      |
| TPE               | Yes            | -                                       | WTIE              | No             | -      |
| TSA               | Yes            | -                                       | WTIF              | Yes            | -      |
| TSALL             | Yes            | -                                       |                   |                |        |

### 36.3.15 Interrupt

Table 36-8 CAN Interrupt Table

| Symbol             | Interrupt flag | Description                          |
|--------------------|----------------|--------------------------------------|
| CAN_x_HOST (x=1-2) | RIF            | Receive interrupt                    |
|                    | ROIF           | Receive overflow interrupt           |
|                    | ROIF           | Receive buffer full interrupt        |
|                    | RAFIF          | Receive buffer almost full interrupt |
|                    | TPIF           | PTB send interrupt                   |
|                    | TSIF           | STB transmit interrupt               |
|                    | EIF            | Error interrupt                      |
|                    | AIF            | Cancel transmitting interrupt        |
|                    | EPIE           | Error passive interrupt              |
|                    | ALIF           | Arbitration lost interrupt           |
|                    | BEIF           | Bus error interrupt                  |
|                    | WTIF           | Trigger watchdog interrupt           |
|                    | TEIF           | Trigger error interrupt              |
|                    | TTIF           | Time triggered interrupt             |

## 36.4 Register Description

### 36.4.1 Register List

CAN System Control Register base address: 0x40055400

Table 36-9 CAN Global Control Register Offset

| Offset | Register                     | Description                 | Bit width |
|--------|------------------------------|-----------------------------|-----------|
| 0x18   | <a href="#">CAN_SYCTLREG</a> | CAN System Control Register | 32        |

- CAN1 base address: 0x40009000
- CAN2 base Address: 0x40078000

Table 36-10 CAN FD Controller (CAN FD) Register Offset

| Offset    | Register                     | Description                                        | Bit width |
|-----------|------------------------------|----------------------------------------------------|-----------|
| 0x00-0x4F | <a href="#">CAN_RBUF</a>     | CAN Receive BUF Register                           | -         |
| 0x50-0x97 | <a href="#">CAN_TBUF</a>     | CAN Transmit BUF Register                          | -         |
| 0xA0      | <a href="#">CAN_CFG_STAT</a> | CAN Configuration and Status Register              | 8         |
| 0xA1      | <a href="#">CAN_TCMD</a>     | CAN Command Register                               | 8         |
| 0xA2      | <a href="#">CAN_TCTRL</a>    | CAN Transmit Control Register                      | 8         |
| 0xA3      | <a href="#">CAN_RCTRL</a>    | CAN Receive Control Register                       | 8         |
| 0xA4      | <a href="#">CAN_RTIE</a>     | CAN Receive and Transmit Interrupt Enable Register | 8         |
| 0xA5      | <a href="#">CAN_RTIF</a>     | CAN Receive and Transmit Interrupt Flag Register   | 8         |
| 0xA6      | <a href="#">CAN_ERRINT</a>   | CAN Error Interrupt Enable and Flag Register       | 8         |
| 0xA7      | <a href="#">CAN_LIMIT</a>    | CAN Warning Limits Register                        | 8         |
| 0xA8      | <a href="#">CAN_SBT</a>      | CAN Slow Bit Timing Register                       | 32        |
| 0xAC      | <a href="#">CAN_FBT</a>      | CAN Fast Bit Timing Register                       | 32        |
| 0xB0      | <a href="#">CAN_EALCAP</a>   | CAN Error and Arbitration Lost Capture Register    | 8         |
| 0xB1      | <a href="#">CAN_TDC</a>      | CAN Transmitter Delay Compensation Register        | 8         |
| 0xB2      | <a href="#">CAN_RECNT</a>    | CAN Receive Error Counter Register                 | 8         |
| 0xB3      | <a href="#">CAN_TECNT</a>    | CAN Transmit Error Counter Register                | 8         |
| 0xB4      | <a href="#">CAN_ACFCTRL</a>  | CAN Filter Group Control Register                  | 8         |
| 0xB6      | <a href="#">CAN_ACFEN</a>    | CAN Filter Group Enable Register                   | 16        |

| Offset | Register                           | Description                                      | Bit width |
|--------|------------------------------------|--------------------------------------------------|-----------|
| 0xB8   | <a href="#">CAN_ACF</a>            | CAN Filter Group Code and Mask Register          | 32        |
| 0xBE   | <a href="#">CAN_TBSLOT</a>         | TTCAN TB Slot Pointer Register                   | 8         |
| 0xBF   | <a href="#">CAN_TTCFG</a>          | TTCAN Time Trigger Configuration Register        | 8         |
| 0xC0   | <a href="#">CAN_REF_MSG</a>        | TTCAN Reference Message Register                 | 32        |
| 0xC4   | <a href="#">CAN_TRG_CFG</a>        | TTCAN Trigger Configuration Register             | 16        |
| 0xC6   | <a href="#">CAN_TT_TRIG</a>        | TTCAN Trigger Time Register                      | 16        |
| 0xC8   | <a href="#">CAN_TT_WTRIG</a>       | TTCAN Trigger Watchdog Time Register             | 16        |
| 0x100  | <a href="#">CAN_RAM_CKCR</a>       | CAN RAM Check Control Register                   | 32        |
| 0x104  | <a href="#">CAN_RAM_CKSR</a>       | CAN RAM Check Status Register                    | 32        |
| 0x108  | <a href="#">CAN_RAM_EIEN</a>       | CAN RAM Error Injection Enable Register          | 32        |
| 0x10C  | <a href="#">CAN_RAM_EIBIT0</a>     | CAN RAM Error Injection Bit Enable Register<br>0 | 32        |
| 0x110  | <a href="#">CAN_RAM_EIBIT1</a>     | CAN RAM Error Injection Bit Enable Register<br>1 | 32        |
| 0x114  | <a href="#">CAN_RAM_ECCERRADDR</a> | CAN RAM ECC Error Address Register               | 32        |

### 36.4.2 CAN Receive BUF Register (CAN\_RBUF)

Offset: 0x00-0x4F

Reset: 0xFFFFFFFF

The RBUF register points to the RB SLOT address of the earliest received CAN mailbox. The RBUF register can be read in any order.

The KOER bit is the EALCAP.KOER register, which is meaningful only when RBALL=1.

The TX bit indicates that a message sent by itself is received in loopback mode.

The CYCLE\_TIME bit is valid only in TTCAN mode and indicates the cycle time at the start of SOF.

CAN\_RBUF only supports WORD access.

The data format of the CAN receive mailbox is as follows:

Table 36-11 Standard Format CAN Receive Mailbox Format

| Address | Bit7             | Bit6 | Bit5 | Bit4 | Bit3     | Bit2     | Bit1 | Bit0 | Function |
|---------|------------------|------|------|------|----------|----------|------|------|----------|
| RBUF    | ID[7:0]          |      |      |      |          |          |      |      | ID       |
| RBUF+1  | Reserved         |      |      |      |          | ID[10:8] |      |      | ID       |
| RBUF+2  | Reserved         |      |      |      |          |          |      |      | ID       |
| RBUF+3  | Reserved         |      |      |      |          |          |      |      | ID       |
| RBUF+4  | IDE=0            | RTR  | FDF  | BRS  | DLC[3:0] |          |      |      | Control  |
| RBUF+5  | KOER[2:0]        |      |      | TX   | Reserved |          |      |      | Status   |
| RBUF+6  | CYCLE_TIME[7:0]  |      |      |      |          |          |      |      | TTCAN    |
| RBUF+7  | CYCLE_TIME[15:8] |      |      |      |          |          |      |      | TTCAN    |
| RBUF+8  | DATA1            |      |      |      |          |          |      |      | Data     |
| RBUF+9  | DATA2            |      |      |      |          |          |      |      | Data     |
| RBUF+10 | DATA3            |      |      |      |          |          |      |      | Data     |
| ...     | ...              |      |      |      |          |          |      |      | Data     |
| RBUF+70 | DATA63           |      |      |      |          |          |      |      | Data     |
| RBUF+71 | DATA64           |      |      |      |          |          |      |      | Data     |

Table 36-12 Extended Format CAN Receive Mailbox Format

| Address | Bit7             | Bit6 | Bit5               | Bit4      | Bit3     | Bit2 | Bit1 | Bit0 | Function              |
|---------|------------------|------|--------------------|-----------|----------|------|------|------|-----------------------|
| RBUF    | ID[7:0]          |      |                    |           |          |      |      |      | ID                    |
| RBUF+1  | ID[15:8]         |      |                    |           |          |      |      |      | ID                    |
| RBUF+2  | ID[23:16]        |      |                    |           |          |      |      |      | ID                    |
| RBUF+3  | Reserved         |      |                    | ID[28:24] |          |      |      |      | ID                    |
| RBUF+4  | IDE=1            | RTR  | CAN<br>FD<br>Frame | BRS       | DLC[3:0] |      |      |      | Control               |
| RBUF+5  | KOER[2:0]        |      |                    | TX        | Reserved |      |      |      | Status                |
| RBUF+6  | CYCLE_TIME[7:0]  |      |                    |           |          |      |      |      | Time-Triggered<br>CAN |
| RBUF+7  | CYCLE_TIME[15:8] |      |                    |           |          |      |      |      | Time-Triggered<br>CAN |
| RBUF+8  | DATA1            |      |                    |           |          |      |      |      | Data                  |
| RBUF+9  | DATA2            |      |                    |           |          |      |      |      | Data                  |
| RBUF+10 | DATA3            |      |                    |           |          |      |      |      | Data                  |
| ...     | ...              |      |                    |           |          |      |      |      | Data                  |
| RBUF+70 | DATA63           |      |                    |           |          |      |      |      | Data                  |
| RBUF+71 | DATA64           |      |                    |           |          |      |      |      | Data                  |

The meanings of the control bits and status bits are as follows:

| Classification | Function                          | Description                                                                                                                                                                          |
|----------------|-----------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Control bit    | IDE (Identifier Extension)        | <ul style="list-style-type: none"> <li>0b0: standard format</li> <li>0b1: extended Format</li> </ul>                                                                                 |
|                | RTR (Remote Transmission Request) | <ul style="list-style-type: none"> <li>0b0: data frame</li> <li>0b1: remote frame (only applicable to CAN 2.0, fixed to 0 for CAN FD)</li> </ul>                                     |
|                | FDF (CAN FD frame)                | <ul style="list-style-type: none"> <li>0b0: CAN2.0 frame</li> <li>0b1: CAN FD frame</li> </ul>                                                                                       |
|                | BRS (Bit Rate Switch)             | <ul style="list-style-type: none"> <li>0b0: entire frame at low baud rate</li> <li>0b1: data and CRC at fast baud rate (only applicable to CAN FD, fixed to 0 when FDF=0)</li> </ul> |

| Classification | Function               | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |               |                  |                  |        |               |     |               |   |        |               |     |        |    |        |    |        |    |        |    |        |    |        |    |        |    |
|----------------|------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------|------------------|------------------|--------|---------------|-----|---------------|---|--------|---------------|-----|--------|----|--------|----|--------|----|--------|----|--------|----|--------|----|--------|----|
| Control bit    | DLC (Data Length Code) | Data length code. For CAN 2.0, the range is 0-8, corresponding to data lengths of 0-8 bytes. For CAN FD, the range is 0-15, corresponding to data lengths of 0-64 bytes.                                                                                                                                                                                                                                                                                                                                                            |               |                  |                  |        |               |     |               |   |        |               |     |        |    |        |    |        |    |        |    |        |    |        |    |        |    |
|                |                        | Table 36-14 DLC Control Bit                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |               |                  |                  |        |               |     |               |   |        |               |     |        |    |        |    |        |    |        |    |        |    |        |    |        |    |
|                |                        | <table><tr><th>Frame Type</th><th>DLC (binary)</th><th>Payload in Bytes</th></tr><tr><td rowspan="2">CAN2.0</td><td>0b0000-0b1000</td><td>0-8</td></tr><tr><td>0b1001-0b1111</td><td>8</td></tr><tr><td rowspan="8">CAN FD</td><td>0b0000-0b1000</td><td>0-8</td></tr><tr><td>0b1001</td><td>12</td></tr><tr><td>0b1010</td><td>16</td></tr><tr><td>0b1011</td><td>20</td></tr><tr><td>0b1100</td><td>24</td></tr><tr><td>0b1101</td><td>32</td></tr><tr><td>0b1110</td><td>48</td></tr><tr><td>0b1111</td><td>64</td></tr></table> | Frame Type    | DLC (binary)     | Payload in Bytes | CAN2.0 | 0b0000-0b1000 | 0-8 | 0b1001-0b1111 | 8 | CAN FD | 0b0000-0b1000 | 0-8 | 0b1001 | 12 | 0b1010 | 16 | 0b1011 | 20 | 0b1100 | 24 | 0b1101 | 32 | 0b1110 | 48 | 0b1111 | 64 |
|                |                        | Frame Type                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | DLC (binary)  | Payload in Bytes |                  |        |               |     |               |   |        |               |     |        |    |        |    |        |    |        |    |        |    |        |    |        |    |
|                |                        | CAN2.0                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | 0b0000-0b1000 | 0-8              |                  |        |               |     |               |   |        |               |     |        |    |        |    |        |    |        |    |        |    |        |    |        |    |
|                |                        |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | 0b1001-0b1111 | 8                |                  |        |               |     |               |   |        |               |     |        |    |        |    |        |    |        |    |        |    |        |    |        |    |
|                |                        | CAN FD                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | 0b0000-0b1000 | 0-8              |                  |        |               |     |               |   |        |               |     |        |    |        |    |        |    |        |    |        |    |        |    |        |    |
|                |                        |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | 0b1001        | 12               |                  |        |               |     |               |   |        |               |     |        |    |        |    |        |    |        |    |        |    |        |    |        |    |
|                |                        |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | 0b1010        | 16               |                  |        |               |     |               |   |        |               |     |        |    |        |    |        |    |        |    |        |    |        |    |        |    |
|                |                        |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | 0b1011        | 20               |                  |        |               |     |               |   |        |               |     |        |    |        |    |        |    |        |    |        |    |        |    |        |    |
| 0b1100         | 24                     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |               |                  |                  |        |               |     |               |   |        |               |     |        |    |        |    |        |    |        |    |        |    |        |    |        |    |
| 0b1101         | 32                     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |               |                  |                  |        |               |     |               |   |        |               |     |        |    |        |    |        |    |        |    |        |    |        |    |        |    |
| 0b1110         | 48                     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |               |                  |                  |        |               |     |               |   |        |               |     |        |    |        |    |        |    |        |    |        |    |        |    |        |    |
| 0b1111         | 64                     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |               |                  |                  |        |               |     |               |   |        |               |     |        |    |        |    |        |    |        |    |        |    |        |    |        |    |
|                |                        |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |               |                  |                  |        |               |     |               |   |        |               |     |        |    |        |    |        |    |        |    |        |    |        |    |        |    |
|                |                        |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |               |                  |                  |        |               |     |               |   |        |               |     |        |    |        |    |        |    |        |    |        |    |        |    |        |    |
|                |                        |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |               |                  |                  |        |               |     |               |   |        |               |     |        |    |        |    |        |    |        |    |        |    |        |    |        |    |
|                |                        |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |               |                  |                  |        |               |     |               |   |        |               |     |        |    |        |    |        |    |        |    |        |    |        |    |        |    |
|                |                        |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |               |                  |                  |        |               |     |               |   |        |               |     |        |    |        |    |        |    |        |    |        |    |        |    |        |    |
|                |                        |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |               |                  |                  |        |               |     |               |   |        |               |     |        |    |        |    |        |    |        |    |        |    |        |    |        |    |
|                |                        |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |               |                  |                  |        |               |     |               |   |        |               |     |        |    |        |    |        |    |        |    |        |    |        |    |        |    |
| Status Bits    | KOER                   | Same as EALCAP.KOER                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |               |                  |                  |        |               |     |               |   |        |               |     |        |    |        |    |        |    |        |    |        |    |        |    |        |    |
|                | TX                     | This bit is set to 1 when receiving a message sent by itself in loopback mode                                                                                                                                                                                                                                                                                                                                                                                                                                                       |               |                  |                  |        |               |     |               |   |        |               |     |        |    |        |    |        |    |        |    |        |    |        |    |        |    |

### 36.4.3 CAN Transmit BUF Register (CAN\_TBUF)

Offset: 0x50-0x97

Reset: 0xFFFFFFFF

The TBUF register points to the next empty CAN transmit BUF SLOT. The TBUF register can be read in any order. The corresponding TBUF SLOT is marked as written by writing 1 to TSNEXT, thereby pointing to the next TBUF SLOT.

CAN\_TBUF only supports WORD access.

The data format of the CAN sending mailbox is as follows:

Table 36-15 Standard Format CAN Transmit Mailbox Format

| Address | Bit7     | Bit6 | Bit5 | Bit4 | Bit3     | Bit2     | Bit1 | Bit0 | Function |
|---------|----------|------|------|------|----------|----------|------|------|----------|
| TBUF    | ID[7:0]  |      |      |      |          |          |      |      | ID       |
| TBUF+1  | Reserved |      |      |      |          | ID[10:8] |      |      | ID       |
| TBUF+2  | Reserved |      |      |      |          |          |      |      | ID       |
| TBUF+3  | Reserved |      |      |      |          |          |      |      | ID       |
| TBUF+4  | IDE=0    | RTR  | FDF  | BRS  | DLC[3:0] |          |      |      | Control  |
| TBUF+5  | Reserved |      |      |      |          |          |      |      | -        |
| TBUF+6  | Reserved |      |      |      |          |          |      |      | -        |
| TBUF+7  | Reserved |      |      |      |          |          |      |      | -        |
| TBUF+8  | DATA1    |      |      |      |          |          |      |      | Data     |
| TBUF+9  | DATA2    |      |      |      |          |          |      |      | Data     |
| TBUF+10 | DATA3    |      |      |      |          |          |      |      | Data     |
| ...     | ...      |      |      |      |          |          |      |      | Data     |
| TBUF+70 | DATA63   |      |      |      |          |          |      |      | Data     |
| TBUF+71 | DATA64   |      |      |      |          |          |      |      | Data     |

Table 36-16 Extended Format CAN Transmit Mailbox Format

| Address | Bit7      | Bit6 | Bit5 | Bit4      | Bit3     | Bit2 | Bit1 | Bit0 | Function |
|---------|-----------|------|------|-----------|----------|------|------|------|----------|
| TBUF    | ID[7:0]   |      |      |           |          |      |      |      | ID       |
| TBUF+1  | ID[15:8]  |      |      |           |          |      |      |      | ID       |
| TBUF+2  | ID[23:16] |      |      |           |          |      |      |      | ID       |
| TBUF+3  | Reserved  |      |      | ID[28:24] |          |      |      |      | ID       |
| TBUF+4  | IDE=1     | RTR  | FDF  | BRS       | DLC[3:0] |      |      |      | Control  |
| TBUF+5  | Reserved  |      |      |           |          |      |      |      | -        |
| TBUF+6  | Reserved  |      |      |           |          |      |      |      | -        |
| TBUF+7  | Reserved  |      |      |           |          |      |      |      | -        |
| TBUF+8  | DATA1     |      |      |           |          |      |      |      | Data     |
| TBUF+9  | DATA2     |      |      |           |          |      |      |      | Data     |
| TBUF+10 | DATA3     |      |      |           |          |      |      |      | Data     |
| ...     | ...       |      |      |           |          |      |      |      | Data     |
| TBUF+70 | DATA63    |      |      |           |          |      |      |      | Data     |
| TBUF+71 | DATA64    |      |      |           |          |      |      |      | Data     |

The meaning of the control bits is described in the "[CAN Receive BUF Register \(CAN\\_RBUF\)](#)" section.



## 36.4.4 CAN Configuration and Status Register (CAN\_CFG\_STAT)

| Offset | Bit Position |      |      |      |      |         |         |        |
|--------|--------------|------|------|------|------|---------|---------|--------|
| 0xA0   | 7            | 6    | 5    | 4    | 3    | 2       | 1       | 0      |
| Reset  | 0x80         |      |      |      |      |         |         |        |
| Name   | RESET        | LBME | LBMI | TPSS | TSSS | RACTIVE | TACTIVE | BUSOFF |
| Access | RW           | RW   | RW   | RW   | RW   | RO      | RO      | RW     |

| Bit | Name    | Function                      | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | Access |
|-----|---------|-------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 7   | RESET   | Reset request                 | <ul style="list-style-type: none"> <li>0b0: Do not request partial reset</li> <li>0b1: Request partial reset</li> </ul> <p>Some registers can only be written when RESET=1. For details, refer to the "<a href="#">Software Reset Function</a>" section.</p> <div>  <b>Note</b><br/>           After RESET=0, it takes 11 CAN bit times for the node to participate in communication.         </div> | RW     |
| 6   | LBME    | Enable external loopback mode | <ul style="list-style-type: none"> <li>0b0: Disable</li> <li>0b1: Enable</li> </ul> <div>  <b>Note</b><br/>           This bit cannot be set during communication.         </div>                                                                                                                                                                                                                  | RW     |
| 5   | LBMI    | Internal loopback mode enable | <ul style="list-style-type: none"> <li>0b0: Disable</li> <li>0b1: Enable</li> </ul> <div>  <b>Note</b><br/>           This bit cannot be set during communication.         </div>                                                                                                                                                                                                                  | RW     |
| 4   | TPSS    | PTB single transfer mode      | <ul style="list-style-type: none"> <li>0b0: Disable PTB single transmission mode</li> <li>0b1: Enable PTB single transmission mode</li> </ul>                                                                                                                                                                                                                                                                                                                                         | RW     |
| 3   | TSSS    | STB single transmission mode  | <ul style="list-style-type: none"> <li>0b0: Disable STB single transmission mode</li> <li>0b1: Enable STB single transmission mode</li> </ul>                                                                                                                                                                                                                                                                                                                                         | RW     |
| 2   | RACTIVE | Receive status signal         | <ul style="list-style-type: none"> <li>0b0: Not receiving</li> <li>0b1: Receiving</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                          | RO     |
| 1   | TACTIVE | Transmit status signal        | <ul style="list-style-type: none"> <li>0b0: Not transmitting</li> <li>0b1: Transmitting</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                    | RO     |

| Bit | Name   | Function      | Description                                                                                                                                                                                                                                                                                                | Access |
|-----|--------|---------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 0   | BUSOFF | Bus off state | <ul style="list-style-type: none"><li>● 0b0: Bus active state</li><li>● 0b1: Bus off state</li></ul> <div> <b>Note</b><br/>Writing 1 clears the TECNT and RECNT registers, but this is only for debugging purposes.</div> | RW     |

## 36.4.5 CAN Command Register (CAN\_TCMD)

| Offset | Bit Position |     |          |     |     |       |       |     |
|--------|--------------|-----|----------|-----|-----|-------|-------|-----|
| 0xA1   | 7            | 6   | 5        | 4   | 3   | 2     | 1     | 0   |
| Reset  | 0x00         |     |          |     |     |       |       |     |
| Name   | TBSEL        | LOM | Reserved | TPE | TPA | TSONE | TSALL | TSA |
| Access | RW           | RW  |          | RW  | RW  | RW    | RW    | RW  |

| Bit | Name            | Function                   | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | Access |
|-----|-----------------|----------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 7   | TBSEL           | Transmit BUF selection bit | <ul style="list-style-type: none"> <li>0b0: PTB</li> <li>0b1: STB</li> </ul> <p>When TTEN=1 &amp; TTTBM=1, TBSEL is reset to its reset value.</p> <div>  <b>Note</b><br/>           This bit needs to be held constant when writing to the TBUF register or TSNEXT bit.         </div>                                                                                                                                                                                                                                                                                                                                                                                                    | RW     |
| 6   | LOM             | Silent mode enable bit     | <ul style="list-style-type: none"> <li>0b0: Disable silent mode</li> <li>0b1: Enable silent mode</li> </ul> <p>When LOM=1 &amp; LBME=0, transmission is disabled. When LOM=1 &amp; LBME=1, acknowledgments to received frames and error frames are disabled, but data can be sent.</p> <div>  <b>Note</b><br/>           This bit cannot be set during communication.         </div>                                                                                                                                                                                                                                                                                                    | RW     |
| 5   | <i>Reserved</i> | <i>Reserved</i>            | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | -      |
| 4   | TPE             | PTB transmit enable bit    | <ul style="list-style-type: none"> <li>0b0: Disable PTB transmission</li> <li>0b1: Enable PTB transmission</li> </ul> <p>After this bit is enabled, the mailbox in PTB will be sent at the next available bit position. An ongoing STB transmission will continue, but the next pending STB transmission will be delayed until the PTB transmission is complete.</p> <p>After this bit is written to 1, it will remain 1 until the PTB transmission is completed or the transmission is canceled by TPA. Software cannot clear this bit by writing 0.</p> <p>The TPE is reset to the reset value by hardware in the following cases:</p> <ul style="list-style-type: none"> <li>RESET=1</li> <li>BUSOFF=1</li> <li>LOM=1&amp;LBME=0</li> <li>TTEN=1&amp;TTTBM=1</li> </ul> | RW     |

| Bit | Name  | Function                              | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | Access |
|-----|-------|---------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 3   | TPA   | PTB transmit cancel bit               | <ul style="list-style-type: none"> <li>● 0b0: Do not cancel</li> <li>● 0b1: Cancel a PTB transmission that has been requested by setting TPE to 1 but has not yet started</li> </ul> <p>This bit is written to 1 by software but cleared by hardware. This bit can be cleared by writing 1, so it is written to 1 simultaneously with TPE.</p> <p>This bit is reset to its reset value by hardware in the following situations:</p> <ul style="list-style-type: none"> <li>● RESET=1</li> <li>● BUSOFF=1</li> <li>● TTEN=1&amp;TTTBM=1</li> </ul>                                                             | RW     |
| 2   | TSONE | Transmit a frame STB data setting bit | <ul style="list-style-type: none"> <li>● 0b0: Do not send</li> <li>● 0b1: Send one frame of STB data</li> </ul> <p>In FIFO mode, the earliest written data is sent. In priority mode, the data with the highest priority is sent.</p> <p>This bit remains set to 1 until the STB transmission is complete or canceled by TSA. Software cannot clear this bit by writing 0.</p> <p>This bit is reset to its reset value by hardware in the following situations:</p> <ul style="list-style-type: none"> <li>● RESET=1</li> <li>● BUFOFF=1</li> <li>● LOM=1&amp;LBME=0</li> <li>● TTEN=1&amp;TTTBM=1</li> </ul> | RW     |
| 1   | TSALL | Transmit All STB Data Setting Bits    | <ul style="list-style-type: none"> <li>● 0b0: Do not send</li> <li>● 0b1: Send all data in STB</li> </ul> <p>This bit remains set to 1 until the STB transmission is complete or canceled by TSA. Software cannot clear this bit by writing 0.</p> <p>This bit is reset to its reset value by hardware in the following situations:</p> <ul style="list-style-type: none"> <li>● RESET=1</li> <li>● BUSOFF=1</li> <li>● LOM=1&amp;LBME=0</li> <li>● TTEN=1&amp;TTTBM=1</li> </ul>                                                                                                                             | RW     |
| 0   | TSA   | STB transmit cancel bit               | <ul style="list-style-type: none"> <li>● 0b0: Do not cancel</li> <li>● 0b1: Cancel an STB transmission that has been requested by setting TSONE or TSALL to 1 but has not yet started.</li> </ul> <p>This bit is written to 1 by software but cleared by hardware. This bit can clear the TSONE or TSALL bit by writing 1.</p> <p>This bit is reset to its reset value by hardware in the following situations:</p> <ul style="list-style-type: none"> <li>● RESET=1</li> <li>● BUSOFF=1</li> </ul>                                                                                                           | RW     |

## 36.4.6 CAN Transmit Control Register (CAN\_TCTRL)

| Offset | Bit Position |        |        |       |          |   |        |   |
|--------|--------------|--------|--------|-------|----------|---|--------|---|
| 0xA2   | 7            | 6      | 5      | 4     | 3        | 2 | 1      | 0 |
| Reset  | 0x90         |        |        |       |          |   |        |   |
| Name   | FD_ISO       | TSNEXT | TSMODE | TTBMB | Reserved |   | TSSTAT |   |
| Access | RW           | RW     | RW     | RW    |          |   | RO     |   |

| Bit | Name   | Function          | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | Access |
|-----|--------|-------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 7   | FD_ISO | CAN FD ISO Mode   | <ul style="list-style-type: none"> <li>0b0: Bosch CAN FD Mode</li> <li>0b1: 11898-1:2015 CAN FD Mode</li> </ul> <div>  <b>Note</b><br/>           This bit can only be written when CAN_CFG_STAT.RESET=1.         </div>                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | RW     |
| 6   | TSNEXT | Next STB SLOT     | <ul style="list-style-type: none"> <li>0b0: No action</li> <li>0b1: The current STB SLOT is filled, pointing to the next SLOT</li> </ul> <p>After the application writes data to TBUF, it sets the TSNEXT bit to indicate that the current STB SLOT has been filled, so the hardware points TBUF to the next STB SLOT.</p> <p>The data in the STB SLOT marked by the TSNEXT bit can be sent using the TSONE or TSALL bit.</p> <p>This bit is written to 1 by the application and cleared by hardware.</p> <p>After all STB SLOTS are filled, TSNEXT remains set to 1 until an STB SLOT is released.</p> <div>  <b>Note</b><br/>           This bit is fixed at 0 in TTCAN mode.         </div> | RW     |
| 5   | TSMODE | STB Transmit mode | <ul style="list-style-type: none"> <li>0b0: FIFO mode</li> <li>0b1: Priority mode</li> </ul> <p>In FIFO mode, data frames are sent in the order they are written.</p> <p>In priority mode, the ID is automatically judged, with smaller IDs having higher priority.</p> <p>In either mode, PTB has the highest priority.</p> <div>  <b>Note</b><br/>           The TSMODE bit can only be set when STB is empty.         </div>                                                                                                                                                                                                                                                                | RW     |

| Bit | Name            | Function        | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | Access |
|-----|-----------------|-----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 4   | TTTBM           | TTCAN BUF mode  | <ul style="list-style-type: none"> <li>● 0b0: Determined by TSMODE, PTB and STB</li> <li>● 0b1: Set by TBPTR and TTPTR</li> </ul> <p>In TTCAN mode, when only the receive timestamp function is required, this bit can be set to 0, and TSMODE determines whether to use PTB or STB.</p> <div>  <b>Note</b><br/>           The TSMODE bit can only be set when STB is empty.<br/>           When TTEN=0, TTTBM has no effect.         </div> | RW     |
| 3:2 | <i>Reserved</i> | <i>Reserved</i> | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | -      |
| 1:0 | TSSTAT          | STB status      | <p>When TTEN=0 or TTEN=1 &amp; TTTBM=0:</p> <ul style="list-style-type: none"> <li>● 0b00: STB empty</li> <li>● 0b01: STB is less than or equal to half full</li> <li>● 0b10: STB is more than half full</li> <li>● 0b11: STB is full</li> </ul> <p>When TTEN=1 and TTTBM=1:</p> <ul style="list-style-type: none"> <li>● 0b00: PTB and STB Empty</li> <li>● 0b01: Both PTB and STB are not full</li> <li>● 0b10: Reserved</li> <li>● 0b11: Both PTB and STB are full</li> </ul>                                              | RO     |

## 36.4.7 CAN Receive Control Register (CAN\_RCTRL)

| Offset | Bit Position |     |     |      |       |          |       |   |
|--------|--------------|-----|-----|------|-------|----------|-------|---|
| 0xA3   | 7            | 6   | 5   | 4    | 3     | 2        | 1     | 0 |
| Reset  | 0x00         |     |     |      |       |          |       |   |
| Name   | SACK         | ROM | ROV | RREL | RBALL | Reserved | RSTAT |   |
| Access | RW           | RW  | RO  | RW   | RW    |          | RO    |   |

| Bit | Name            | Function                                | Description                                                                                                                                                                                                                                                                                       | Access |
|-----|-----------------|-----------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 7   | SACK            | Self-Acknowledge                        | <ul style="list-style-type: none"> <li>0b0: No self-acknowledge</li> <li>0b1: When LBME=1, enable the self-acknowledge function</li> </ul>                                                                                                                                                        | RW     |
| 6   | ROM             | Receive BUF Overflow Mode Setting Bit   | <ul style="list-style-type: none"> <li>0b0: The earliest received data is overwritten</li> <li>0b1: Newly received data is not stored</li> </ul>                                                                                                                                                  | RW     |
| 5   | ROV             | Receive BUF Overflow Flag               | <ul style="list-style-type: none"> <li>0b0: No overflow</li> <li>0b1: Overflow, with at least one data loss<br/>Cleared by writing 1 to RREL.</li> </ul>                                                                                                                                          | RO     |
| 4   | RREL            | Receive Buffer Release                  | <ul style="list-style-type: none"> <li>0b0: Do not release</li> <li>0b1: Indicates that the receive BUF has been read, and the RBUF register points to the next RB SLOT</li> </ul>                                                                                                                | RW     |
| 3   | RBALL           | Receive BUF data stores all data frames | <ul style="list-style-type: none"> <li>0b0: Normal Mode</li> <li>0b1: Store all data, including data with errors</li> </ul>                                                                                                                                                                       | RW     |
| 2   | <i>Reserved</i> | <i>Reserved</i>                         | -                                                                                                                                                                                                                                                                                                 | -      |
| 1:0 | RSTAT           | Receive BUF Status                      | <ul style="list-style-type: none"> <li>0b00: RBUF empty</li> <li>0b01: RBUF is not empty but less than the AFWL programmed value</li> <li>0b10: RBUF is greater than or equal to the AFWL programmed value but not full</li> <li>0b11: Full (this value is maintained during overflow)</li> </ul> | RO     |

## 36.4.8 CAN Receive and Transmit Interrupt Enable Register (CAN\_RTIE)

| Offset | Bit Position |      |      |       |      |      |     |      |
|--------|--------------|------|------|-------|------|------|-----|------|
| 0xA4   | 7            | 6    | 5    | 4     | 3    | 2    | 1   | 0    |
| Reset  | 0xFE         |      |      |       |      |      |     |      |
| Name   | RIE          | ROIE | RFIE | RAFIE | TPIE | TSIE | EIE | TSFF |
| Access | RW           | RW   | RW   | RW    | RW   | RW   | RW  | RO   |

| Bit | Name  | Function                                 | Description                                                                                                                                                                                                                                                                                                                                                                                                                       | Access |
|-----|-------|------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 7   | RIE   | Receive Interrupt Enable                 | <ul style="list-style-type: none"> <li>0b0: Disable</li> <li>0b1: Enable</li> </ul>                                                                                                                                                                                                                                                                                                                                               | RW     |
| 6   | ROIE  | Receive Overflow Interrupt Enable        | <ul style="list-style-type: none"> <li>0b0: Disable</li> <li>0b1: Enable</li> </ul>                                                                                                                                                                                                                                                                                                                                               | RW     |
| 5   | RFIE  | Receive BUF Full Interrupt Enable        | <ul style="list-style-type: none"> <li>0b0: Disable</li> <li>0b1: Enable</li> </ul>                                                                                                                                                                                                                                                                                                                                               | RW     |
| 4   | RAFIE | Receive BUF Almost Full Interrupt Enable | <ul style="list-style-type: none"> <li>0b0: Disable</li> <li>0b1: Enable</li> </ul>                                                                                                                                                                                                                                                                                                                                               | RW     |
| 3   | TPIE  | PTB Transmit Interrupt Enable            | <ul style="list-style-type: none"> <li>0b0: Disable</li> <li>0b1: Enable</li> </ul>                                                                                                                                                                                                                                                                                                                                               | RW     |
| 2   | TSIE  | STB Transmit Interrupt Enable            | <ul style="list-style-type: none"> <li>0b0: Disable</li> <li>0b1: Enable</li> </ul>                                                                                                                                                                                                                                                                                                                                               | RW     |
| 1   | EIE   | Error Interrupt Enable                   | <ul style="list-style-type: none"> <li>0b0: Disable</li> <li>0b1: Enable</li> </ul>                                                                                                                                                                                                                                                                                                                                               | RW     |
| 0   | TSFF  | Transmit BUF Full Flag                   | <p>When TTEN=0 or TTTBM=0: Transmit Secondary Buffer Full Flag</p> <ul style="list-style-type: none"> <li>0b0: STB SLOT is not fully filled</li> <li>0b1: STB SLOT is fully filled</li> </ul> <p>TTEN=1 and TTTBM=1: Transmit Buffer Full Flag</p> <ul style="list-style-type: none"> <li>0b0: The transmit BUF selected by TBPTR is not fully filled</li> <li>0b1: The transmit BUF selected by TBPTR is fully filled</li> </ul> | RO     |



## 36.4.9 CAN Receive and Transmit Interrupt Flag Register (CAN\_RTIF)

| Offset | Bit Position |      |      |       |      |      |     |     |
|--------|--------------|------|------|-------|------|------|-----|-----|
| 0xA5   | 7            | 6    | 5    | 4     | 3    | 2    | 1   | 0   |
| Reset  | 0x00         |      |      |       |      |      |     |     |
| Name   | RIF          | ROIF | RFIF | RAFIF | TPIF | TSIF | EIF | AIF |
| Access | RW           | RW   | RW   | RW    | RW   | RW   | RW  | RW  |

| Bit | Name  | Function                               | Description                                                                                                                                                                                                                                                                                                                                                                                                      | Access |
|-----|-------|----------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 7   | RIF   | Receive Interrupt Flag                 | Write 1 to clear flag, write 0 has no effect<br>Read:<br><ul style="list-style-type: none"> <li>0b0: No data frame received</li> <li>0b1: A valid data frame or remote frame is received</li> </ul>                                                                                                                                                                                                              | RW     |
| 6   | ROIF  | Receive Overflow Interrupt Flag        | Write 1 to clear flag, write 0 has no effect<br>Read:<br><ul style="list-style-type: none"> <li>0b0: No RB is overwritten</li> <li>0b1: At least one RB is overwritten</li> </ul> <div>  <b>Note</b><br/>           When overflow occurs, ROIF and RFIF are set simultaneously.         </div>                                | RW     |
| 5   | RFIF  | Receive BUF Full Interrupt Flag        | Write 1 to clear flag, write 0 has no effect<br>Read:<br><ul style="list-style-type: none"> <li>0b0: RB FIFO is not full</li> <li>0b1: RB FIFO is full</li> </ul>                                                                                                                                                                                                                                                | RW     |
| 4   | RAFIF | Receive BUF Almost Full Interrupt Flag | Write 1 to clear flag, write 0 has no effect<br>Read:<br><ul style="list-style-type: none"> <li>0b0: The number of filled RB SLOTS is less than the AFWL setting</li> <li>0b1: The number of filled RB SLOTS is greater than or equal to the AFWL setting</li> </ul>                                                                                                                                             | RW     |
| 3   | TPIF  | PTB send interrupt flag                | Write 1 to clear flag, write 0 has no effect<br>Read:<br><ul style="list-style-type: none"> <li>0b0: No PTB transmission completion</li> <li>0b1: The requested PTB transmission is successfully completed</li> </ul> <div>  <b>Note</b><br/>           In TTCAN mode, TPIF is invalid, and only TSIF is used.         </div> | RW     |

| Bit | Name | Function                       | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | Access |
|-----|------|--------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 2   | TSIF | STB Transmit Interrupt Flag    | <p>Write 1 to clear flag, write 0 has no effect</p> <p>Read:</p> <ul style="list-style-type: none"> <li>● 0b0: No STB transmission completion</li> <li>● 0b1: The requested STB transmission is successfully completed</li> </ul> <div>  <b>Note</b><br/>           In TTCAN mode, TPIF is invalid, and only TSIF is used.         </div>                                                                                                                                                                                                       | RW     |
| 1   | EIF  | Error Interrupt Flag           | <p>Write 1 to clear flag, write 0 has no effect</p> <p>Read:</p> <ul style="list-style-type: none"> <li>● 0b0: The BUSOFF bit has not changed, or the relative relationship between the value of the error counter and the ERROR warning limit setting has not changed</li> <li>● 0b1: The BUSOFF bit has changed, or the relative relationship between the value of the error counter and the ERROR warning limit setting has changed. For example, the value of the error counter changes from less than the set value to greater than the set value, or from greater than the set value to less than the set value</li> </ul> | RW     |
| 0   | AIF  | Cancel Transmit Interrupt Flag | <p>Write 1 to clear flag, write 0 has no effect</p> <p>Read:</p> <ul style="list-style-type: none"> <li>● 0b0: No transmission cancellation</li> <li>● 0b1: The requested transmission message is successfully canceled through TPA and TSA</li> </ul>                                                                                                                                                                                                                                                                                                                                                                           | RW     |

## 36.4.10 CAN Error Interrupt Enable and Flag Register (CAN\_ERRINT)

| Offset | Bit Position |       |      |      |      |      |      |      |
|--------|--------------|-------|------|------|------|------|------|------|
| 0xA6   | 7            | 6     | 5    | 4    | 3    | 2    | 1    | 0    |
| Reset  | 0x00         |       |      |      |      |      |      |      |
| Name   | EWARN        | EPASS | EPIE | EPIF | ALIE | ALIF | BEIE | BEIF |
| Access | RO           | RO    | RW   | RW   | RW   | RW   | RW   | RW   |

| Bit | Name  | Function                          | Description                                                                                                                                                                                                                                                                                            | Access |
|-----|-------|-----------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 7   | EWARN | ERROR WARNING LIMIT Reached       | <ul style="list-style-type: none"> <li>0b0: RECNT or TECNT is less than the EWL setting</li> <li>0b1: RECNT or TECNT is greater than or equal to the EWL setting</li> </ul>                                                                                                                            | RO     |
| 6   | EPASS | Error Passive                     | <ul style="list-style-type: none"> <li>0b0: The node is in Error Active state</li> <li>0b1: The node is in Error Passive state</li> </ul>                                                                                                                                                              | RO     |
| 5   | EPIE  | Error Passive Interrupt Enable    | <ul style="list-style-type: none"> <li>0b0: Disable</li> <li>0b1: Enable</li> </ul>                                                                                                                                                                                                                    | RW     |
| 4   | EPIF  | Error Passive Interrupt Flag      | Write 1 to clear flag, write 0 has no effect<br>Read: <ul style="list-style-type: none"> <li>0b0: No change from Error Active to Error Passive or from Error Passive to Error Active</li> <li>0b1: A change from Error Active to Error Passive or from Error Passive to Error Active occurs</li> </ul> | RW     |
| 3   | ALIE  | Arbitration lost Interrupt Enable | <ul style="list-style-type: none"> <li>0b0: Disable</li> <li>0b1: Enable</li> </ul>                                                                                                                                                                                                                    | RW     |
| 2   | ALIF  | Arbitration lost Interrupt Flag   | Write 1 to clear flag, write 0 has no effect<br>Read: <ul style="list-style-type: none"> <li>0b0: Arbitration success</li> <li>0b1: Arbitration lost</li> </ul>                                                                                                                                        | RW     |
| 1   | BEIE  | Bus Error Interrupt Enable        | <ul style="list-style-type: none"> <li>0b0: Disable</li> <li>0b1: Enable</li> </ul>                                                                                                                                                                                                                    | RW     |
| 0   | BEIF  | Bus Error Interrupt Flag          | Write 1 to clear flag, write 0 has no effect<br>Read: <ul style="list-style-type: none"> <li>0b0: No Bus Error</li> <li>0b1: Bus Error</li> </ul>                                                                                                                                                      | RW     |

## 36.4.11 CAN Slow Bit Timing Register (CAN\_SBT)

| Offset | Bit Position |    |    |    |    |    |    |    |          |       |    |    |    |    |    |    |    |          |         |    |    |    |   |   |   |   |         |   |   |   |   |   |  |  |
|--------|--------------|----|----|----|----|----|----|----|----------|-------|----|----|----|----|----|----|----|----------|---------|----|----|----|---|---|---|---|---------|---|---|---|---|---|--|--|
| 0xA8   | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23       | 22    | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14       | 13      | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5       | 4 | 3 | 2 | 1 | 0 |  |  |
| Reset  | 0x01020203   |    |    |    |    |    |    |    |          |       |    |    |    |    |    |    |    |          |         |    |    |    |   |   |   |   |         |   |   |   |   |   |  |  |
| Name   | S_PRESC      |    |    |    |    |    |    |    | Reserved | S_SJW |    |    |    |    |    |    |    | Reserved | S_SEG_2 |    |    |    |   |   |   |   | S_SEG_1 |   |   |   |   |   |  |  |
| Access | RW           |    |    |    |    |    |    |    |          | RW    |    |    |    |    |    |    |    |          | RW      |    |    |    |   |   |   |   | RW      |   |   |   |   |   |  |  |

| Bit   | Name     | Function                                          | Description                                                                       | Access |
|-------|----------|---------------------------------------------------|-----------------------------------------------------------------------------------|--------|
| 31:24 | S_PRESC  | Prescaler Setting                                 | This register sets the module communication clock to (S_PRESC + 1) division as TQ | RW     |
| 23    | Reserved | Reserved                                          | -                                                                                 | -      |
| 22:16 | S_SJW    | Resynchronization Compensation Width Time Setting | Resynchronization compensation width time = (S_SJW + 1) * TQ                      | RW     |
| 15    | Reserved | Reserved                                          | -                                                                                 | -      |
| 14:8  | S_SEG_2  | Bit Segment 2 Time Setting                        | Segment 2 Time = (S_SEG_2 + 1) * TQ                                               | RW     |
| 7:0   | S_SEG_1  | Bit Segment 1 Time Setting                        | Segment 1 Time = (S_SEG_1 + 2) * TQ                                               | RW     |

## 36.4.12 CAN Fast Bit Timing Register (CAN\_FBT)

| Offset | Bit Position |    |    |    |    |    |    |    |          |    |    |    |       |    |    |    |          |    |    |    |         |    |   |   |          |   |   |   |         |   |   |   |  |  |  |  |
|--------|--------------|----|----|----|----|----|----|----|----------|----|----|----|-------|----|----|----|----------|----|----|----|---------|----|---|---|----------|---|---|---|---------|---|---|---|--|--|--|--|
| 0xAC   | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23       | 22 | 21 | 20 | 19    | 18 | 17 | 16 | 15       | 14 | 13 | 12 | 11      | 10 | 9 | 8 | 7        | 6 | 5 | 4 | 3       | 2 | 1 | 0 |  |  |  |  |
| Reset  | 0x01020203   |    |    |    |    |    |    |    |          |    |    |    |       |    |    |    |          |    |    |    |         |    |   |   |          |   |   |   |         |   |   |   |  |  |  |  |
| Name   | F_PRESC      |    |    |    |    |    |    |    | Reserved |    |    |    | F_SJW |    |    |    | Reserved |    |    |    | F_SEG_2 |    |   |   | Reserved |   |   |   | F_SEG_1 |   |   |   |  |  |  |  |
| Access | RW           |    |    |    |    |    |    |    |          |    |    |    | RW    |    |    |    |          |    |    |    | RW      |    |   |   |          |   |   |   | RW      |   |   |   |  |  |  |  |

| Bit   | Name     | Function                                          | Description                                                                       | Access |
|-------|----------|---------------------------------------------------|-----------------------------------------------------------------------------------|--------|
| 31:24 | F_PRESC  | Prescaler Setting                                 | This register sets the module communication clock to (F_PRESC + 1) division as TQ | RW     |
| 23:20 | Reserved | Reserved                                          | -                                                                                 | -      |
| 19:16 | F_SJW    | Resynchronization Compensation Width Time Setting | Resynchronization compensation width time = (F_SJW + 1) * TQ                      | RW     |
| 15:12 | Reserved | Reserved                                          | -                                                                                 | -      |
| 11:8  | F_SEG_2  | Bit Segment 2 Time Setting                        | Segment 2 Time = (F_SEG_2 + 1) * TQ                                               | RW     |
| 7:5   | Reserved | Reserved                                          | -                                                                                 | -      |
| 4:0   | F_SEG_1  | Bit Segment 1 Time Setting                        | Segment 1 Time = (F_SEG_1 + 2) * TQ                                               | RW     |

## 36.4.13 CAN Transmitter Delay Compensation Register (CAN\_TDC)

| Offset | Bit Position |        |   |   |   |   |   |   |
|--------|--------------|--------|---|---|---|---|---|---|
| 0xB1   | 7            | 6      | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00         |        |   |   |   |   |   |   |
| Name   | TDCEN        | SSPOFF |   |   |   |   |   |   |
| Access | RW           | RW     |   |   |   |   |   |   |

| Bit | Name   | Function                                           | Description                                                                                                   | Access |
|-----|--------|----------------------------------------------------|---------------------------------------------------------------------------------------------------------------|--------|
| 7   | TDCEN  | Transmitter Delay Compensation Function Enable Bit | Used for CAN FD BRS=1:<br><ul style="list-style-type: none"> <li>0b0: Disable</li> <li>0b1: Enable</li> </ul> | RW     |
| 6:0 | SSPOFF |                                                    | Set the unit as TQ                                                                                            | RW     |

## 36.4.14 CAN Error and Arbitration Lost Capture Register (CAN\_EALCAP)

| Offset | Bit Position |   |   |     |   |   |   |   |
|--------|--------------|---|---|-----|---|---|---|---|
| 0xB0   | 7            | 6 | 5 | 4   | 3 | 2 | 1 | 0 |
| Reset  | 0x00         |   |   |     |   |   |   |   |
| Name   | KOER         |   |   | ALC |   |   |   |   |
| Access | RO           |   |   | RO  |   |   |   |   |

| Bit | Name | Function                          | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | Access |
|-----|------|-----------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 7:5 | KOER | Error type                        | <ul style="list-style-type: none"> <li>● 0b000: No error</li> <li>● 0b001: Bit error</li> <li>● 0b010: Format error</li> <li>● 0b011: Stuff error</li> <li>● 0b100: Acknowledgment error</li> <li>● 0b101: CRC error</li> <li>● 0b110: Other errors</li> <li>● 0b111: Reserved</li> </ul> <div>  <b>Note</b><br/>           The KOER bit is updated when there is an error, and the KOER bit remains unchanged when sending and receiving normally.         </div> | RO     |
| 4:0 | ALC  | Arbitration lost position capture | When arbitration fails, ALC records the bit position where arbitration fails in a frame of data                                                                                                                                                                                                                                                                                                                                                                                                                                                      | RO     |

## 36.4.15 CAN Warning Limits Register (CAN\_LIMIT)

| Offset | Bit Position |   |   |   |     |   |   |   |
|--------|--------------|---|---|---|-----|---|---|---|
| 0xA7   | 7            | 6 | 5 | 4 | 3   | 2 | 1 | 0 |
| Reset  | 0x1B         |   |   |   |     |   |   |   |
| Name   | AFWL         |   |   |   | EWL |   |   |   |
| Access | RW           |   |   |   | RW  |   |   |   |

| Bit | Name | Function                              | Description                                                                                   | Access |
|-----|------|---------------------------------------|-----------------------------------------------------------------------------------------------|--------|
| 7:4 | AFWL | Receive BUF Almost Full Warning Limit | The setting range is 1-8<br>AFWL=0 is meaningless and is treated as AFWL=1                    | RW     |
| 3:0 | EWL  | Programmable Error Warning Limit      | Error Waring Limit = (EWL + 1) * 8<br>The setting value of this register affects the EIF flag | RW     |

## 36.4.16 CAN Receive Error Counter Register (CAN\_RECNT)

| Offset | Bit Position |   |   |   |   |   |   |   |
|--------|--------------|---|---|---|---|---|---|---|
| 0xB2   | 7            | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00         |   |   |   |   |   |   |   |
| Name   | RECNT        |   |   |   |   |   |   |   |
| Access | RO           |   |   |   |   |   |   |   |

| Bit | Name  | Function              | Description                                                                                                                                                                          | Access |
|-----|-------|-----------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 7:0 | RECNT | Receive Error Counter | The receive error counter increases or decreases according to the error counting rules specified in the CAN protocol. This counter does not overflow, with 255 as the maximum value. | RO     |



## 36.4.17 CAN Transmit Error Counter Register (CAN\_TECNT)

| Offset | Bit Position |   |   |   |   |   |   |   |
|--------|--------------|---|---|---|---|---|---|---|
| 0xB3   | 7            | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00         |   |   |   |   |   |   |   |
| Name   | TECNT        |   |   |   |   |   |   |   |
| Access | RO           |   |   |   |   |   |   |   |

| Bit | Name  | Function              | Description                                                                                                                                                                                                                                                                                                                                                                                               | Access |
|-----|-------|-----------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 7:0 | TECNT | Receive error counter | <p>The transmit error counter increases or decreases according to the error counting rules specified in the CAN protocol. This counter does not overflow, with 255 as the maximum value.</p> <div>  <b>Note</b><br/>           When entering Bus_Off, TECNT may overflow, so it may hold a smaller value.         </div> | RO     |

## 36.4.18 CAN Filter Group Control Register (CAN\_ACFCTRL)

| Offset | Bit Position |   |         |          |        |   |   |   |
|--------|--------------|---|---------|----------|--------|---|---|---|
| 0xB4   | 7            | 6 | 5       | 4        | 3      | 2 | 1 | 0 |
| Reset  | 0x00         |   |         |          |        |   |   |   |
| Name   | Reserved     |   | SELMASK | Reserved | ACFADR |   |   |   |
| Access |              |   | RW      |          | RW     |   |   |   |

| Bit | Name     | Function                       | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | Access |
|-----|----------|--------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 7:6 | Reserved | Reserved                       | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | -      |
| 5   | SELMASK  | Filter selection mask register | <ul style="list-style-type: none"> <li>0b0: ACF points to the filter ID register</li> <li>0b1: ACF points to the filter MASK register</li> </ul> Select a specific filter register group through ACFADR.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | RW     |
| 4   | Reserved | Reserved                       | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | -      |
| 3:0 | ACFADR   | Filter address                 | ACFADR points to a specific filter, and distinguishes between ID and MASK through SELMASK: <ul style="list-style-type: none"> <li>0b0000: Points to ACF_1</li> <li>0b0001: Points to ACF_2</li> <li>0b0010: Points to ACF_3</li> <li>0b0011: Points to ACF_4</li> <li>0b0100: Points to ACF_5</li> <li>0b0101: Points to ACF_6</li> <li>0b0110: Points to ACF_7</li> <li>0b0111: Points to ACF_8</li> <li>0b1000: Points to ACF_9</li> <li>0b1001: Points to ACF_10</li> <li>0b1010: Points to ACF_11</li> <li>0b1011: Points to ACF_12</li> <li>0b1100: Points to ACF_13</li> <li>0b1101: Points to ACF_14</li> <li>0b1110: Points to ACF_15</li> <li>0b1111: Points to ACF_16</li> </ul> | RW     |

## 36.4.19 CAN Filter Group Enable Register (CAN\_ACFEN)

| Offset | Bit Position |       |       |       |       |       |       |      |      |      |      |      |      |      |      |      |
|--------|--------------|-------|-------|-------|-------|-------|-------|------|------|------|------|------|------|------|------|------|
| 0xB6   | 15           | 14    | 13    | 12    | 11    | 10    | 9     | 8    | 7    | 6    | 5    | 4    | 3    | 2    | 1    | 0    |
| Reset  | 0x0001       |       |       |       |       |       |       |      |      |      |      |      |      |      |      |      |
| Name   | AE_16        | AE_15 | AE_14 | AE_13 | AE_12 | AE_11 | AE_10 | AE_9 | AE_8 | AE_7 | AE_6 | AE_5 | AE_4 | AE_3 | AE_2 | AE_1 |
| Access | RW           | RW    | RW    | RW    | RW    | RW    | RW    | RW   | RW   | RW   | RW   | RW   | RW   | RW   | RW   | RW   |

| Bit | Name  | Function      | Description                                                                         | Access |
|-----|-------|---------------|-------------------------------------------------------------------------------------|--------|
| 15  | AE_16 | ACF_16 enable | <ul style="list-style-type: none"> <li>0b0: Disable</li> <li>0b1: Enable</li> </ul> | RW     |
| 14  | AE_15 | ACF_15 enable | <ul style="list-style-type: none"> <li>0b0: Disable</li> <li>0b1: Enable</li> </ul> | RW     |
| 13  | AE_14 | ACF_14 enable | <ul style="list-style-type: none"> <li>0b0: Disable</li> <li>0b1: Enable</li> </ul> | RW     |
| 12  | AE_13 | ACF_13 enable | <ul style="list-style-type: none"> <li>0b0: Disable</li> <li>0b1: Enable</li> </ul> | RW     |
| 11  | AE_12 | ACF_12 enable | <ul style="list-style-type: none"> <li>0b0: Disable</li> <li>0b1: Enable</li> </ul> | RW     |
| 10  | AE_11 | ACF_11 enable | <ul style="list-style-type: none"> <li>0b0: Disable</li> <li>0b1: Enable</li> </ul> | RW     |
| 9   | AE_10 | ACF_10 enable | <ul style="list-style-type: none"> <li>0b0: Disable</li> <li>0b1: Enable</li> </ul> | RW     |
| 8   | AE_9  | ACF_9 enable  | <ul style="list-style-type: none"> <li>0b0: Disable</li> <li>0b1: Enable</li> </ul> | RW     |
| 7   | AE_8  | ACF_8 enable  | <ul style="list-style-type: none"> <li>0b0: Disable</li> <li>0b1: Enable</li> </ul> | RW     |
| 6   | AE_7  | ACF_7 enable  | <ul style="list-style-type: none"> <li>0b0: Disable</li> <li>0b1: Enable</li> </ul> | RW     |
| 5   | AE_6  | ACF_6 enable  | <ul style="list-style-type: none"> <li>0b0: Disable</li> <li>0b1: Enable</li> </ul> | RW     |
| 4   | AE_5  | ACF_5 enable  | <ul style="list-style-type: none"> <li>0b0: Disable</li> <li>0b1: Enable</li> </ul> | RW     |
| 3   | AE_4  | ACF_4 enable  | <ul style="list-style-type: none"> <li>0b0: Disable</li> <li>0b1: Enable</li> </ul> | RW     |
| 2   | AE_3  | ACF_3 enable  | <ul style="list-style-type: none"> <li>0b0: Disable</li> <li>0b1: Enable</li> </ul> | RW     |

| Bit | Name | Function     | Description                                                                          | Access |
|-----|------|--------------|--------------------------------------------------------------------------------------|--------|
| 1   | AE_2 | ACF_2 enable | <ul style="list-style-type: none"><li>● 0b0: Disable</li><li>● 0b1: Enable</li></ul> | RW     |
| 0   | AE_1 | ACF_1 enable | <ul style="list-style-type: none"><li>● 0b0: Disable</li><li>● 0b1: Enable</li></ul> | RW     |

## 36.4.20 CAN Filter Group Code and Mask Register (CAN\_ACF)

| Offset | Bit Position |       |      |             |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|-------|------|-------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0xB8   | 31           | 30    | 29   | 28          | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0xFFFFFFFF   |       |      |             |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | Reserved     | AIDEE | AIDE | ACODE/AMASK |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access |              | RW    | RW   | RW          |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit  | Name            | Function                    | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | Access |
|------|-----------------|-----------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31   | Reserved        | Reserved                    | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | -      |
| 30   | AIDEE           | IDE Bit Comparison Enable   | <ul style="list-style-type: none"> <li>0b0: The filter receives standard format and extended format frames</li> <li>0b1: The filter receives standard format or extended format frames defined by the AIDE bit</li> </ul> <div>  <b>Note</b><br/> Only valid when SELMASK=1. </div>                                                                                                                       | RW     |
| 29   | AIDE            | IDE Bit MASK                | <ul style="list-style-type: none"> <li>0b0: The filter only receives standard format frames</li> <li>0b1: The filter only receives extended format frames</li> </ul>                                                                                                                                                                                                                                                                                                                        | RW     |
| 28:0 | ACODE/<br>AMASK | Filter CODE/<br>Filter MASK | ACODE: <ul style="list-style-type: none"> <li>Point to a specific filter through ACFADR</li> <li>When SELMASK=0, it represents the filter's CODE</li> <li>For standard format, use Bit10-Bit0; for extended format, use Bit28-Bit0</li> </ul> AMASK: <ul style="list-style-type: none"> <li>Point to a specific filter through ACFADR</li> <li>When SELMASK=1, it represents the filter's MASK</li> <li>For standard format, use Bit10-Bit0; for extended format, use Bit28-Bit0</li> </ul> | RW     |

## 36.4.21 TTCAN TB Slot Pointer Register (CAN\_TBSLOT)

| Offset | Bit Position |     |          |   |   |       |   |   |
|--------|--------------|-----|----------|---|---|-------|---|---|
| 0xBE   | 7            | 6   | 5        | 4 | 3 | 2     | 1 | 0 |
| Reset  | 0x00         |     |          |   |   |       |   |   |
| Name   | TBE          | TBF | Reserved |   |   | TBPTR |   |   |
| Access | RW           | RW  |          |   |   | RW    |   |   |

| Bit | Name     | Function          | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | Access |
|-----|----------|-------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 7   | TBE      | Set TB Empty      | <ul style="list-style-type: none"> <li>0b0: No operation</li> <li>0b1: The SLOT selected by TBPTR is marked as empty</li> </ul> <p>When the SLOT is marked as empty and TSFF=0, TBE is automatically reset to 0.</p> <p>If this bit is set to 1 and there is data being transmitted in the selected SLOT, TBE=1. TBE is reset to 0 after transmission completion, transmission error, or transmission cancellation.</p> <p>TBE has higher priority than TBF.</p>                                                                                                     | RW     |
| 6   | TBF      | Set TB as filled. | <ul style="list-style-type: none"> <li>0b0: No operation</li> <li>0b1: The SLOT selected by TBPTR is marked as filled</li> </ul> <p>When the SLOT is marked as filled and TSFF=1, TBE is automatically reset to 0.</p>                                                                                                                                                                                                                                                                                                                                               | RW     |
| 5:3 | Reserved | Reserved          | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | -      |
| 2:0 | TBPTR    | TB SLOT Pointer   | <ul style="list-style-type: none"> <li>0b000: Points to PTB</li> <li>0b001: Points to STB SLOT1</li> <li>0b010: Points to STB SLOT2</li> <li>0b011: Points to STB SLOT3</li> <li>Others: Setting prohibited</li> </ul> <p>The pointed TB SLOT can be read/written via TBUF and marked as filled or empty using TBE and TBF. In TTCAN mode, TBSEL and TSNEXT registers are invalid.</p> <div>  <b>Note</b><br/>           This bit can only be written when TSFF=0.         </div> | RW     |

## 36.4.22 TTCAN Time Trigger Configuration Register (CAN\_TTCFG)

| Offset | Bit Position |      |      |      |      |         |   |      |
|--------|--------------|------|------|------|------|---------|---|------|
| 0xBF   | 7            | 6    | 5    | 4    | 3    | 2       | 1 | 0    |
| Reset  | 0x90         |      |      |      |      |         |   |      |
| Name   | WTIE         | WTIF | TEIF | TTIE | TTIF | T_PRESC |   | TTEN |
| Access | RW           | RW   | RW   | RW   | RW   | RW      |   | RW   |

| Bit | Name | Function                           | Description                                                                                                                                                                                                                                                                                                               | Access |
|-----|------|------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 7   | WTIE | Trigger Watch-dog Interrupt Enable | <ul style="list-style-type: none"> <li>0b0: Disable</li> <li>0b1: Enable</li> </ul>                                                                                                                                                                                                                                       | RW     |
| 6   | WTIF | Trigger Watch-dog Interrupt Flag   | Write 1 to clear flag, write 0 has no effect<br>Read: <ul style="list-style-type: none"> <li>0b0: No flag generated</li> <li>0b1: Generate flag</li> </ul> When CYCLE COUNT value = TT_WTRIG setting and WTIE=1, WTIF is set.                                                                                             | RW     |
| 5   | TEIF | Trigger Error Interrupt Flag       | Write 1 to clear flag, write 0 has no effect<br>Read: <ul style="list-style-type: none"> <li>0b0: No flag generated</li> <li>0b1: Generate flag</li> </ul> When TT_TTIG setting is less than the actual CYCLE_TIME, TEIF is set.                                                                                          | RW     |
| 4   | TTIE | Time Trigger Interrupt Enable      | <ul style="list-style-type: none"> <li>0b0: Disable</li> <li>0b1: Enable</li> </ul>                                                                                                                                                                                                                                       | RW     |
| 3   | TTIF | Time Trigger Interrupt Flag        | Write 1 to clear flag, write 0 has no effect<br>Read: <ul style="list-style-type: none"> <li>0b0: No flag generated</li> <li>0b1: Generate flag</li> </ul> When CYCLE COUNT value = TT_TRIG setting and TTIE=1, TTIF is set.<br>If TT_TRIG is not updated, TTIF is set only once, and is not set in the next basic CYCLE. | RW     |

| Bit | Name    | Function                | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | Access |
|-----|---------|-------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 2:1 | T_PRESC | TTCAN Counter Prescaler | <ul style="list-style-type: none"> <li>● 0b00: 1 division of the bit time set by the SBT register</li> <li>● 0b01: 2 division of the bit time set by the SBT register</li> <li>● 0b10: 4 division of the bit time set by the SBT register</li> <li>● 0b11: 8 division of the bit time set by the SBT register</li> </ul> <div>  <b>Note</b><br/> T_PRESC can be written when TTEN=0 or when writing TTEN=1 simultaneously. </div> | RW     |
| 0   | TTEN    | TTCAN Enable            | <ul style="list-style-type: none"> <li>● 0b0: Disable</li> <li>● 0b1: Enable TTCAN, and the counter starts counting</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                     | RW     |



## 36.4.23 TTCAN Reference Message Register (CAN\_REF\_MSG)

| Offset | Bit Position |          |        |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----------|--------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0xC0   | 31           | 30       | 29     | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |          |        |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | REF_IDE      | Reserved | REF_ID |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access | RW           |          | RW     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit   | Name     | Function                        | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | Access |
|-------|----------|---------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31    | REF_IDE  | IDE bits for reference messages | <ul style="list-style-type: none"> <li>0b0: Standard format</li> <li>0b1: Extended format</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | RW     |
| 30:29 | Reserved | Reserved                        | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | -      |
| 28:0  | REF_ID   | Reference Message ID Bit        | <ul style="list-style-type: none"> <li>REF_IDE=0: REF_ID[10:0] is valid</li> <li>REF_IDE=1: REF_ID[28:0] is valid</li> </ul> <p>REF_ID is used to detect reference messages, applicable to both transmission and reception. After detecting a reference message, the Sync_Mark of the current frame becomes Ref_Mark.</p> <p>REF_ID[2:0] is fixed at 0 and its value is not checked, supporting up to 8 potential time masters.</p> <p>After the highest byte write operation of REF_MSG, wait for 6 CAN clock cycles to complete the transfer of REF_MSG to the CAN clock domain.</p> | RW     |

## 36.4.24 TTCAN Trigger Configuration Register (CAN\_TRG\_CFG)

| Offset | Bit Position |    |    |    |          |       |   |   |   |          |   |   |   |       |   |   |
|--------|--------------|----|----|----|----------|-------|---|---|---|----------|---|---|---|-------|---|---|
| 0xC4   | 15           | 14 | 13 | 12 | 11       | 10    | 9 | 8 | 7 | 6        | 5 | 4 | 3 | 2     | 1 | 0 |
| Reset  | 0x0000       |    |    |    |          |       |   |   |   |          |   |   |   |       |   |   |
| Name   | TEW          |    |    |    | Reserved | TTYPE |   |   |   | Reserved |   |   |   | TTPTR |   |   |
| Access | RW           |    |    |    |          | RW    |   |   |   |          |   |   |   | RW    |   |   |

| Bit   | Name     | Function                         | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | Access |
|-------|----------|----------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 15:12 | TEW      | Transmit Enable Window           | Used for TTCAN Single Shot Transmit Trigger, a window of TEW+1 cycle time can be set, and transmission is only allowed within this window.                                                                                                                                                                                                                                                                                                                                                                            | RW     |
| 11    | Reserved | Reserved                         | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | -      |
| 10:8  | TTYPE    | Trigger Type                     | <ul style="list-style-type: none"> <li>0b000: Immediate Trigger for immediate transmission</li> <li>0b001: Time Trigger for receive triggers</li> <li>0b010: Single Shot Transmit Trigger for exclusive time windows</li> <li>0b011: Transmit Start Trigger for merged arbitrating time windows</li> <li>0b100: Transmit Stop Trigger for merged arbitrating time windows</li> <li>Others: Setting prohibited</li> </ul> Trigger time is set through the TT_TRIG register, and the TB Slot is selected through TTPTR. | RW     |
| 7:3   | Reserved | Reserved                         | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | -      |
| 2:0   | TTPTR    | Transmit Trigger TB slot Pointer | <ul style="list-style-type: none"> <li>0b000: Points to PTB</li> <li>0b001: Points to STB SLOT1</li> <li>0b010: Points to STB SLOT2</li> <li>0b011: Points to STB SLOT3</li> <li>Others: Setting prohibited</li> </ul> If the pointed TB SLOT is marked empty, TEIF is set when the trigger time is reached.                                                                                                                                                                                                          | RW     |

## 36.4.25 TTCAN Trigger Time Register (CAN\_TT\_TRIG)

| Offset | Bit Position |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0xC6   | 15           | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x0000       |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | TT_TRIG      |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access | RW           |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit  | Name    | Function     | Description                                                                                                                                                                                                                                                                                                                             | Access |
|------|---------|--------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 15:0 | TT_TRIG | Trigger Time | Used to specify the cycle time for the trigger. For transmit triggers, the transmit SOF time is approximately TT_TRIG setting + 1.<br>After the highest byte write operation of TT_TRIG, the TT_TRIG value starts transferring to the CAN clock domain. Therefore, if writing by BYTE, write the low byte first and then the high byte. | RW     |

## 36.4.26 TTCAN Trigger Watchdog Time Register (CAN\_TT\_WTRIG)

| Offset | Bit Position |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0xC8   | 15           | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0xFFFF       |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | TT_WTRIG     |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access | RW           |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit  | Name     | Function     | Description                                                                                                                                                                                                                                                     | Access |
|------|----------|--------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 15:0 | TT_WTRIG | Trigger Time | Used to specify the cycle time for the watchdog trigger.<br>After the highest byte write operation of TT_WTRIG, the TT_WTRIG value starts transferring to the CAN clock domain. Therefore, if writing by BYTE, write the low byte first and then the high byte. | RW     |

## 36.4.27 CAN RAM Check Control Register (CAN\_RAM\_CKCR)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |        |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|--------|---|
| 0x100  | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1      | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |        |   |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   | ECCMOD |   |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   | RW     |   |

| Bit  | Name     | Function                     | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | Access |
|------|----------|------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:2 | Reserved | Reserved                     | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | -      |
| 1:0  | ECCMOD   | CAN RAM ECC Check Enable Bit | <ul style="list-style-type: none"> <li>● 0b00: Disable ECC check function</li> <li>● 0b01: <ul style="list-style-type: none"> <li>▶ If a 1-bit error occurs, ECC corrects it without generating a 1-bit error flag or interrupt</li> <li>▶ If a 2-bit error occurs, ECC detects the error, sets the 2-bit error flag, and generates an interrupt</li> </ul> </li> <li>● 0b10: <ul style="list-style-type: none"> <li>▶ If a 1-bit error occurs, ECC corrects the error, sets the 1-bit error flag, and no interrupt is generated</li> <li>▶ If a 2-bit error occurs, ECC detects the error, sets the 2-bit error flag, and generates an interrupt</li> </ul> </li> <li>● 0b11: <ul style="list-style-type: none"> <li>▶ If a 1-bit error occurs, ECC corrects the error, sets the 1-bit error flag, and generates an interrupt</li> <li>▶ If a 2-bit error occurs, ECC detects the error, sets the 2-bit error flag, and generates an interrupt</li> </ul> </li> </ul> | RW     |

## 36.4.28 CAN RAM Check Status Register (CAN\_RAM\_CKSR)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |          |          |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|----------|----------|
| 0x104  | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1        | 0        |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |          |          |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   | RAM_2ERR | RAM_1ERR |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   | RW       | RW       |

| Bit  | Name     | Function                     | Description                                                                                                                                                                          | Access |
|------|----------|------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:2 | Reserved | Reserved                     | -                                                                                                                                                                                    | -      |
| 1    | RAM_2ERR | CAN RAM ECC 2-bit Error Flag | Write 1 to clear flag, write 0 has no effect<br>Read:<br><ul style="list-style-type: none"> <li>0b0: No 2-bit ECC error occurred</li> <li>0b1: A 2-bit ECC error occurred</li> </ul> | RW     |
| 0    | RAM_1ERR | CAN RAM ECC 1-bit Error Flag | Write 1 to clear flag, write 0 has no effect<br>Read:<br><ul style="list-style-type: none"> <li>0b0: No 1-bit ECC error occurred</li> <li>0b1: A 1-bit ECC error occurred</li> </ul> | RW     |

## 36.4.29 CAN RAM Error Injection Enable Register (CAN\_RAM\_EIEN)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |      |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|------|
| 0x108  | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0    |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |      |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   | EIEN |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   | RW   |

| Bit  | Name     | Function               | Description                                                                         | Access |
|------|----------|------------------------|-------------------------------------------------------------------------------------|--------|
| 31:1 | Reserved | Reserved               | -                                                                                   | -      |
| 0    | EIEN     | Fault injection enable | <ul style="list-style-type: none"> <li>0b0: Disable</li> <li>0b1: Enable</li> </ul> | RW     |

## 36.4.30 CAN RAM Error Injection Bit Enable Register 0 (CAN\_RAM\_EIBIT0)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0x10C  | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | EIBIT[31:0]  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access | RW           |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit  | Name        | Function                   | Description                                                                                                                                                                                                                                                                                                                                                                                 | Access |
|------|-------------|----------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:0 | EIBIT[31:0] | Error injection enable bit | Controls whether bits 31-0 of the CAN RAM output data are inverted: <ul style="list-style-type: none"> <li>0b0: No inversion, no ECC error injection for this bit</li> <li>0b1: Invert, inject ECC error for this bit</li> </ul> <div>  <b>Note</b><br/>Each bit corresponds to an output data bit. </div> | RW     |

## 36.4.31 CAN RAM Error Injection Bit Enable Register 1 (CAN\_RAM\_EIBIT1)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |              |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|--------------|---|---|---|---|---|---|---|
| 0x110  | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7            | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |              |   |   |   |   |   |   |   |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   | EIBIT[38:32] |   |   |   |   |   |   |   |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   | RW           |   |   |   |   |   |   |   |

| Bit  | Name         | Function                   | Description                                                                                                                                                                                                                                                                                                                                                                                      | Access |
|------|--------------|----------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:7 | Reserved     | Reserved                   | -                                                                                                                                                                                                                                                                                                                                                                                                | -      |
| 6:0  | EIBIT[38:32] | Error injection enable bit | Controls whether bits 38-32 of the CAN RAM output data are inverted: <ul style="list-style-type: none"> <li>0b0: No inversion, no ECC error injection for this bit.</li> <li>0b1: Invert, inject ECC error for this bit.</li> </ul> <div>  <b>Note</b><br/>Each bit corresponds to an output data bit. </div> | RW     |

## 36.4.32 CAN RAM ECC Error Address Register (CAN\_RAM\_ECCERRADDR)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |            |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|------------|---|---|---|---|---|---|---|---|---|
| 0x114  | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9          | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |            |   |   |   |   |   |   |   |   |   |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | ECCERRADDR |   |   |   |   |   |   |   |   |   |
|        |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |            |   |   |   |   |   |   |   |   |   |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RO         |   |   |   |   |   |   |   |   |   |
|        |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |            |   |   |   |   |   |   |   |   |   |

| Bit   | Name       | Function          | Description                                                                      | Access |
|-------|------------|-------------------|----------------------------------------------------------------------------------|--------|
| 31:11 | Reserved   | Reserved          | -                                                                                | -      |
| 10:0  | ECCERRADDR | ECC Error Address | Records the access address when a 1-bit or 2-bit ECC error occurs in the CAN RAM | RO     |

## 36.4.33 CAN System Control Register (CAN\_SYCTLREG)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |         |         |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---------|---------|
| 0x18   | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1       | 0       |
| Reset  | 0x00000002   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |         |         |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   | CAN2FDE | CAN1FDE |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   | RW      | RW      |

| Bit  | Name     | Function                | Description                                                                         | Access |
|------|----------|-------------------------|-------------------------------------------------------------------------------------|--------|
| 31:2 | Reserved | Reserved                | -                                                                                   | -      |
| 1    | CAN2FDE  | CAN2 FD function enable | <ul style="list-style-type: none"> <li>0b0: Disable</li> <li>0b1: Enable</li> </ul> | RW     |
| 0    | CAN1FDE  | CAN1 FD function enable | <ul style="list-style-type: none"> <li>0b0: Disable</li> <li>0b1: Enable</li> </ul> | RW     |

## 36.5 Precautions

### 36.5.1 CAN Bus Anti-Interference Measures

The CAN bus is widely used in the automotive, industrial control, and other industries. If the electromagnetic environment of the CAN application site is harsh, with factors such as circuit imbalance, spatial electromagnetic fields, and power grid inlet lines, it can cause a large amount of communication noise on the CAN bus due to radiation and conduction interference. This leads to an increase in bus error frames, frequent retransmissions, and a failure for correct data to arrive in a timely manner, seriously affecting data communication quality. Therefore, in practical applications, efforts should be made to eliminate noise interference and ensure stable operation of the CAN bus network.

The following are several commonly used CAN bus anti-interference measures (including but not limited to):

- Add Electrical Isolation for the CAN Bus Interface
- Common transceiver signal ground
- Use shielded twisted pair cables and ground them properly
- Increase the twisting rate of CAN transmission lines
- Add signal protector
- Improve network topology
- Application layer software anti-jamming mechanism

### 36.5.2 CAN Controller Noise Constraints

In the CAN bus network, ensure that the communication bit time meets the requirements of the standard protocol. If noise interference that does not meet the bit time width is introduced, it may cause abnormal operation of the CAN controller.



## 37 Controller Area Network (MCAN)

### 37.1 Introduction

The Controller Area Network (CAN) subsystem consists of two CAN modules, a shared message RAM memory for the base addresses of these components, see the "Memory Mapping" chapter.

Both CAN modules (MCAN1 and MCAN2) comply with ISO 11898-1:2015 (CAN Protocol Specification Version 2.0 Parts A, B) and CAN FD Protocol Specification Version 1.0 (CAN with Flexible Data-Rate Specification Version 1.0).

The 4KB message RAM memory can realize the filter (Rx Filter), receive FIFO (Rx FIFO), receive buffer (Rx Buffer), transmit event FIFO (Tx Event FIFO), and transmit buffer (Tx Buffer) functions. The message RAM is shared between MCAN1 and MCAN2 modules.

### 37.2 Features

- Compliant with ISO 11898-1:2015 standard
- CAN FD supports up to 64 bytes of data per frame for transmission and reception
- Supports CAN error logging
- Supports AUTOSAR and SAE J1939
- Support the receive filtering function
- Two configurable receive FIFOs
- Dedicated signal output on reception of high-priority messages
- Up to 64 dedicated receive buffers
- Up to 32 dedicated transmit buffers
- Configurable transmit FIFO or queue
- Configurable transmit event FIFO
- Message RAM accessible directly by the host CPU
- MCAN1 and MCAN2 share a common message RAM
- Supports programmable loopback test mode
- Supports maskable module interrupts
- Two clock domains: CAN core clock and APB bus interface clock

## 37.3 Functional Description

### 37.3.1 MCAN System Block Diagram

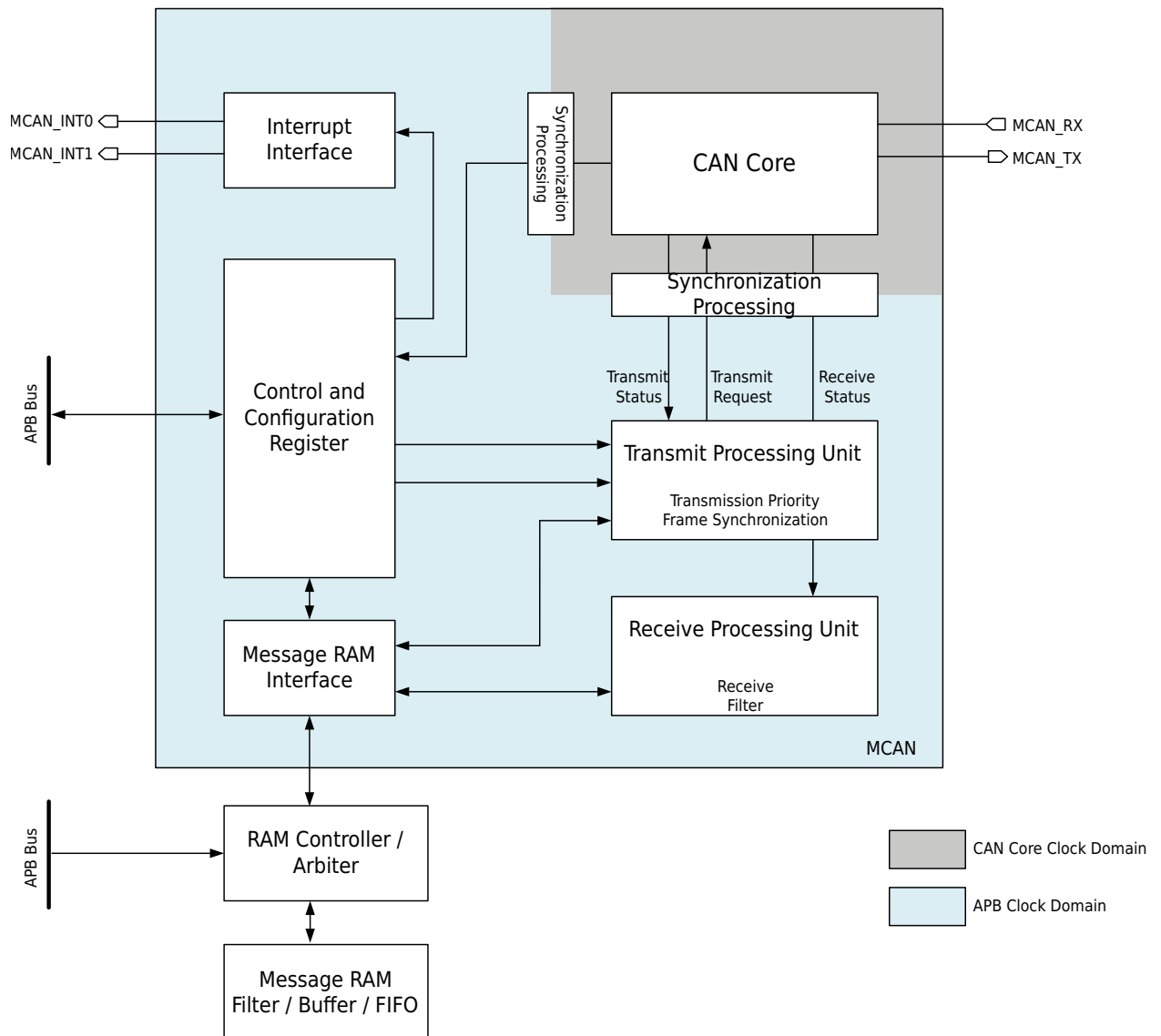


Figure 37-1 MCAN System Block Diagram

#### CAN Core

The CAN core includes a protocol controller and transmit/receive shift registers. It supports all protocol functions defined in ISO 11898-1:2015 and allows for both 11-bit standard IDs and 29-bit extended IDs.

#### Synchronization

The synchronization module synchronizes the signal from the APB clock domain to the CAN core clock domain, and synchronizes the signal from the CAN core clock domain to the APB clock domain.

The APB clock domain provides the MCAN control logic clock, while the CAN core clock domain provides the MCAN communication clock. For details, refer to the "Working Clock Specifications" section in the "Clock Module Unit (CMU)" chapter. To ensure stable MCAN communication, the control logic clock frequency must be greater than or equal to the communication clock frequency.

## Transmit Processing Unit

This unit controls the transfer of messages from the external message RAM to the CAN core. Up to 32 transmit buffers can be configured for message transmission. Transmit buffers can be used as dedicated transmit buffers, a transmit FIFO (or queue), or a combination of both. A Transmit Event FIFO stores the transmission timestamp along with the corresponding message ID. Transmission cancellation is also supported.

## Receive Processing Unit

This unit controls the transfer of received messages from the CAN core to the external message RAM. It supports two configurable receive FIFOs and up to 64 dedicated receive buffers for storing messages that pass the receive filters. Unlike receive FIFOs, dedicated receive buffers are only used to store messages with specific IDs. Each received message is stored together with its receive timestamp. Up to 128 filters can be defined for 11-bit standard IDs, and up to 64 filters for 29-bit extended IDs.

## APB Interface

Connects the MCAN to the APB bus.

## Message RAM

The interface connects MCAN access to an external 4KB message RAM through a RAM controller and arbiter.

## Dual Interrupt Lines

MCAN provides two interrupt lines: MCAN\_INT0 and MCAN\_INT1. By programming the MCAN\_ILE.EINT0 and MCAN\_ILE.EINT1 bits, these two interrupt lines can be enabled or disabled independently.

## 37.3.2 Pin Description

Table 37-1 MCAN Pin Description

| Pin Name | Direction | Description       |
|----------|-----------|-------------------|
| MCAN_RX  | Input     | MCAN receive pin  |
| MCAN_TX  | Output    | MCAN transmit pin |

## 37.3.3 Working Mode

### 37.3.3.1 Software Initialization

Software initialization is started by setting MCAN\_CCCR.INIT. This bit is also set by a software reset, hardware reset, or upon entering Bus\_Off state. When MCAN\_CCCR.INIT is set, both incoming and outgoing messages on the CAN bus stop and the state of the MCAN\_TX pin becomes recessive (high). The counters of the Error Management Logic (EML) remain unchanged. Setting MCAN\_CCCR.INIT does not change any configuration registers. Clearing MCAN\_CCCR.INIT to zero completes software initialization, after which the bit stream processing unit (BSP) waits for a sequence of 11 consecutive recessive bits to appear on the bus (Bus\_Idle) to synchronize itself with the data transmission on the CAN bus before participating in bus activities and starting message transmission.

The MCAN configuration registers can be accessed only when both MCAN\_CCCR.INIT and MCAN\_CCCR.CCE are set. The MCAN configuration registers are shown in the following table.

Table 37-2 MCAN Configuration Register List

| Register   | Description                                                      |
|------------|------------------------------------------------------------------|
| MCAN_DBTP  | MCAN Data Bit Time and Prescaler Register                        |
| MCAN_TEST  | MCAN Test Register                                               |
| MCAN_RWD   | MCAN RAM Watchdog Register                                       |
| MCAN_CCCR  | MCAN CC Control Register                                         |
| MCAN_NBTP  | MCAN Nominal Bit Time and Prescaler Register                     |
| MCAN_TSCC  | MCAN Timestamp Counter Configuration Register                    |
| MCAN_TOCC  | MCAN Timeout Counter Configuration Register                      |
| MCAN_TDCR  | MCAN Transmitter Delay Compensation Register                     |
| MCAN_GFC   | MCAN Global Filter Configuration Register                        |
| MCAN_SIDFC | MCAN Standard ID Filter Configuration Register                   |
| MCAN_XIDFC | MCAN Extended ID Filter Configuration Register                   |
| MCAN_XIDAM | MCAN Extended ID and Mask Register                               |
| MCAN_RXF0C | MCAN Receive FIFO0 Configuration Register                        |
| MCAN_RXBC  | MCAN Receive Buffer Configuration Register                       |
| MCAN_RXF1C | MCAN Receive FIFO1 Configuration Register                        |
| MCAN_RXESC | MCAN Receive Buffer and FIFO Element Size Configuration Register |
| MCAN_TXBC  | MCAN Transmit Buffer Configuration Register                      |
| MCAN_TXESC | MCAN Transmit Buffer Element Size Configuration Register         |
| MCAN_TXEFC | MCAN Transmit Event FIFO Configuration Register                  |

MCAN\_CCCR.CCE can be set or cleared only when MCAN\_CCCR.INIT is set, and is automatically cleared when MCAN\_CCCR.INIT is cleared.

When MCAN\_CCCR.CCE is set, the following registers will be reset:

- MCAN\_HPMS: High priority message status
- MCAN\_RXF0S: Receive FIFO 0 status
- MCAN\_RXF1S: Receive FIFO 1 status
- MCAN\_TXFQS: Transmit FIFO/queue status
- MCAN\_TXBRP: Transmit buffer request pending
- MCAN\_TXBTO: Transmit buffer transmission occurs
- MCAN\_TXBCF: Transmit buffer cancel completed
- MCAN\_TXEFS: Transmit event FIFO status
- MCAN\_TXBAR: Transmit buffer add request
- MCAN\_TXBCR: Transmit buffer cancel request

The host can set MCAN\_CCCR.TEST and MCAN\_CCCR.MON only when MCAN\_CCCR.INIT and MCAN\_CCCR.CCE are both set. The host can reset MCAN\_CCCR.TEST and MCAN\_CCCR.MON at any time. MCAN\_CCCR.DAR can be set or cleared only when both MCAN\_CCCR.INIT and MCAN\_CCCR.CCE are set.

### 37.3.3.2 Normal Operation

After the MCAN has completed initialization and the MCAN\_CCCR.INIT bit is cleared, the MCAN synchronizes itself with the CAN bus and becomes ready for communication.

When receiving messages, those that pass the receive filters (including message ID and DLC) are stored in a dedicated receive buffer, receive FIFO0, or receive FIFO1.

When transmitting messages, the dedicated transmit buffers and/or the transmit FIFO (queue) must be initialized or updated first. Automatic transmission in response to a received remote frame is not supported.

### 37.3.3.3 CAN FD Operation

There are two types of CAN FD frame transmission. The first is a CAN FD frame without bit rate switching. The second is a CAN FD frame with bit rate switching, in which the control field, data field and CRC field are sent at a higher bit rate than the start and end of the frame.

The r0 bit in the Classical CAN standard frame (after the IDE bit, a reserved bit in the standard frame format) and the r1 bit in the extended frame (after the RTR bit, the first reserved bit in the extended frame format) are decoded as the FDF bit in the CAN FD frame format. The FDF bit is recessive (logic 1) to indicate a CAN FD frame and dominant (logic 0) to indicate a Classic CAN frame. In the CAN FD frame, the two bits res and BRS (Bit Rate Switch) after the FDF bit determine whether to switch the bit rate in the CAN FD frame, and indicate that the bit rate has been switched through res explicit and BRS implicit. The implicit encoding of the res bit is reserved for future protocol extensions. If the FDF bit and the res bit in the frame received by MCAN are both recessive, a protocol exception event will be issued by setting MCAN\_PSR.PXE. If protocol exception handling is enabled (MCAN\_CCCR.PXHD=0), the MCAN will change the operating state from receiving (MCAN\_PSR.ACT=0b10) to synchronizing (MCAN\_PSR.ACT=0b00) at the next sampling point. If protocol exception handling is disabled (MCAN\_CCCR.PXHD=1), the MCAN will treat the recessive res bit as a format error and will respond with an error frame.

CAN FD mode is enabled by programming MCAN\_CCCR.FDOE. If MCAN\_CCCR.FDOE=1, the sending and receiving of CAN FD frames will be enabled. Classic CAN frames can always be sent and received. Whether a CAN FD frame or a Classical CAN frame is transmitted is determined by the FDF bit in the respective transmit buffer element. If MCAN\_CCCR.FDOE=0, all received frames are treated as Classical CAN frames. As a result, reception of a CAN FD frame will trigger an error frame. If CAN FD mode is disabled, even if the FDF bit in the transmit buffer element is set, a CAN FD frame will not be transmitted. The MCAN\_CCCR.FDOE and MCAN\_CCCR.BRSE bits can only be modified when both MCAN\_CCCR.INIT and MCAN\_CCCR.CCE are set.

If MCAN\_CCCR.FDOE=0, the setting of the FDF and BRS bits are ignored, and the frame is transmitted in Classical CAN format. If MCAN\_CCCR.FDOE=1 and MCAN\_CCCR.BRSE=0, only the FDF bit in the transmit buffer is evaluated. If MCAN\_CCCR.FDOE=1 and MCAN\_CCCR.BRSE=1, transmission of CAN FD frames with bit rate switching is enabled. In this case, all transmit buffer elements with both FDF and BRS bits set will be transmitted as CAN FD frames with bit rate switching.

It is recommended to switch modes during CAN operation only under the following condition:

- The failure rate in the CAN FD data phase is significantly higher than the failure rate in the CAN FD arbitration phase. In this case, bit rate switching for CAN FD transmission should be disabled.
- During system startup, all nodes transmit classical CAN messages until communication using CAN FD format is verified. Once verification is complete, all nodes switch to CAN FD operation.
- In CAN Partial Networking, wake-up messages must be transmitted using Classical CAN format.

- During EOL programming (End-of-line programming), not all nodes support CANFD, and non-CANFD nodes will remain in silent mode until the end of programming. Afterward, all nodes switch back to Classical CAN communication.

In the CAN FD format, DLC codes 0 to 8 have the same encoding (i.e., corresponding data byte lengths) as in Classical CAN, while DLC codes 9 to 15 differ from Classical CAN, as shown in the table below.

Table 37-3 DLC Encoding in MCAN

| DLC Code        |               | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9  | 10 | 11 | 12 | 13 | 14 | 15 |
|-----------------|---------------|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|
| Data Byte Count | Classical CAN | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 8  | 8  | 8  | 8  | 8  | 8  | 8  |
|                 | CAN FD        | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 12 | 16 | 20 | 24 | 32 | 48 | 64 |

In CAN FD frames, if the BRS bit is recessive, the bit timing will switch within the frame after the BRS (bit rate switching) bit. Before the BRS bit, during the CAN FD arbitration phase, the bit timing follows the definition in the MCAN\_NBTP (Nominal Bit Timing & Prescaler Register). During the following CAN FD data phase, the bit timing follows the definition in the MCAN\_DBTP (Data Bit Timing & Prescaler Register). The bit timing will switch back to the data phase bit timing at the CRC delimiter or when an error is detected, whichever occurs first.

The maximum configurable bit rate in the CAN FD data phase depends on the CAN communication clock frequency. For example, if the CAN communication clock frequency is 20 MHz, the shortest configurable bit time is 4 time quanta (tq), which results in a data phase bit rate of 5Mbps.

In both CAN FD frames with bit rate switching and CAN FD frames without bit rate switching, the ESI (Error Status Indicator) bit is set based on the error status of the transmitting node at the time of transmission. If the transmitting node is in error passive state, the ESI bit is transmitted recessive. Otherwise, it is transmitted dominant.

#### 37.3.3.4 Transmit Delay Compensation

In the CAN FD data phase, only one node transmits, while all other nodes are receivers. The length of the bus does not affect this. When transmitting through the MCAN\_TX pin, the MCAN receives the transmitted data from its local CAN transceiver via the MCAN\_RX pin. The received data is delayed due to the transmission delay, and if the delay exceeds TSEG1 (the time segment before the sample point), a bit error will be detected. To allow the data phase bit time to be shorter than the transmission delay, delay compensation is introduced. Without delay compensation, the bit rate in the CAN FD frame data phase would be limited by the transmission delay.

The MCAN protocol unit implements a delay compensation mechanism to compensate for the transmission delay, enabling higher bit rates during the CAN FD data phase without being affected by the specific CAN transceiver's delay.

To check for bit errors in the data phase of the transmitting node, the delayed transmitted data is compared with the received data at the second sample point (SSP). If a bit error is detected, the transmitting node will react to the bit error at the next regular sample point. In the arbitration phase, delay compensation is always disabled.

Transmit delay compensation supports configurations where the data bit time is shorter than the transmission delay. ISO 11898-1:2015 provides a detailed description of this mechanism, which is enabled by setting MCAN\_DBTP.TDC.

At the second sample point (SSP), the received bit is compared with the transmitted bit. The position of the SSP is defined as the delay from the MCAN\_TX pin through the transceiver to the MCAN\_RX pin, added to the transmission delay compensation offset configured by MCAN\_TDCR.TDCO. The transmission delay compensation offset is used to adjust the position of the SSP within the received bit (for example, half the bit time in the data phase). The position of the second sample point is rounded down to the nearest mtq (Minimum Time Quantum, which is one MCAN communication clock period).

The value of the actual transmission delay compensation is displayed in MCAN\_PSR.TDCV. When MCAN\_CCCR.INIT is set, MCAN\_PSR.TDCV is cleared; When MCAN\_DBTP.TDC is set, MCAN\_PSR.TDCV is updated each time an FD frame is transmitted.

For transmit delay compensation implemented in MCAN, the following boundary conditions must be considered:

- The measured delay from MCAN\_TX to MCAN\_RX, plus the configured transmission delay compensation offset (MCAN\_TDCR.TDCO), must be less than 6 bit times in the data phase.
- The measured delay from MCAN\_TX to MCAN\_RX, plus the configured transmission delay compensation offset (MCAN\_TDCR.TDCO), must be less than or equal to 127 mtq. If this sum exceeds 127 mtq, the transmit delay compensation uses the maximum value of 127 mtq.
- The data phase ends at the CRC delimiter sample point, after which the received bits will not be checked at the SSP.

If MCAN\_DBTP.TDC=1 is programmed to enable transmit delay compensation, measurement starts at the falling edge between the FDF bit and res bit of each CAN FD frame. When this edge is detected at the transmitting node's MCAN\_RX pin, the measurement stops. The resolution of this measurement is one mtq.

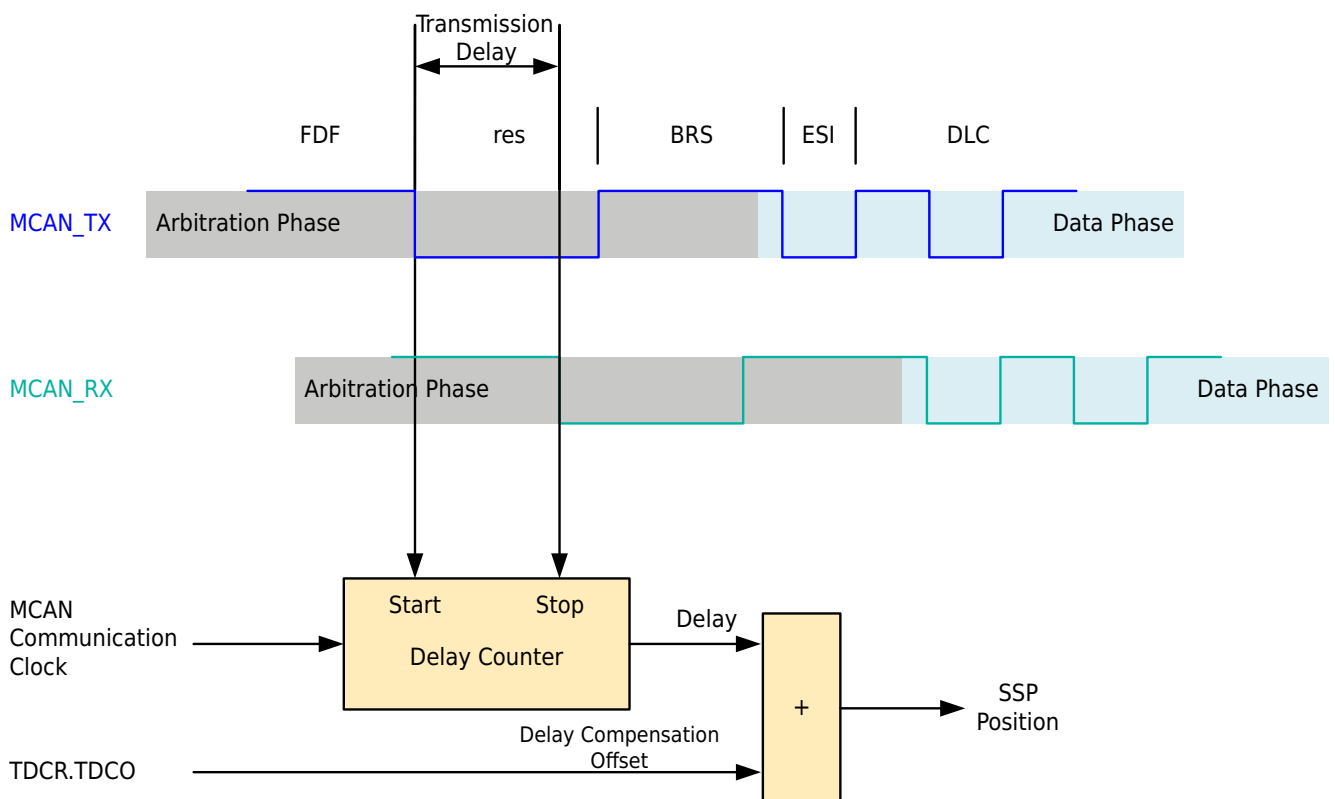


Figure 37-2 Transceiver Delay Measurement

To avoid dominant glitches in the received FDF bit, the delay compensation measurement should end before the falling edge of the received res bit, which would otherwise cause the SSP position to



advance prematurely. This can be achieved by programming MCAN\_TDCR.TDCF to enable a transmit delay compensation filtering window, which defines the minimum value of the SSP position. A dominant edge on the MCAN\_RX pin will cause the SSP position to advance further, but this will be ignored in the context of the transmit delay measurement. Measurement stops when the SSP position is at least MCAN\_TDCR.TDCF and the MCAN\_RX pin is at a low level.

**Note**

The transmit delay used by MCAN is a real-time measured value, and it does not support configuring a fixed transmit delay.

### 37.3.3.5 Restricted Working Mode

In restricted working mode, the node is able to receive data frames and remote frames, and can acknowledge active frames, but it will not send data frames, remote frames, active error frames, or overload frames. In the event of an error or overload condition, the node will not send a dominant bit but will wait for the bus idle condition to re-synchronize with the CAN communication. When error recording occurs (MCAN\_TDCR.CEL), the error counters (MCAN\_TDCR.REC, MCAN\_TDCR.TEC) are frozen. The host can set the MCAN\_CCCR.ASM bit to configure MCAN to restricted working mode, but this can only be done when both MCAN\_CCCR.CCE and MCAN\_CCCR.INIT are set to 1. The MCAN\_CCCR.ASM bit can be cleared at any time by the host.

If the transmit processing unit is unable to read data from the message RAM in time, the node will automatically enter restricted working mode. To exit restricted working mode, the host CPU must clear the MCAN\_CCCR.ASM bit.

Restricted working mode can be used to accommodate applications with different CAN bit rates. In this case, the application will test different bit rates and exit restricted working mode after receiving a active frame.

**Note**

Restricted working mode should not be used in conjunction with loopback mode (either internal or external).

### 37.3.3.6 Bus Monitoring Mode

By setting MCAN\_CCCR.MON to 1, the MCAN is configured to bus monitoring mode. In bus monitoring mode (see ISO 11898-1:2015, 10.14 Bus Monitoring), MCAN is able to receive active data frames and remote frames, but it cannot initiate transmission. In this mode, MCAN sends dominant bits on the CAN bus. If MCAN needs to send a recessive bit (such as the ACK bit, overload flag, or active error flag), that bit is internally rerouted to allow MCAN to listen to the recessive bit, while the CAN bus remains in the dominant state. In bus monitoring mode, the MCAN\_TXBRP register remains in a reset state.

Bus monitoring mode can be used to analyze traffic on the CAN bus without causing interference on the bus by transmitting recessive bits. The following figure shows the connection between MCAN\_TX and MCAN\_RX signals and MCAN in bus monitoring mode.



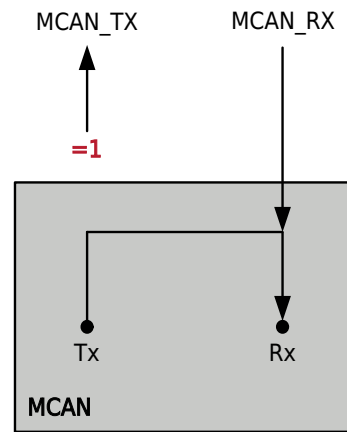


Figure 37-3 Pin Control in Bus Monitoring Mode

### 37.3.3.7 Disable Automatic Retransmission (DAR) Mode

According to the CAN specification (see ISO 11898-1:2015, 8.3.4 Recovery Management), MCAN can automatically retransmit frames that fail arbitration or are disturbed during transmission. By default, automatic retransmission is enabled.

In DAR mode, all transmissions that are initiated on the CAN bus will be automatically canceled. The transmission request pending bit in the transmission buffer MCAN\_TXBRP.TRPx will be cleared after a successful transmission. If the transmission is canceled before it begins, due to arbitration failure, or if an error occurs during the frame transmission, the bit will also be reset.

- Successful transmission:
  - ▶ The corresponding transmission buffer transmission occurred bit MCAN\_TXBTO.TOx is set.
  - ▶ The corresponding transmission buffer cancellation completed bit MCAN\_TXBCF.CFx is not set.
- Transmission canceled but still successful:
  - ▶ The corresponding transmission buffer transmission occurred bit MCAN\_TXBTO.TOx is set.
  - ▶ The corresponding transmission buffer cancellation completed bit MCAN\_TXBCF.CFx is set.
- Arbitration failure or frame transmission disturbed:
  - ▶ The corresponding transmission buffer transmission occurred bit MCAN\_TXBTO.TOx is not set.
  - ▶ The corresponding transmission buffer cancellation completed bit MCAN\_TXBCF.CFx is set.

If the frame transmission is successful and event storage for transmission is enabled, the transmission event will be written to the transmission event FIFO element, where the event type ET=0b10 (indicating transmission even if canceled).

### 37.3.3.8 Power-Down (Sleep Mode)

MCAN can be set to Power-down mode through the CC Control Register MCAN\_CCCR.CSR. As long as the clock stop request is active, the MCAN\_CCCR.CSR bit will read as 1.

After all pending transmission requests are completed, MCAN will wait until the bus idle state is detected. Once the bus idle state is detected, MCAN will set MCAN\_CCCR.INIT to 1 to prevent any further CAN transmission. Then, MCAN will confirm that it is ready to enter Power-down mode by setting MCAN\_CCCR.CSA to 1. Before the module clock (both control logic clock and communication clock) is turned off, the registers are still accessible, but MCAN\_CCCR.INIT remains set to 1.

**Note**

In cases of severe interference on the CAN bus, the bus may never reach the idle state, and therefore MCAN will not set MCAN\_CCCR.INIT. This situation can be detected by polling the MCAN\_PSR.ACT bit. If MCAN does not enter the idle state, software can write CCCR.INIT=1, which will immediately stop MCAN's CAN communication, regardless of whether transmission or reception is in progress.

To exit Power-down mode, the application must first enable the module clock before resetting the CC control register flag MCAN\_CCCR.CSR. MCAN will confirm that the clock is enabled by resetting MCAN\_CCCR.CSA. Afterward, the application can restart CAN communication by resetting MCAN\_CCCR.INIT.

### 37.3.3.9 Test Pattern

To enable write access to the MCAN Test Register MCAN\_TEST (refer to "[MCAN Test Register \(MCAN\\_TEST\)](#)"), the MCAN\_CCCR.TEST bit must be set to 1, allowing the configuration of test modes and test functions.

By programming MCAN\_TEST.TX, the MCAN\_TX pin can perform four output functions. In addition to the default serial data output function, it can also drive the CAN sampling point signal to monitor the bit timing of the MCAN and output a constant dominant or recessive level. The actual value of the MCAN\_RX pin can be read from MCAN\_TEST.RX. These two functions (MCAN\_TEST.TX and MCAN\_TEST.RX) can be used to check the physical layer of the CAN bus.

Due to the synchronization mechanism between the CAN communication clock and the control logic clock, there may be a delay of several control logic clock periods from configuring MCAN\_TEST.TX to implementing the specified function on the MCAN\_TX pin. This also applies to reading the value of MCAN\_RX through MCAN\_TEST.RX.

**Note**

Test mode is intended for production testing and self-diagnostics only. Software control of the MCAN\_TX pin will interfere with all CAN protocol functions. Test mode is not recommended for use in actual applications.

### 37.3.3.10 Loop Mode

#### External Loopback Mode

To set MCAN to external loopback mode, configure MCAN\_TEST.LBCK to 1. In this mode, MCAN processes its own transmitted messages as received messages, and stores them in the receive buffer or receive FIFO, provided these messages pass the receive filtering. In external loopback mode, the MCAN\_TX signal is internally fed back to the receive processing unit. The actual value at the MCAN\_RX input pin is ignored.

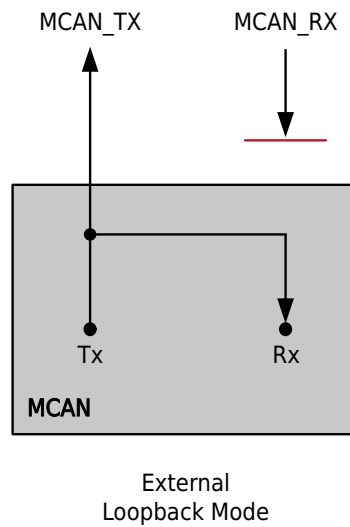


Figure 37-4 External Loopback Mode Pin Control

This mode is used for hardware self-test. To prevent external influences, MCAN ignores ACK errors in loopback mode (e.g., when a recessive bit is sampled in the ACK field of a data or remote frame). In this mode, the signal output by MCAN\_TX is directly fed back to the receiving processing unit internally, and the actual value of the MCAN\_RX input pin is ignored. The sent message can be monitored from the pin MCAN\_TX.

#### Internal Loopback Mode

To set MCAN to internal loopback mode, configure both MCAN\_TEST.LBCK and MCAN\_CCCR.MON to 1. This mode is also referred to as "Hot Selftest," meaning MCAN can be tested without affecting the ongoing CAN system. In this mode, the MCAN\_RX pin is disconnected from MCAN, and the MCAN\_TX pin remains recessive (dominant). In this mode, the MCAN\_RX pin is disconnected from MCAN, and the MCAN\_TX pin remains recessive (dominant).

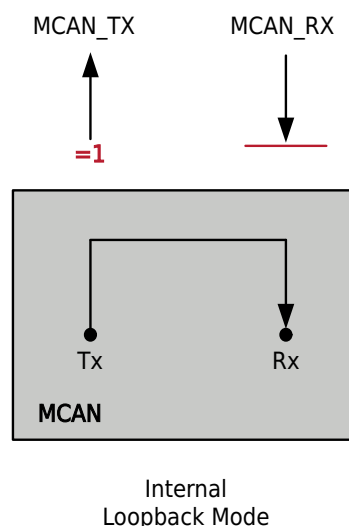


Figure 37-5 Pin Control for Internal Loopback Mode

### 37.3.4 Timestamp Generation

MCAN provides a 16-bit cyclic counter to generate internal timestamps. The counting time unit can be configured to 1-16 times the CAN bit time through the prescaler MCAN\_TSCC.TCP. The counter value can

be read through MCAN\_TSCV.TCV. A write access to the register MCAN\_TSCV resets the counter to 0. The interrupt flag MCAN\_IR.TSW is set when the timestamp counter periods.

When SOF (Start Of Frame) is received or sent, the count value is captured and stored in the timestamp part of the receive buffer or receive FIFO (RXTS[15:0]) or transmit event FIFO (TXTS[15:0]) element.

### 37.3.5 Timeout Counter

The MCAN provides a 16-bit timeout counter to indicate timeout conditions for receive FIFO0, receive FIFO1 and transmit event FIFO. It operates as a down counter and, like the timestamp counter, uses a prescaler controlled by MCAN\_TSCC.TCP. The timeout counter is configured via register MCAN\_TOCC. The actual counter value can be read from MCAN\_TOCV.TOC. The timeout counter is started only when MCAN\_CCCR.INIT=0. The timeout counter stops counting when MCAN\_CCCR.INIT=1, e.g. when the CAN enters the Bus\_Off state.

The operating mode is selected via MCAN\_TOCC.TOS. When operating in continuous mode, the counter starts when MCAN\_CCCR.INIT is reset. Writing to MCAN\_TOCV presets the counter to the value configured by MCAN\_TOCC.TOP and continues counting down.

When the timeout counter is controlled by one of the FIFOs, an empty FIFO presets the counter to the value configured by MCAN\_TOCC.TOP. After the first FIFO element is stored, it starts counting down. Writing to MCAN\_TOCV has no effect.

When the counter reaches 0, the interrupt flag MCAN\_IR.TOO is set. In continuous mode, the counter restarts immediately after reaching MCAN\_TOCC.TOP.



#### Note

The clock signal for the timeout counter comes from the CAN core sample point signal. Therefore, the time at which the timeout counter is decremented may differ due to the CAN core synchronization (resynchronization) mechanism. If the bit rate switching feature is used in CAN FD, the timeout counter counts with different clocks in the Arbitration phase and the Data phase.

### 37.3.6 Reception Processing

The receive processing unit controls receive filtering, transfer of received messages to the receive buffer or one of the two receive FIFOs, and receive FIFO put and get indexes.

#### 37.3.6.1 Receive Filtering

MCAN is able to configure two sets of receive filters, one for standard IDs and one for extended IDs. These two sets of filters can be assigned to the receive buffer, receive FIFO0 or receive FIFO1. For receive filtering, each filter list is compared starting from the 0th element until the first matching element. Receive filtering stops at the first matching element. The remaining filter elements are not evaluated for this message.

- Key features include:
  - ▶ Each filter element can be configured as
    - ◆ ID range filter (start and end range)
    - ◆ Filters for one or two dedicated IDs
    - ◆ Classic filter with ID bit masking
  - ▶ Each filter element can be configured to implement receive filtering or reject filtering
  - ▶ Each filter element can be enabled/disabled individually
  - ▶ Filter elements are compared in order and execution stops after the first matching filter element

- Related configuration register bits:
  - ▶ Global filter configuration (MCAN\_GFC)
  - ▶ Standard ID filter configuration (MCAN\_SIDFC)
  - ▶ Extended ID filter configuration (MCAN\_XIDFC)
  - ▶ Extended ID and operation mask (MCAN\_XIDAM)
- Depending on the configuration of the filter element (SFEC/EFEC), one of the following actions is triggered when a match occurs:
  - ▶ Store the received frame in receive FIFO0 or FIFO1
  - ▶ Store the received frame in the receive buffer
  - ▶ Reject the received frame
  - ▶ Set the high priority message interrupt flag MCAN\_IR.HPM to 1
  - ▶ Set the high priority message interrupt flag MCAN\_IR.HPM to 1 and store the received frame in receive FIFO0 or FIFO1

After receiving the complete ID, receive filtering begins. After the reception filtering is completed, if a matching reception buffer or reception FIFO is found, the message processing unit will start writing the received message data in the form of 32 bits into the matching reception buffer or reception FIFO. If the CAN protocol controller detects an error condition (e.g. CRC error), the message is discarded with the following effects:

- Receive buffer
 

The new data flag of the matching receive buffer is not set, but the receive buffer is (partially) overwritten with the received data. For error types, see MCAN\_PSR.LEC and MCAN\_PSR.DLEC.
- Receive FIFO
 

The put index of the matching receive FIFO is not updated, but the associated receive FIFO element is (partially) overwritten with the received data. For error types, see MCAN\_PSR.LEC and MCAN\_PSR.DLEC. Receive FIFO If the matching receive FIFO operates in overwrite mode, the boundary conditions described in Receive FIFO overwrite mode must be taken into account.



#### Note

When the received message is written to one of the two receive FIFOs or to the receive buffer, the unmatched received ID is stored independently of the used filter. The process of receive filtering depends strongly on the order of the configured filter elements.

### 37.3.6.2 ID Range Filter

This filter checks all received frames for a match with the message IDs in the range defined by SF1ID/SF2ID and EF1ID/EF2ID. When range filtering is used with extended frames, there are two possible cases:

- EFT=0b00: The message ID of the received frame is ANDed with the extended ID and the operation mask (MCAN\_XIDAM) before the range filter is applied
- EFT=0b11: Do not use the extended ID and mask (MCAN\_XIDAM) for range filtering

### 37.3.6.3 Dedicated ID Filter

The filter element can be configured to filter on one or two specific message IDs. To filter a specific message ID, the filter element must be configured with SF1ID=SF2ID and EF1ID=EF2ID.

### 37.3.6.4 ID Bit Mask Filter

ID bit mask filtering is used to filter groups of message IDs by masking certain bits of the received message ID. When performing ID bit mask filtering, SF1ID/EF1ID is used as a message ID filter and SF2ID/EF2ID is used as a filter ID mask.

Bits that are 0 in the ID mask mask the corresponding bits of the configured ID filter, i.e., the value of the received message ID at that bit position is not compared with the value of the corresponding bit of the received filter, only the bits of the received message ID for which the corresponding mask bit is 1 are compared.

If all mask bits are 1, a match occurs only if the received message ID and the message ID filter are exactly the same. If all mask bits are 0, all message IDs match.

### 37.3.6.5 Standard Message ID Filtering

The following diagram illustrates the filtering process for a standard message ID (11-bit ID). "Standard Message ID Filter Element" describes the standard message ID filter element.

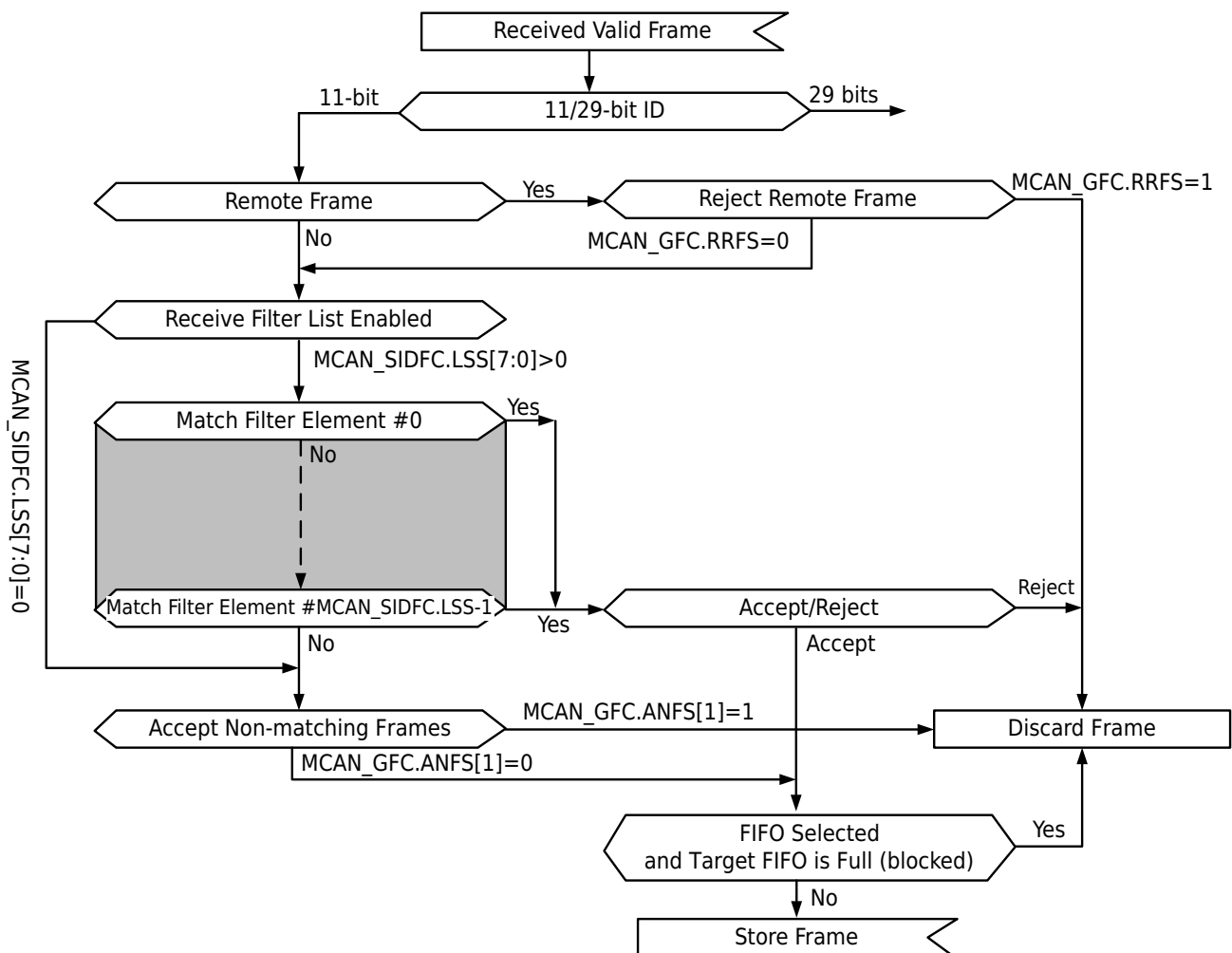


Figure 37-6 Standard Message ID Filter Path

Under the control of the global filter configuration register (MCAN\_GFC) and the standard ID filter configuration register (MCAN\_SIDFC), the ID, remote transmission request bit (RTR), and ID extension bit (IDE) of the received message are compared to the configured list of filter elements.

### 37.3.6.6 Extended Message ID Filtering

The following figure describes the filtering process for extended message IDs (29-bit IDs). "Extended Message ID Filter Elements" describes the extended message ID filter elements.

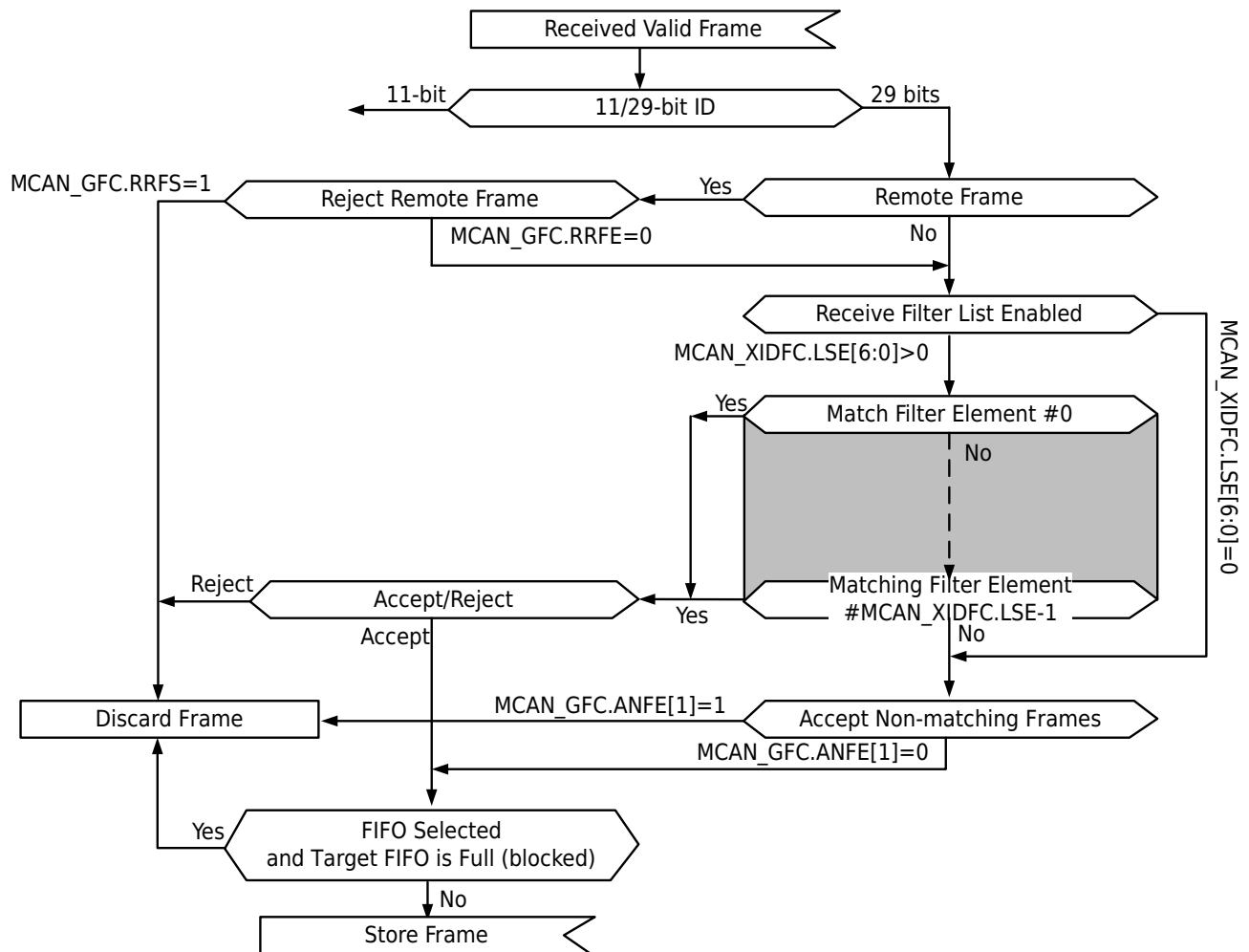


Figure 37-7 Extended Message ID Filter Path

The received message ID, remote transmission request bit (RTR), and ID extension bit (IDE) are compared to the configured filter element list under the control of the global filter configuration register (MCAN\_GFC) and the extended ID filter configuration register (MCAN\_XIDFC). The extended ID and mask (MCAN\_XIDAM) is ANDed with the received ID before executing the filter list.

### 37.3.6.7 Receive FIFO

Receive FIFO0 and receive FIFO1 can store up to 64 elements each (the actual number of elements depends on the capacity of the message RAM). The configuration of the two receive FIFOs is done via registers MCAN\_RXF0C and MCAN\_RXF1C.

Messages that pass the receive filter are transferred to the receive FIFO configured with the matching filter element. For a description of the filter mechanism that can be used for receive FIFO0 and receive FIFO1, see "Receive Filtering". "Receive Buffer and FIFO Elements" describes the receive buffer and FIFO elements.

To avoid receive FIFO overflow, a receive FIFO watermark can be used. When the receive FIFO fill level reaches the receive FIFO watermark configured by MCAN\_RXFnC.FnWM, the interrupt flag MCAN\_IR.RFnW is set. When the put index and get index of the receive FIFO meet, the receive FIFO full condition occurs, the flag bit MCAN\_RXFnS.FnF is set, and the interrupt flag bit MCAN\_IR.RFnF is set.

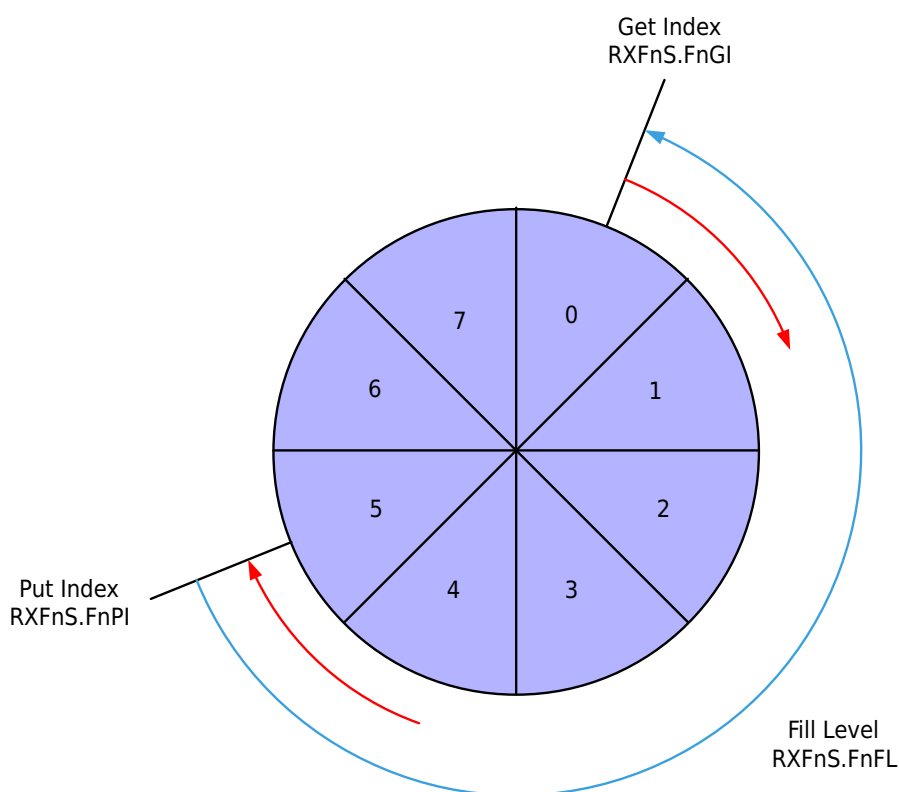


Figure 37-8 Receive FIFO Status

When reading a message from the receive FIFO, the receive FIFO get index `MCAN_RXFnS.FnGI` multiplied by the FIFO element size (words), multiplied by 4, must be added to the start address `MCAN_RXFnC.FnSA`.

Table 37-4 Receive buffer and FIFO element size

| <code>RXESC.RBDS[2:0]</code><br><code>RXESC.FnDS[2:0]</code> | Data Field (Bytes) | Element Size (Words) |
|--------------------------------------------------------------|--------------------|----------------------|
| 0b000                                                        | 8                  | 4                    |
| 0b001                                                        | 12                 | 5                    |
| 0b010                                                        | 16                 | 6                    |
| 0b011                                                        | 20                 | 7                    |
| 0b100                                                        | 24                 | 8                    |
| 0b101                                                        | 32                 | 10                   |
| 0b110                                                        | 48                 | 14                   |
| 0b111                                                        | 64                 | 18                   |

### 37.3.6.8 Receive FIFO Blocking Mode

Receive FIFO blocking mode is configured by `MCAN_RXFnC.FnOM=0`, which is the default operating mode of the receive FIFO.

When the receive FIFO full condition is reached (`MCAN_RXFnS.FnPI=MCAN_RXFnS.FnGI`), the flag bit `RXFnS.FnF=1` and the interrupt flag `MCAN_IR.RFnF=1`. At this time, the MCAN will not continue to write



messages to the corresponding receive FIFO unless the message is read from the corresponding receive FIFO and its fetch index has been incremented.

If a message is received when the corresponding receive FIFO is full, this message will be discarded and the message loss will be indicated by `MCAN_RXFnS.RFnL=1`. In addition, the interrupt flag `MCAN_IR.RFnL` will also be set.

### 37.3.6.9 Receive FIFO Overwrite Mode

Receive FIFO overwrite mode is configured by `MCAN_RXFnC.FnOM=1`.

When the receive FIFO full condition is reached (`MCAN_RXFnS.FnPI=MCAN_RXFnS.FnGI`), the flag bit `RXFnS.FnF=1`. At this time, the next message received by the FIFO will overwrite the oldest received message, and both the put index and the get index will increase by 1.

If the receive FIFO operates in overwrite mode and the receive FIFO is full, the receive FIFO elements should be read from at least the get index plus 1. This is because when the CPU is reading a message from the message RAM, a received message is being written to the message RAM, which may cause the data read by the CPU to be inconsistent with the actual data. Adding an offset to the get index when reading data from the receive FIFO can avoid this problem. The offset depends on the speed at which the CPU accesses the receive FIFO. The figure below shows that the get index offset is 2 when reading the receive FIFO. In this case, the two messages stored in elements 1 and 2 are lost.

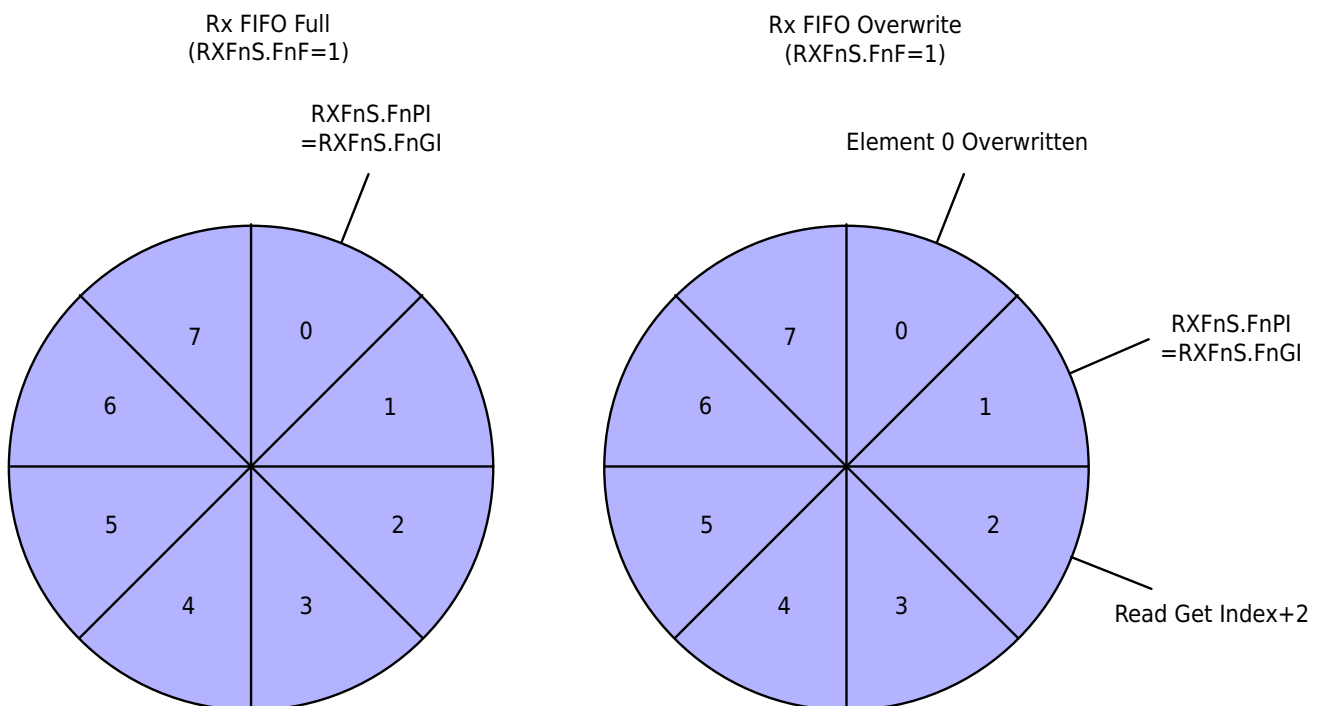


Figure 37-9 Receive FIFO Overflow Processing

After reading data from the receive FIFO, the number of the last element read must be written into the receive FIFO confirmation index register `MCAN_RXFnA.FnA` so that the incremented acquisition index points to the next element. If the put index does not increase to the receive FIFO element number, the receive FIFO full flag will be reset (`MCAN_RXFnS.FnF=0`).

### 37.3.6.10 Dedicated Receive Buffer

The MCAN supports up to 64 dedicated receive buffers. The starting address of the dedicated receive buffer is configured via `RXBC.RBSA`.

To use the dedicated receive buffer, the SFEC bits of the standard message ID filter element must be configured to 0b111 and the SFID2 bits to 0b00 for standard ID messages (see "[Standard Message ID Filter Element](#)"); for extended ID messages, the EFEC bits of the extended message ID filter element must be configured to 0b111 and the EFID2 bits to 0b00 (see "[Extended Message ID Filter Element](#)"). The following table shows an example filter configuration for a dedicated receive buffer.

After a received message is accepted by a filter element, it is stored in the receive buffer in the message RAM referenced by the filter element. The storage format is the same as for the receive FIFO element. In addition, the flag MCAN\_IR.DRX in the interrupt register (message stored in the dedicated receive buffer) is set to 1.

**Table 37-5 Receive Buffer Filter Configuration Example**

| Filter Elements | SFID1[10:0]  | SFID2[10:9] | SFID2[5:0] |
|-----------------|--------------|-------------|------------|
|                 | EFID1[28:0]  | EFID2[10:9] | EFID2[5:0] |
| 0               | ID Message 1 | 0b00        | 0b000000   |
| 1               | ID Message 2 | 0b00        | 0b000001   |
| 2               | ID Message 3 | 0b00        | 0b000010   |

After the last word of the matching received message is written to the message RAM, the corresponding new data flag in register MCAN\_NDAT1/MCAN\_NDAT2 is set. As long as the new data flag remains set, the corresponding receive buffer is locked to prevent it from being updated with a new matching message. The new data flag must be reset by the host writing 1 to the corresponding bit.

When the receive buffer new data flag is set, its corresponding message ID filter element will not perform receive filtering, and other filter elements will continue to perform receive filtering. Other message ID filter elements may cause the received message to be stored in another receive buffer or receive FIFO, or rejected, depending on the filter configuration.

### 37.3.6.11 Receive Buffer Processing

- Reset interrupt flag MCAN\_IR.DRX
- Read new data register
- Read message from message RAM
- Reset new data flag of processed message

### 37.3.6.12 Filtering Debug Messages

To filter debug messages, configure a standard or extended message ID filter element for each of the three debug messages. To enable the filter element to filter debug messages, SFEC/EFEC must be programmed to 0b111. In this case, SFID1/SFID2 and EFID1/EFID2 have different meanings (see "[Standard Message ID Filter Element](#)" and "[Extended Message ID Filter Element](#)"). SFID2/EFID2[10:9] controls the debug message processing state machine, while SFID2/EFID2[5:0] controls the storage location of the received debug message. The following table shows an example of the debug message filter configuration.

When storing debug messages, the corresponding new data flag and MCAN\_IR.DRX are not set, and the reception of debug messages can be monitored through MCAN\_RXF1S.DMS.

Table 37-6 Debug Message Filter Configuration Example

| Filter Elements | SFID1[10:0]<br>EFID1[28:0] | SFID2[10:9]<br>EFID2[10:9] | SFID2[5:0]<br>EFID2[5:0] |
|-----------------|----------------------------|----------------------------|--------------------------|
| 0               | ID Debug Message A         | 0b01                       | 0b111101                 |
| 1               | ID Debug Message B         | 0b10                       | 0b111110                 |
| 2               | ID Debug Message C         | 0b11                       | 0b111111                 |

### 37.3.7 Transmission Processing

The transmit processing unit handles the send requests for the dedicated transmit buffers, the transmit FIFOs and the transmit queues. It controls the transmission of the send messages to the CAN core, puts and gets the index and transmit event FIFO. Up to 32 send buffers can be set for message transmission, each of which can be independently configured with the send mode (Classic CAN mode or CAN FD mode). For a description of the send buffers, see "[Transmit Buffer Elements](#)". The following table describes the possible configurations of the send frame.

Table 37-7 Possible configurations of the send frame

| MCAN_CCCR |      | Transmit Buffer Elements |        | Transmit Frame                |
|-----------|------|--------------------------|--------|-------------------------------|
| BRSE      | FDOE | FDF                      | BRS    |                               |
| Ignore    | 0    | Ignore                   | Ignore | Classic CAN                   |
| 0         | 1    | 0                        | Ignore | Classic CAN                   |
| 0         | 1    | 1                        | Ignore | FD without bit rate switching |
| 1         | 1    | 0                        | Ignore | Classic CAN                   |
| 1         | 1    | 1                        | 0      | FD without bit rate switching |
| 1         | 1    | 1                        | 1      | FD with bit rate switching    |



#### Note

AUTOSAR requires at least three transmit queue buffers and supports transmission cancellation.

When the Transmit Buffer Request Pending Register (MCAN\_TXBRP) is updated, the transmission handler initiates a transmit scan to check for the highest priority pending transmission request (i.e., the transmit buffer with the smallest message ID). The conditions for MCAN\_TXBRP to be updated are that a transmit buffer is sent completely, or the host adds a transmit request by writing MCAN\_TXBAR, or the host cancels a pending transmission by writing MCAN\_TXBCR.

#### 37.3.7.1 Transmit Pause

The Transmit Pause feature is used in CAN systems where CAN message IDs are (permanently) assigned to specific values and cannot be easily changed. These message IDs may have a higher CAN arbitration priority than other defined messages, and in certain applications their relative arbitration priorities should be reversed. This may result in a situation where a burst of multiple CAN messages with a higher arbitration priority from one ECU causes the CAN messages of another ECU with a lower arbitration priority to be delayed.

For example, if CAN ECU-1 has this feature enabled and its application software requests to send four messages, after successfully sending the first message it will wait for two CAN bit times of free

bus time before allowing the next requested message to start. If other ECUs have pending messages, they will start sending these messages during the free time without arbitration with the next message from ECU-1. After receiving a message, ECU-1 can start the next transmission as soon as the received message releases the CAN bus.

This feature is controlled via MCAN\_CCCR.TXP. If this bit is set, the MCAN will pause for two CAN bit times after each successful message before starting the next transmission. This allows messages to be sent even if other nodes in the network have lower priority IDs. This feature is disabled by default (MCAN\_CCCR.TXP=0). This feature allows bursts of multiple messages from a single node to be more sparse, preventing the ECU sending lower priority messages from causing the application to mistakenly believe that it has requested too many transmissions.

### 37.3.7.2 Dedicated Transmit Buffer

Dedicated transmit buffers allow messages to be sent entirely under the control of the host CPU. Each dedicated transmit buffer is configured with a specific message ID. If multiple transmit buffers are configured with the same message ID, the message from the buffer with the lowest buffer number is transmitted first. These buffers should be requested for transmission in ascending order, i.e., by requesting the buffer with the lowest number first, or by issuing a single write to MCAN\_TXBAR to request multiple buffers simultaneously.

If the message in a buffer is updated, transmission can be requested by setting the corresponding bit in MCAN\_TXBAR.ARn. A requested transmission will first undergo internal arbitration against messages in the transmit FIFO or transmit queue, based on message ID, followed by external arbitration on the CAN bus. The message is then transmitted according to the arbitration result.

The size of each dedicated transmit buffer element in the message RAM is a configurable number of 32-bit words(4 bytes). Therefore, the starting address of the dedicated transmit buffer in the message RAM is calculated by multiplying the transmit buffer index (0 to 31) by the element size (words), multiplied by 4 (converted to bytes), and then added to MCAN\_TXBC.TBSA.

Table 37-8 Transmit Buffer, FIFO (queue) Element Size

| TXESC.TBDS2[2:0] | Data Field (bytes) | Element Size (words) |
|------------------|--------------------|----------------------|
| 0b000            | 8                  | 4                    |
| 0b001            | 12                 | 5                    |
| 0b010            | 16                 | 6                    |
| 0b011            | 20                 | 7                    |
| 0b100            | 24                 | 8                    |
| 0b101            | 32                 | 10                   |
| 0b110            | 48                 | 14                   |
| 0b111            | 64                 | 18                   |

### 37.3.7.3 Transmit FIFO

Transmit FIFO mode is enabled by configuring MCAN\_TXBC.TFQM to 0. Messages stored in the transmit FIFO are sent starting from the buffer pointed to by the Get Index (MCAN\_TXFQS.TFGI). After each transmission, the Get Index is incremented in a circular manner until the FIFO becomes empty. The transmit FIFO sends messages with the same ID from different transmit buffers in the order in which

they were written. MCAN calculates the number of available (free) transmit FIFO elements (TXFQS.TFFL) by subtracting the Get Index from the Put Index.

New messages must be written starting from the transmit buffer pointed to by the Put Index (MCAN\_TXFQS.TFQPI). Adding a transmit request increments the Put Index to the next available FIFO element. When the Put Index catches up with the Get Index, the FIFO is considered full (MCAN\_TXFQS.TFQF=1). In this state, no further messages should be written to the FIFO until the next message is transmitted and the Get Index is updated.

If a single message is added to the FIFO, its transmission is triggered by writing 1 to the corresponding bit in MCAN\_TXBAR for the buffer at the Put Index.

If multiple messages (e.g., n messages) are added, they are written to n consecutive transmit buffers starting from the Put Index. Transmission is then requested via MCAN\_TXBAR, and the Put Index is incremented circularly by n. The number of transmit requests must not exceed the number of free FIFO elements indicated by MCAN\_TXFQS.TFFL.

If the transmission request for the buffer at the Get Index is canceled, the Get Index is incremented to the next pending transmit buffer, and the FIFO free level (MCAN\_TXFQS.TFFL) is recalculated. Canceling transmission from any other buffer does not affect the Get Index or the FIFO free level.

Each transmit FIFO element in the Message RAM is allocated a number of 32-bit words (4 bytes each). Thus, the start address of the next available FIFO element is calculated as: (Put Index [MCAN\_TXFQS.TFQPI](0-31) × Element Size in words × 4) + MCAN\_TXBC.TBSA.

#### 37.3.7.4 Transmission Sequence

Transmit Queue mode is enabled by configuring MCAN\_TXBC.TFQM to 1. Messages stored in the transmit queue are sent in order of message ID priority, starting from the smallest ID (i.e., the highest priority). If multiple transmit queue buffers are configured with the same message ID, the send order is determined by the buffer indices used to store the messages, which are influenced by the current state of the Put Index and therefore cannot be predicted.

New messages must be written into the transmit buffer pointed to by index MCAN\_TXFQS.TFQPI. The Put Index always points to the lowest-numbered free buffer within the transmit queue. If the transmit queue is full (MCAN\_TXFQS.TFQF=1), the Put Index becomes inactive, and no new messages should be written to the transmit queue until at least one pending transmission request is completed or canceled.

Alternatively, the application can use the MCAN\_TXBRP register instead of the Put Index to write messages into any transmit buffer that does not have a pending transmission request.

Each transmit queue element in Message RAM occupies a certain number of 32-bit words (4 bytes each). Thus, the start address of the next available transmit queue buffer is calculated as: (Put Index [MCAN\_TXFQS.TFQPI](0-31) × Element Size in words × 4) + MCAN\_TXBC.TBSA.

#### 37.3.7.5 Mixed Use of Dedicated Transmit Buffers and Transmit FIFO

In this configuration, the transmit buffers in the Message RAM are divided into two parts: a group of dedicated transmit buffers and a Transmit FIFO. The number of dedicated transmit buffers is configured via MCAN\_TXBC.NDTB. The number of transmit buffers assigned to the Transmit FIFO is configured via MCAN\_TXBC.TFQS. If MCAN\_TXBC.TFQS is set to 0, only dedicated transmit buffers are used.

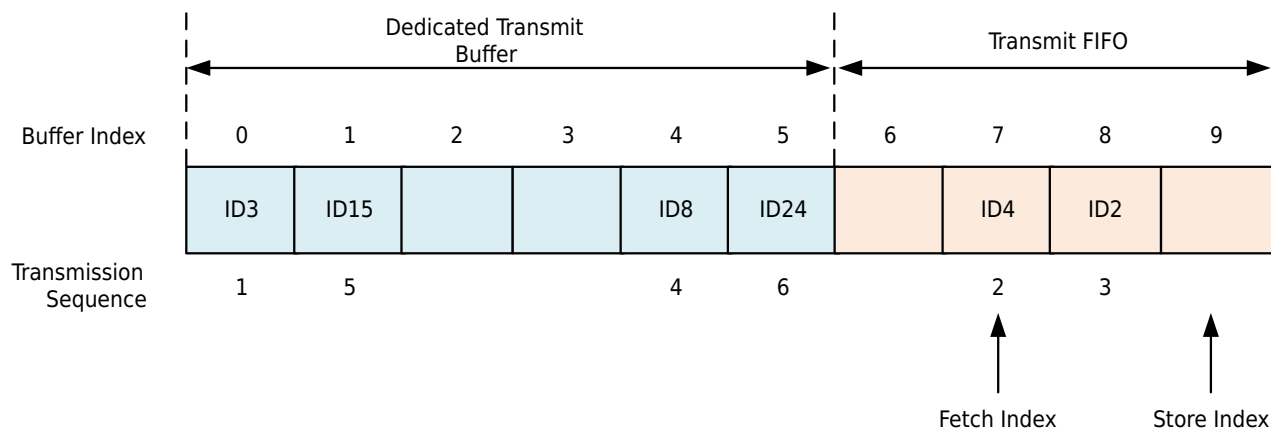


Figure 37-10 Example of Mixed Configuration of Dedicated Transmit Buffers and Transmit FIFO

#### Transmission Priority

- The transmission unit scans all dedicated transmit buffers with active transmission requests, along with the earliest pending request in the Transmit FIFO (i.e., the buffer pointed to by MCAN\_TXFQS.TFGI).
- The message with the lowest message ID has the highest priority and will be selected for transmission next.

#### 37.3.7.6 Mixed Use of Dedicated Transmit Buffers and Transmit Queue

In this configuration, the transmit buffers in the message RAM are divided into two parts: a set of dedicated transmit buffers and a transmit queue. The number of dedicated transmit buffers is configured via MCAN\_TXBC.NDTB. The number of transmit queue buffers is configured via MCAN\_TXBC.TFQS. If MCAN\_TXBC.TFQS is set to 0, only dedicated transmit buffers are used.

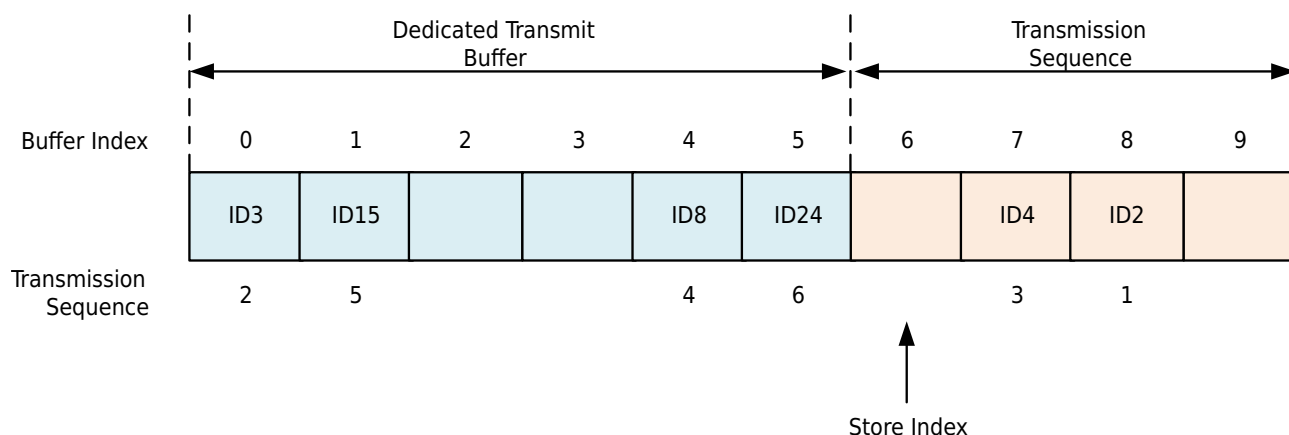


Figure 37-11 Example of Mixed Configuration of Dedicated Transmit Buffers and Transmit Queue

#### Transmission Priority

- The transmission unit scans all dedicated transmit buffers and transmit queue buffers with active transmission requests.
- The buffer with the lowest message ID has the highest priority and its message will be transmitted next.

#### 37.3.7.7 Transmission Cancellation

MCAN supports transmission cancellation, which particularly useful for gateway applications and AUTO-SAR-based systems. To cancel a transmission request from either a dedicated transmit buffer or a

transmit queue buffer, the host must write a 1 to the corresponding bit (based on the buffer index) in the MCAN\_TXBCR register. Transmission cancellation is not supported in transmit FIFO mode.

Upon successful cancellation, the corresponding bit in the MCAN\_TXBCF register is set to 1.

If a cancellation is requested while the buffer is in the process of transmitting, the corresponding bit in the MCAN\_TXBRP register remains set to 1 as long as the transmission is active. If the transmission completes successfully, the respective bits in both MCAN\_TXBTO and MCAN\_TXBCF will be set to 1. If the transmission fails, it will not be retried, and only the corresponding bit in MCAN\_TXBCF is set.

**Note**

In cases where a pending transmission is cancelled before it begins, a short delay window may exist during which no message is transmitted—even if another message is pending in the node. This allows another node on the CAN network to transmit a message, which may have a lower priority than the second message in the original node.

### 37.3.7.8 Transmission Event Processing

MCAN uses a Transmit Event FIFO to support the handling of transmission events. After a message is transmitted on the CAN bus, its message ID and timestamp are stored in a transmit event FIFO element. To associate a transmission event with the respective transmit event FIFO entry, the message marker from the transmitted transmit buffer is copied into the transmit event FIFO element.

The transmit event FIFO can be configured with up to 32 elements. For details, refer to the section "[Transmit Event FIFO Element](#)".

The purpose of the transmit event FIFO is to decouple the processing of transmission status information from the processing of transmission data. This allows transmit buffers to store only the message content, while transmission status is independently tracked in the FIFO. This approach offers clear advantages—especially when working with dynamically managed transmission queues. After a message is successfully transmitted, its corresponding transmit buffer can immediately be reused for a new message.

When MCAN\_IR.TEFF indicates that the transmit event FIFO is full, no further elements are written to the transmit event FIFO until at least one element has been read and the transmit event FIFO get index has been incremented. If a transmit event occurs when the transmit event FIFO is full, the event is discarded and the interrupt flag MCAN\_IR.TEFL is set.

To prevent the transmit event FIFO from overflowing, the transmit event FIFO watermark can be used. When the transmit event FIFO fill count reaches the transmit event FIFO watermark configured by TXEFC.EFWM, the interrupt flag MCAN\_IR.TEFW is set.

When reading data from the transmit event FIFO, the transmit event FIFO get index TXEFS.EFGI must be multiplied by 2 (the transmit event FIFO element size is two words), then multiplied by 4, and then added to the transmit event FIFO start address TXEFC.EFSA to get the address where the target data is located.

### 37.3.8 FIFO Confirmation Processing

The acquisition index of receive FIFO0, receive FIFO1 and transmit event FIFO is controlled by writing to the corresponding FIFO confirmation index register, see "[MCAN Receive FIFO0 Confirmation Register \(MCAN\\_RXF0A\)](#)", "[MCAN Receive FIFO1 Confirmation Register \(MCAN\\_RXF1A\)](#)" and "[MCAN Transmit Event FIFO Confirmation Register \(MCAN\\_TXEFA\)](#)". A write operation to the FIFO acknowledgement index sets



the FIFO get index to the FIFO acknowledgement index plus 1, which in turn updates the FIFO fill level. There are two use cases:

1. If only one element is read from the FIFO (the element pointed to by the get index), the get index value should be written to the FIFO acknowledge index register.
2. If a sequence of elements has been read from the FIFO, the FIFO get index can be updated simply by writing the index value of the last element to the FIFO acknowledge index register at the end of the sequence read.

Since the CPU has free access to the MCAN's message RAM, special care needs to be taken when reading FIFO elements in arbitrary order (regardless of the fetch index), especially when reading high priority messages from one of the two receive FIFOs. In this case, the FIFO acknowledge index register should not be written, otherwise the fetch index will point to the wrong location and the FIFO fill level will be changed, and some older FIFO elements may be lost.

**Note**

The application must ensure that the value written to the FIFO confirmation index register is correct; the MCAN does not check whether the value is correct.

### 37.3.9 Message RAM

#### 37.3.9.1 Message RAM Configuration

The width of the message RAM is 32 bits. The MCAN module can allocate up to 1024 words in the message RAM, as shown in the figure below. In actual applications, it is not required to configure all parts, and there is no restriction on the order between the parts.



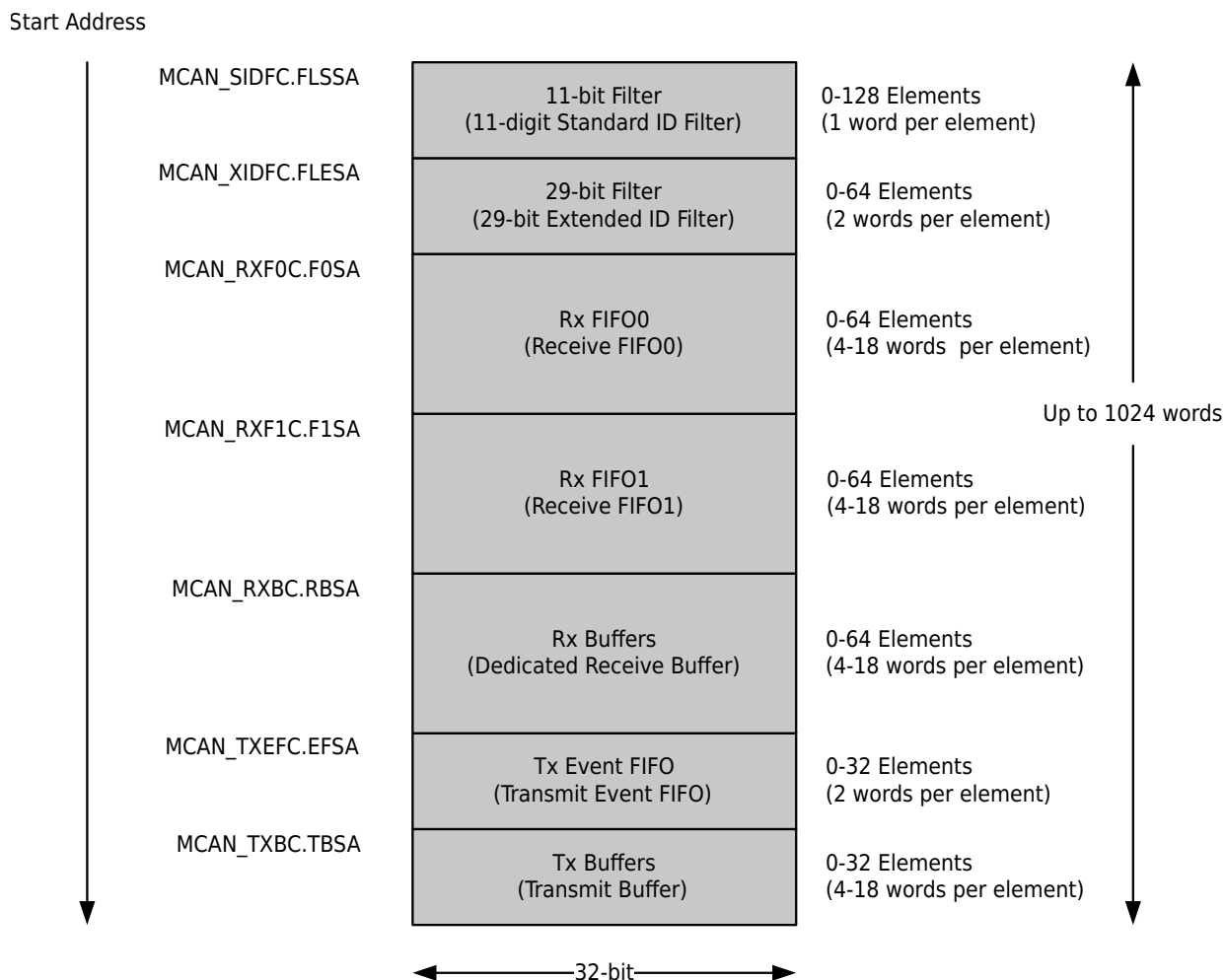


Figure 37-12 Message RAM Configuration

When the MCAN accesses the message RAM, it needs to address it according to 32-bit aligned addresses instead of byte-aligned addresses. The configurable starting address is a 32-bit word address, that is, only 15 to 2 bits of the address are evaluated, and the two least significant bits are ignored.

The message RAM has ECC check (Error Checking and Correcting). The ECC check is a one-correction-two-check code, that is, it can correct 1-bit errors and check 2-bit errors. ECC check has an error injection function. Based on the settings of the error injection enable register MCANRAM\_EIEN and the error injection bit enable registers MCANRAM\_EIBIT0/ MCANRAM\_EIBIT1, the hardware circuitry will invert the data read from RAM bit by bit to achieve error injection. By setting the error injection bit enable registers MCANRAM\_EIBIT0/MCANRAM\_EIBIT1, 1-bit or multi-bit errors can be induced. Since ECC checking is a single error correction and double error detection code, errors exceeding 2-bit may not be detected. The message RAM check error address register (MCANRAM\_ECCERRADDR) records the access address (offset address relative to the message RAM base address) when a 1-bit or 2-bit ECC error occurs in the message RAM.

**Note**

MCAN does not check whether there are errors in the configuration of the message RAM, so you must be careful when configuring the starting addresses of the different parts and the number of elements in each part to avoid data intrusion or loss.

37.3.9.2 Receive Buffers and FIFO Elements

Up to 64 receive buffers and 2 receive FIFOs can be configured in the message RAM. Each receive FIFO can be configured to store up to 64 messages. The structure of the receive buffer and receive FIFO elements and their description are shown in the following table. The element size can be configured via register MCAN\_RXESC. The data field of a CAN FD message can be up to 64 bytes.

Table 37-9 Receive Buffers and FIFO Elements

|       |       |      |     |    |    |    |    |    |       |     |     |     |    |    |    |       |      |    |    |    |    |    |       |   |   |   |   |   |   |   |   |   |
|-------|-------|------|-----|----|----|----|----|----|-------|-----|-----|-----|----|----|----|-------|------|----|----|----|----|----|-------|---|---|---|---|---|---|---|---|---|
| Bit   | 31    | 30   | 29  | 28 | 27 | 26 | 25 | 24 | 23    | 22  | 21  | 20  | 19 | 18 | 17 | 16    | 15   | 14 | 13 | 12 | 11 | 10 | 9     | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| R0    | ESI   | XTD  | RTR | ID |    |    |    |    |       |     |     |     |    |    |    |       |      |    |    |    |    |    |       |   |   |   |   |   |   |   |   |   |
| R1    | ANMF  | FIDX |     |    |    |    |    |    | Res.  | FDF | BRS | DLC |    |    |    |       | RXTS |    |    |    |    |    |       |   |   |   |   |   |   |   |   |   |
| R2    | DB3   |      |     |    |    |    |    |    | DB2   |     |     |     |    |    |    | DB1   |      |    |    |    |    |    | DB0   |   |   |   |   |   |   |   |   |   |
| R3    | DB7   |      |     |    |    |    |    |    | DB6   |     |     |     |    |    |    | DB5   |      |    |    |    |    |    | DB4   |   |   |   |   |   |   |   |   |   |
| ..... | ..... |      |     |    |    |    |    |    | ..... |     |     |     |    |    |    | ..... |      |    |    |    |    |    | ..... |   |   |   |   |   |   |   |   |   |
| Rn    | DBm   |      |     |    |    |    |    |    | DBm-1 |     |     |     |    |    |    | DBm-2 |      |    |    |    |    |    | DBm-3 |   |   |   |   |   |   |   |   |   |

Table 37-10 Receive Buffer and FIFO Elements Description

| Bit    | Name | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|--------|------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| R0[31] | ESI  | Error State Indicator <ul style="list-style-type: none"><li>● 0b0: The sending node is in error active state</li><li>● 0b1: The sending node is in error passive state</li></ul>                                                                                                                                                                                                                                                                                                                                          |
| R0[30] | XTD  | Extended Identifier<br>Indicates to the host whether the ID of the received frame is a standard ID or an extended ID. <ul style="list-style-type: none"><li>● 0b0: 11-bit standard ID</li><li>● 0b1: 29-bit extended ID</li></ul>                                                                                                                                                                                                                                                                                         |
| R0[29] | RTR  | Remote Transmission Request<br>Indicates to the host whether the received frame is a data frame or a remote frame. <ul style="list-style-type: none"><li>● 0b0: The received frame is a data frame</li><li>● 0b1: The received frame is a remote frame</li></ul> <div> <b>Note</b><br/>CAN FD does not have remote frames. In the CAN FD frame (FDF=1), the dominant bit RRS (Remote Request Substitution) replaces the RTR bit.</div> |

| Bit       | Name       | Description                                                                                                                                                                                                                                                                                                                              |
|-----------|------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| R0[28:0]  | ID[28:0]   | ID (Identifier)<br>The standard ID or extended ID depends on the XTD bit, and the standard ID is stored in ID[28:18].                                                                                                                                                                                                                    |
| R1[31]    | ANMF       | Accepted Non-matching Frame<br>Acceptance of non-match frames can be enabled via MCAN_GFC.ANFS and MCAN_GFC.ANFE.<br><ul style="list-style-type: none"> <li>● 0b0: The received frame matches the filter element pointed to by filter index FIDX</li> <li>● 0b1: The received frame does not match any receive filter element</li> </ul> |
| R1[30:24] | FIDX[6:0]  | Filter Index<br><ul style="list-style-type: none"> <li>● 0-127: Index of the matching receive filter element (inactive when ANMF=1)</li> </ul> The range is 0 to MCAN_SIDFC.LSS] minus 1 or MCAN_XIDFC.LSE minus 1.                                                                                                                      |
| R1[21]    | FDF        | FD Format<br><ul style="list-style-type: none"> <li>● 0b0: Classic CAN frame format.</li> <li>● 0b1: CAN FD frame format (new DLC encoding and CRC).</li> </ul>                                                                                                                                                                          |
| R1[20]    | BRS        | Bit Rate Switch<br><ul style="list-style-type: none"> <li>● 0b0: Receive frame without bit rate switching</li> <li>● 0b1: Receive frame with bit rate switching</li> </ul>                                                                                                                                                               |
| R1[19:16] | DLC[3:0]   | Data Length Code<br><ul style="list-style-type: none"> <li>● 0-8: Classic CAN and CAN FD: The received frame contains 0-8 data bytes</li> <li>● 9-15: CAN: The received frame contains 8 data bytes</li> <li>● 9-15: CAN FD: The received frame contains 12/16/20/24/32/48/64 data bytes</li> </ul>                                      |
| R1[15:0]  | RXTS[15:0] | Rx Timestamp<br>The timestamp counter value captured when a SOF (Start Of Frame) is received.<br>The resolution depends on the configuration of the timestamp counter prescaler MCAN_TSCC.TCP.                                                                                                                                           |
| R2[31:24] | DB3[7:0]   | Data Byte 3                                                                                                                                                                                                                                                                                                                              |
| R2[23:16] | DB2[7:0]   | Data Byte 2                                                                                                                                                                                                                                                                                                                              |
| R2[15:8]  | DB1[7:0]   | Data Byte 1                                                                                                                                                                                                                                                                                                                              |
| R2[7:0]   | DB0[7:0]   | Data Byte 0                                                                                                                                                                                                                                                                                                                              |
| R3[31:24] | DB7[7:0]   | Data Byte 7                                                                                                                                                                                                                                                                                                                              |

| Bit       | Name       | Description   |
|-----------|------------|---------------|
| R3[23:16] | DB6[7:0]   | Data Byte 6   |
| R3[15:8]  | DB5[7:0]   | Data Byte 5   |
| R3[7:0]   | DB4[7:0]   | Data Byte 4   |
| ...       | ...        | ...           |
| Rn[31:24] | DBm[7:0]   | Data Byte m   |
| Rn[23:16] | DBm-1[7:0] | Data Byte m-1 |
| Rn[15:8]  | DBm-2[7:0] | Data Byte m-2 |
| Rn[7:0]   | DBm-3[7:0] | Data Byte m-3 |

### 37.3.9.3 Transmit Buffer Elements

The transmit buffer can be configured as a dedicated transmit buffer and a transmit FIFO (or queue). If the transmit buffer is shared by a dedicated transmit buffer and a transmit FIFO (or queue), the dedicated transmit buffer is at the beginning of the transmit buffer, followed by the transmit FIFO (or queue). The transmit processing unit evaluates the transmit buffer configurations MCAN\_TXBC.TFQS and MCAN\_TXBC.NDTB to distinguish between the dedicated transmit buffer and the transmit FIFO (queue). The structure of the transmit buffer element and its related description are shown in the following table. The element size can be configured through the register MCAN\_TXESC, and the data field of the CAN FD message can be up to 64 bytes.

**Table 37-11 Transmit Buffer Elements**

|       |         |     |     |    |    |    |    |    |       |      |     |     |     |    |    |    |          |    |    |    |    |    |   |   |       |   |   |   |   |   |   |   |
|-------|---------|-----|-----|----|----|----|----|----|-------|------|-----|-----|-----|----|----|----|----------|----|----|----|----|----|---|---|-------|---|---|---|---|---|---|---|
| Bit   | 31      | 30  | 29  | 28 | 27 | 26 | 25 | 24 | 23    | 22   | 21  | 20  | 19  | 18 | 17 | 16 | 15       | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7     | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| T0    | ESI     | XTD | RTR | ID |    |    |    |    |       |      |     |     |     |    |    |    |          |    |    |    |    |    |   |   |       |   |   |   |   |   |   |   |
| T1    | MM[7:0] |     |     |    |    |    |    |    | EFC   | Res. | FDF | BRS | DLC |    |    |    | MM[15:8] |    |    |    |    |    |   |   | Res.  |   |   |   |   |   |   |   |
| T2    | DB3     |     |     |    |    |    |    |    | DB2   |      |     |     |     |    |    |    | DB1      |    |    |    |    |    |   |   | DB0   |   |   |   |   |   |   |   |
| T3    | DB7     |     |     |    |    |    |    |    | DB6   |      |     |     |     |    |    |    | DB5      |    |    |    |    |    |   |   | DB4   |   |   |   |   |   |   |   |
| ..... | .....   |     |     |    |    |    |    |    | ..... |      |     |     |     |    |    |    | .....    |    |    |    |    |    |   |   | ..... |   |   |   |   |   |   |   |
| Tn    | DBm     |     |     |    |    |    |    |    | DBm-1 |      |     |     |     |    |    |    | DBm-2    |    |    |    |    |    |   |   | DBm-3 |   |   |   |   |   |   |   |

Table 37-12 Transmit Buffer Elements Description

| Bit       | Name     | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|-----------|----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| T0[31]    | ESI      | <p>Error State Indicator</p> <ul style="list-style-type: none"> <li>● 0b0: The ESI bit of the CAN FD format depends only on the error passive flag</li> <li>● 0b1: The ESI bit of the CAN FD format is sent as a recessive bit</li> </ul> <div>  <b>Note</b><br/>           The ESI bit of the transmit buffer is ORed with the error passive flag to determine the value of the FD frame ESI bit to be transmitted. As required by the CAN FD protocol specification, an error active node may choose to transmit a recessive ESI bit, but an error passive node will always transmit a recessive ESI bit.         </div> |
| T0[30]    | XTD      | <p>Extended Identifier</p> <ul style="list-style-type: none"> <li>● 0b0: 11-bit standard ID</li> <li>● 0b1: 29-bit extended ID</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| T0[29]    | RTR      | <p>Remote Transmission Request</p> <ul style="list-style-type: none"> <li>● 0b0: Transmit data frame</li> <li>● 0b1: Transmit remote frame</li> </ul> <div>  <b>Note</b><br/>           When RTR=1, the MCAN sends remote frames as specified in ISO 11898-1:2015 even if MCAN_CCCR.FDOE enables transmission in CAN FD format.         </div>                                                                                                                                                                                                                                                                             |
| T0[28:0]  | ID[28:0] | <p>ID (Identifier)</p> <p>Standard ID or extended ID, depending on the XTD bit, the standard ID must be written in ID[28:18]</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| T1[31:24] | MM[7:0]  | <p>Message Marker</p> <p>Written by the CPU during transmit buffer configuration. Copied to the send event FIFO element to identify the send message status.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| T1[23]    | EFC      | <p>Event FIFO Control</p> <ul style="list-style-type: none"> <li>● 0b0: Do not store transmission events</li> <li>● 0b1: Store transmission events</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| T1[21]    | FDF      | <p>FD Format</p> <ul style="list-style-type: none"> <li>● 0b0: Send frames in classic CAN format</li> <li>● 0b1: Send frames in CAN FD format</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |

| Bit       | Name     | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
|-----------|----------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| T1[20]    | BRS      | <p>Bit Rate Switch</p> <ul style="list-style-type: none"> <li>● 0b1: Switch bitrate when sending CAN FD frame</li> <li>● 0b0: Do not switch bitrate when sending CAN FD frame</li> </ul> <div>  <b>Note</b><br/>           ESI, FDF, and BRS bits are only evaluated when CAN FD is enabled (MCAN_CCCR.FDOE=1). The BRS bit is only evaluated when MCAN_CCCR.BRSE=1 is additionally set.         </div> |
| T1[19:16] | DLC[3:0] | <p>Data Length Code</p> <ul style="list-style-type: none"> <li>● 0-8: Classic CAN and CAN FD: The transmitted frame contains 0 to 8 data bytes.</li> <li>● 9-15: CAN: The transmitted frame contains 8 data bytes.</li> <li>● 9-15: CAN FD: The transmitted frame contains 12/16/20/24/32/48/64 data bytes.</li> </ul>                                                                                                                                                                   |
| T1[15:8]  | MM[15:8] | <p>High byte of Wide Message Marker</p> <p>Written by the CPU during the configuration of the transmit buffer. Copied to the send event FIFO element to identify the send message status. Active only when MCAN_CCCR.WMM=1.</p>                                                                                                                                                                                                                                                          |
| T2[31:24] | DB3[7:0] | <p>Data Byte 3</p> <div>  <b>Note</b><br/>           According to the configuration of the MCAN_TXESC register, 2 to 16 (Tn=2 to 17) 32-bit words are used to store the data field of the CAN message.         </div>                                                                                                                                                                                   |
| T2[23:16] | DB2[7:0] | Data Byte 2                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| T2[15:8]  | DB1[7:0] | Data Byte 1                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| T2[7:0]   | DB0[7:0] | Data Byte 0                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| T3[31:24] | DB7[7:0] | Data Byte 7                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| T3[23:16] | DB6[7:0] | Data Byte 6                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| T3[15:8]  | DB5[7:0] | Data Byte 5                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| T3[7:0]   | DB4[7:0] | Data Byte 4                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| ...       | ...      | ...                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |

| Bit       | Name       | Description   |
|-----------|------------|---------------|
| Tn[31:24] | DBm[7:0]   | Data Byte m   |
| Tn[23:16] | DBm-1[7:0] | Data Byte m-1 |
| Tn[15:8]  | DBm-2[7:0] | Data Byte m-2 |
| Tn[7:0]   | DBm-3[7:0] | Data Byte m-3 |

### 37.3.9.4 Transmit Event FIFO Elements

Each element stores information about the message that has been sent. By reading the Transmit Event FIFO, the host CPU obtains this information in the order in which the messages were sent. The status information about the Transmit Event FIFO can be obtained from the register MCAN\_TXEFS. The structure of the Transmit Event FIFO elements and related descriptions are shown in the following table.

- E1A: When MCAN\_CCCR.WMM=0, E1A.TXTS[15:0] stores the 16-bit timestamp generated by the internal timestamp logic of the MCAN.
- E1B: When the 16-bit wide message marker is enabled (MCAN\_CCCR.WMM=1), E1B.MM[15:8] is the upper 8 bits of the wide message marker.

Table 37-13 Transmit Event FIFO Elements

|     |         |     |     |    |    |    |    |    |    |    |     |     |     |    |    |    |          |    |    |    |    |    |   |   |      |   |   |   |   |   |   |   |  |  |  |  |  |
|-----|---------|-----|-----|----|----|----|----|----|----|----|-----|-----|-----|----|----|----|----------|----|----|----|----|----|---|---|------|---|---|---|---|---|---|---|--|--|--|--|--|
| Bit | 31      | 30  | 29  | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21  | 20  | 19  | 18 | 17 | 16 | 15       | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7    | 6 | 5 | 4 | 3 | 2 | 1 | 0 |  |  |  |  |  |
| E0  | ESI     | XTD | RTR | ID |    |    |    |    |    |    |     |     |     |    |    |    |          |    |    |    |    |    |   |   |      |   |   |   |   |   |   |   |  |  |  |  |  |
| E1A | MM[7:0] |     |     |    |    |    |    |    | ET |    | FDF | BRS | DLC |    |    |    | TXTS     |    |    |    |    |    |   |   |      |   |   |   |   |   |   |   |  |  |  |  |  |
| E1B | MM[7:0] |     |     |    |    |    |    |    | ET |    | FDF | BRS | DLC |    |    |    | MM[15:8] |    |    |    |    |    |   |   | Res. |   |   |   |   |   |   |   |  |  |  |  |  |

Table 37-14 Transmit Event FIFO Element Description

| Bit    | Name | Description                                                                                                                                                                         |
|--------|------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| E0[31] | ESI  | Error State Indicator <ul style="list-style-type: none"> <li>● 0b0: The sending node is in error active state</li> <li>● 0b1: The sending node is in error passive state</li> </ul> |
| E0[30] | XTD  | Extended Identifier <ul style="list-style-type: none"> <li>● 0b0: 11-bit standard ID</li> <li>● 0b1: 29-bit extended ID</li> </ul>                                                  |

| Bit          | Name       | Description                                                                                                                                                                                                                                                                                                             |
|--------------|------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| E0[29]       | RTR        | Remote Transmission Request <ul style="list-style-type: none"> <li>● 0b0: The transmitted frame is a data frame.</li> <li>● 0b1: The transmitted frame is a remote frame.</li> </ul>                                                                                                                                    |
| E0[28:0]     | ID[28:0]   | Identifier<br>Standard ID or extended ID, depending on the XTD bit. The standard ID is stored in ID[28:18].                                                                                                                                                                                                             |
| E1A/B[31:24] | MM[7:0]    | Message Marker<br>Copied from the transmit buffer to the transmit event element, used to identify the transmission message status.                                                                                                                                                                                      |
| E1A/B[23:22] | ET[1:0]    | Event Type <ul style="list-style-type: none"> <li>● 0b00: Reserved</li> <li>● 0b01: Transmit Event</li> <li>● 0b10: Transmit after cancellation (this value is always used when transmitting in DAR mode).</li> <li>● 0b11: Reserved</li> </ul>                                                                         |
| E1A/B[21]    | FDF        | FD Format <ul style="list-style-type: none"> <li>● 0b0: Classic CAN frame format.</li> <li>● 0b1: CAN FD frame format (new DLC encoding and CRC).</li> </ul>                                                                                                                                                            |
| E1A/B[20]    | BRS        | Bit Rate Switch <ul style="list-style-type: none"> <li>● 0b0: No bitrate switching when sending CAN FD frame</li> <li>● 0b1: Bitrate switching when sending CAN FD frame.</li> </ul>                                                                                                                                    |
| E1A/B[19:16] | DLC[3:0]   | Data Length Code <ul style="list-style-type: none"> <li>● 0-8: Classic CAN and CAN FD: The transmitted frame contains 0 to 8 data bytes.</li> <li>● 9-15: Classic CAN: The transmitted frame contains 8 data bytes.</li> <li>● 9-15: CAN FD: The transmitted frame contains 12/16/20/24/32/48/64 data bytes.</li> </ul> |
| E1A[15:0]    | TXTS[15:0] | Tx Timestamp<br>Timestamp counter value captured when sending SOF (Start Of Frame). The resolution depends on the configuration of the time-stamp counter prescaler MCAN_TSCC.TCP.                                                                                                                                      |
| E1B[15:8]    | MM[15:8]   | High byte of Wide Message Marker<br>Written by the CPU during the configuration of the transmit buffer. Copied to the send event FIFO element to identify the send message status.                                                                                                                                      |



### 37.3.9.5 Standard Message ID Filter Element

Up to 128 filter elements can be configured for 11-bit standard IDs (each element is 1 word in size). The address of the filter element is the filter element index (0-127) multiplied by 4 (in bytes), added to the base address MCAN\_SIDFC.FLSSA. The structure of the standard message ID element and its related description are shown in the table below.

**Table 37-15 Standard Message ID Filter Element**

|     |     |    |      |    |    |       |    |    |    |    |    |    |    |    |    |    |      |    |    |    |    |       |   |   |   |   |   |   |   |   |   |   |
|-----|-----|----|------|----|----|-------|----|----|----|----|----|----|----|----|----|----|------|----|----|----|----|-------|---|---|---|---|---|---|---|---|---|---|
| Bit | 31  | 30 | 29   | 28 | 27 | 26    | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15   | 14 | 13 | 12 | 11 | 10    | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| S0  | SFT |    | SFEC |    |    | SFID1 |    |    |    |    |    |    |    |    |    |    | Res. |    |    |    |    | SFID2 |   |   |   |   |   |   |   |   |   |   |

**Table 37-16 Standard Message ID Filter Element Description**

| Bit       | Name      | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|-----------|-----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| S0[31:30] | SFT[1:0]  | <p>Standard Filter Type</p> <ul style="list-style-type: none"> <li>● 0b00: Range filter from SFID1 to SFID2 (<math>SFID2 \geq SFID1</math>)</li> <li>● 0b01: Dual ID filter defined by SFID1 or SFID2</li> <li>● 0b10: Classic filter: <math>SFID1=ID</math>, <math>SFID2=mask</math></li> <li>● 0b11: Disable this filter element</li> </ul> <div>  <b>Note</b><br/>           When SFT=0b11, this filter element is disabled and acceptance filtering continues (same behavior as when SFEC=0b000).         </div>                                                                                                                                                                                                                                                                                                                                                                                                              |
| S0[29:27] | SFEC[2:0] | <p>Standard Filter Element Configuration</p> <p>All enabled filter elements are used for reception filtering of standard frames. When an enabled filter element is matched or the end of the filter list is reached, reception filtering stops. If SFEC=0b100, 0b101 or 0b110, the interrupt flag MCAN_IR.HPM is set when a match occurs and an interrupt is generated (if enabled). In this case, the register MCAN_HPMS is updated with the status of the priority match.</p> <ul style="list-style-type: none"> <li>● 0b000: Disable this filter element</li> <li>● 0b001: If matched, stored in receive FIFO0</li> <li>● 0b010: If a match is found, store it in receive FIFO1</li> <li>● 0b011: If match, reject ID</li> <li>● 0b100: If match, set priority, do not store</li> <li>● 0b101: If match, set priority and store in FIFO0</li> <li>● 0b110: If match, set priority and store in FIFO1</li> <li>● 0b111: Store in receive buffer or as debug message, ignore configuration of SFT[1:0]</li> </ul> |

| Bit       | Name        | Description                                                                                                                                                                                                                                                                                                                                      |
|-----------|-------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| S0[26:16] | SFID1[10:0] | Standard Filter ID 1<br>First ID of the standard ID filter element. When filtering for receive buffers or debug messages, this bitfield defines the ID of the standard message to be stored. The received ID must be matched exactly, no mask mechanism is used.                                                                                 |
| S0[10:0]  | SFID2[10:0] | Standard Filter ID 2<br>This bit field has different meanings, depending on the configuration of SFEC: <ul style="list-style-type: none"> <li>● When SFEC=0b001-0b110: The second ID of the standard ID filter element</li> <li>● When SFEC=0b111: Filter for receive buffer or debug message</li> </ul>                                         |
|           | SFID2[10:9] | Determines whether the received message is stored in the receive buffer or processed as message A, B, or C in the debug message sequence. <ul style="list-style-type: none"> <li>● 0b00: store the message into the receive buffer</li> <li>● 0b01: Debug message A</li> <li>● 0b10: Debug message B</li> <li>● 0b11: Debug message C</li> </ul> |
|           | SFID2[5:0]  | Defines the offset (buffer index) from the receive buffer where the matching message is stored to the starting address RXBC.RBSA.                                                                                                                                                                                                                |

### 37.3.9.6 Extended Message ID Filter Element

Up to 64 filter elements (each element is 2 words in size) can be configured for a 29-bit extended ID. When accessing an extended message ID filter element, its address is the filter element index (0 to 63) multiplied by 2, then multiplied by 4 (converted to bytes), and then added to the starting address MCAN\_XIDFC.FLESA. The structure of the extended message ID element and its related description are shown in the following table.

**Table 37-17 Extended Message ID Filter Element**

| Bit | 31   | 30 | 29   | 28    | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-----|------|----|------|-------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| F0  | EFEC |    |      | EFID1 |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| F1  | EFT  |    | Res. | EFID2 |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

**Table 37-18 Extended Message ID Filter Element Description**

| Bit       | Name      | Description                                                                                                                                                                                                                              |
|-----------|-----------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| F0[31:29] | EFEC[2:0] | Extended Filter Element Configuration<br>All enabled filter elements are used for reception filtering of extended frames. When an enabled filter element is matched or the end of the filter list is reached, reception filtering stops. |

| Bit       | Name        | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
|-----------|-------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| F0[31:29] | EFEC[2:0]   | <ul style="list-style-type: none"> <li>● 0b000: Disable this filter element</li> <li>● 0b001: If matched, stored in receive FIFO0</li> <li>● 0b010: If a match is found, store it in receive FIFO1</li> <li>● 0b011: If match, reject ID</li> <li>● 0b100: If matched, set priority</li> <li>● 0b101: If match, set priority and store into FIFO0</li> <li>● 0b110: If match, set priority and store in FIFO1</li> <li>● 0b111: Store in the receive buffer or as a debug message, ignoring the configuration of EFT[1:0]</li> </ul> <p>If EFEC=0b100, 0b101 or 0b110, the interrupt flag MCAN_IR.HPM is set when a match occurs, and an interrupt is generated (if enabled). In this case, the register MCAN_HPMS is updated with the status of the priority match.</p> |
| F0[28:0]  | EFID1[28:0] | <p>Extended Filter ID 1</p> <p>The first ID of the extended ID filter element. When filtering for receive buffer or debug messages, this bitfield defines the ID of the extended message to be stored. The received ID must be matched exactly, only using the MCAN_XIDAM mask mechanism.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| F1[31:30] | EFT[1:0]    | <p>Extended Filter Type</p> <ul style="list-style-type: none"> <li>● 0b00: Range filter from EFID1 to EFID2 (<math>EFID2 \geq EFID1</math>)</li> <li>● 0b01: Dual ID filter defined by EFID1 or EFID2</li> <li>● 0b10: Classic filter: <math>EFID1=ID</math>, <math>EFID2=mask</math></li> <li>● 0b11: Range filter from EFID1 to EFID2 (<math>EFID2 \geq EFID1</math>), do not use MCAN_XIDAM mask</li> </ul>                                                                                                                                                                                                                                                                                                                                                           |
| F1[28:0]  | EFID2[28:0] | <p>Extended Filter ID 2</p> <p>This bit field has different meanings, depending on the configuration of EFEC:</p> <ul style="list-style-type: none"> <li>● EFEC=0b001-0b110: The second ID of the extended ID filter element</li> <li>● EFEC=0b111: Filter for receiving buffer or debug messages</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
|           | EFID2[10:9] | <p>Determines whether the received message is stored in the receive buffer or processed as message A, B, or C in the debug message sequence.</p> <ul style="list-style-type: none"> <li>● 0b00: Store the message into the receive buffer</li> <li>● 0b01: Debug message A</li> <li>● 0b10: Debug message B</li> <li>● 0b11: Debug message C</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                  |
|           | EFID2[5:0]  | <p>Defines the offset (buffer index) of the receive buffer where matching messages are stored to the starting address MCAN_RXBC.RBSA.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |

## 37.4 Register Description

### 37.4.1 Register List

- MCAN1 base address: 0x4002B000
- MCAN2 base address: 0x4002B400

Table 37-19 Controller Area Network (MCAN) Register Offset

| Offset | Register   | Description                                    | Width |
|--------|------------|------------------------------------------------|-------|
| 0x04   | MCAN_ENDN  | MCAN Byte Order Register                       | 32    |
| 0x0C   | MCAN_DBTP  | MCAN Data Bit Time and Prescaler Register      | 32    |
| 0x10   | MCAN_TEST  | MCAN Test Register                             | 32    |
| 0x14   | MCAN_RWD   | MCAN RAM Watchdog Register                     | 32    |
| 0x18   | MCAN_CCCR  | MCAN CC Control Register                       | 32    |
| 0x1C   | MCAN_NBTP  | MCAN Nominal Bit Time and Prescaler Register   | 32    |
| 0x20   | MCAN_TSCC  | MCAN Timestamp Counter Configuration Register  | 32    |
| 0x24   | MCAN_TSCV  | MCAN Timestamp Counter Value Register          | 32    |
| 0x28   | MCAN_TOCC  | MCAN Timeout Counter Configuration Register    | 32    |
| 0x2C   | MCAN_TOCV  | MCAN Timeout Counter Value Register            | 32    |
| 0x40   | MCAN_ECR   | MCAN Error Counter Register                    | 32    |
| 0x44   | MCAN_PSR   | MCAN Protocol Status Register                  | 32    |
| 0x48   | MCAN_TDCR  | MCAN Transmitter Delay Compensation Register   | 32    |
| 0x50   | MCAN_IR    | MCAN Interrupt Register                        | 32    |
| 0x54   | MCAN_IE    | MCAN Interrupt Enable Register                 | 32    |
| 0x58   | MCAN_ILS   | MCAN Interrupt Line Selection Register         | 32    |
| 0x5C   | MCAN_ILE   | MCAN Interrupt Line Enable Register            | 32    |
| 0x80   | MCAN_GFC   | MCAN Global Filter Configuration Register      | 32    |
| 0x84   | MCAN_SIDFC | MCAN Standard ID Filter Configuration Register | 32    |
| 0x88   | MCAN_XIDFC | MCAN Extended ID Filter Configuration Register | 32    |
| 0x90   | MCAN_XIDAM | MCAN Extended ID and Mask Register             | 32    |
| 0x94   | MCAN_HPMS  | MCAN High Priority Message Status Register     | 32    |
| 0x98   | MCAN_NDAT1 | MCAN New Data 1 Register                       | 32    |

| Offset | Register    | Description                                                       | Width |
|--------|-------------|-------------------------------------------------------------------|-------|
| 0x9C   | MCAN_NDAT2  | MCAN New Data 2 Register                                          | 32    |
| 0xA0   | MCAN_RXF0C  | MCAN Receive FIFO0 Configuration Register                         | 32    |
| 0xA4   | MCAN_RXF0S  | MCAN Receive FIFO0 Status Register                                | 32    |
| 0xA8   | MCAN_RXF0A  | MCAN Receive FIFO0 Acknowledge Register                           | 32    |
| 0xAC   | MCAN_RXBC   | MCAN Receive Buffer Configuration Register                        | 32    |
| 0xB0   | MCAN_RXF1C  | MCAN Receive FIFO1 Configuration Register                         | 32    |
| 0xB4   | MCAN_RXF1S  | MCAN Receive FIFO1 Status Register                                | 32    |
| 0xB8   | MCAN_RXF1A  | MCAN Receive FIFO1 Acknowledge Register                           | 32    |
| 0xBC   | MCAN_RXESC  | MCAN Receive Buffer and FIFO Element Size Configuration Registers | 32    |
| 0xC0   | MCAN_TXBC   | MCAN Transmit Buffer Configuration Register                       | 32    |
| 0xC4   | MCAN_TXFQS  | MCAN Transmit FIFO/Queue Status Register                          | 32    |
| 0xC8   | MCAN_TXESC  | MCAN Transmit Buffer Element Size Configuration Register          | 32    |
| 0xCC   | MCAN_TXBRP  | MCAN Transmit Buffer Request Pending Register                     | 32    |
| 0xD0   | MCAN_TXBAR  | MCAN Transmit Buffer Add Request Register                         | 32    |
| 0xD4   | MCAN_TXBCR  | MCAN Transmit Buffer Cancel Request Register                      | 32    |
| 0xD8   | MCAN_TXBTO  | MCAN Transmit Buffer Transmit Occurred Register                   | 32    |
| 0xDC   | MCAN_TXBCF  | MCAN Transmit Buffer Cancel Completed Register                    | 32    |
| 0xE0   | MCAN_TXBTIE | MCAN Transmit Buffer Transmit Interrupt Enable Register           | 32    |
| 0xE4   | MCAN_TXBCIE | MCAN Transmit Buffer Cancel Completed Interrupt Enable Register   | 32    |
| 0xF0   | MCAN_TXEFC  | MCAN Transmit Event FIFO Configuration Register                   | 32    |
| 0xF4   | MCAN_TXEFS  | MCAN Transmit Event FIFO Status Register                          | 32    |
| 0xF8   | MCAN_TXEFA  | MCAN Transmit Event FIFO Acknowledge Register                     | 32    |

MCANRAM ECC base address: 0x4002D000

Table 37-20 MCANRAM register offset

| Offset | Register                           | Description                                            | Width |
|--------|------------------------------------|--------------------------------------------------------|-------|
| 0x00   | <a href="#">MCANRAM_CKCR</a>       | MCAN Message RAM Check Control Register                | 32    |
| 0x04   | <a href="#">MCANRAM_CKSR</a>       | MCAN Message RAM Check Status Register                 | 32    |
| 0x08   | <a href="#">MCANRAM_EIEN</a>       | MCAN Message RAM Error Injection Enable Register       | 32    |
| 0x0C   | <a href="#">MCANRAM_EIBIT0</a>     | MCAN Message RAM Error Injection Bit Enable Register 0 | 32    |
| 0x10   | <a href="#">MCANRAM_EIBIT1</a>     | MCAN Message RAM Error Injection Bit Enable Register 1 | 32    |
| 0x14   | <a href="#">MCANRAM_ECCERRADDR</a> | MCAN Message RAM Check Error Address Register          | 32    |

37.4.2 MCAN Byte Order Register (MCAN\_ENDN)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0x04   | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x87654321   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | ETV          |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access | RO           |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit  | Name | Function              | Description                  | Access |
|------|------|-----------------------|------------------------------|--------|
| 31:0 | ETV  | Byte order test value | The test value is 0x87654321 | RO     |

### 37.4.3 MCAN Data Bit Time and Prescaler Register (MCAN\_DBTP)

Write access to this register is possible only when both the MCAN\_CCCR.CCE and MCAN\_CCCR.INIT bits are set. The programmable range for the CAN bit time is from 4 to 49 time quanta(tq). The programmable range for the CAN time quanta is from 1 to 32 MCAN communication clock periods (Minimum time quantum, mtq).  $tq = (DBRP + 1) \cdot mtq$ .  $tq = (DBRP + 1) \cdot mtq$ .

DTSEG1 is the sum of Prop\_Seg and Phase\_Seg1. DTSEG2 is Phase\_Seg2.

Therefore, the length of the bit time is (programmed value) [DTSEG1 + DTSEG2 + 3] tq or (functional value) [Sync\_Seg + Prop\_Seg + Phase\_Seg1 + Phase\_Seg2] tq.

The information processing time (IPT) is zero, which means that the next bit of data is available on the first clock edge after the sampling point.

| Offset | Bit Position |    |    |    |    |    |    |    |     |          |      |    |    |    |          |        |    |    |    |        |     |    |   |      |   |     |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|-----|----------|------|----|----|----|----------|--------|----|----|----|--------|-----|----|---|------|---|-----|---|---|---|---|---|---|
| 0x0C   | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23  | 22       | 21   | 20 | 19 | 18 | 17       | 16     | 15 | 14 | 13 | 12     | 11  | 10 | 9 | 8    | 7 | 6   | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000A33   |    |    |    |    |    |    |    |     |          |      |    |    |    |          |        |    |    |    |        |     |    |   |      |   |     |   |   |   |   |   |   |
| Name   | Reserved     |    |    |    |    |    |    |    | TDC | Reserved | DBRP |    |    |    | Reserved | DTSEG1 |    |    |    | DTSEG2 |     |    |   | DSJW |   |     |   |   |   |   |   |   |
| Access |              |    |    |    |    |    |    |    | RWA |          | RWA  |    |    |    |          | RWA    |    |    |    |        | RWA |    |   |      |   | RWA |   |   |   |   |   |   |

| Bit   | Name     | Function                            | Description                                                                                                                                                                                                                                                                                        | Access |
|-------|----------|-------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:24 | Reserved | Reserved                            | -                                                                                                                                                                                                                                                                                                  | -      |
| 23    | TDC      | Transmit delay compensation         | <ul style="list-style-type: none"> <li>0b0: Disable</li> <li>0b1: Enable</li> </ul>                                                                                                                                                                                                                | RWA    |
| 22:21 | Reserved | Reserved                            | -                                                                                                                                                                                                                                                                                                  | -      |
| 20:16 | DBRP     | Data bit rate prescaler             | The MCAN communication clock prescaler used to generate time quanta for bit timing; the bit time is a multiple of this time quantum. Active prescaler values range from 0 to 31. When TDC=1, the range is limited to 0 or 1. The actual value used by the hardware is the programmed value plus 1. | RWA    |
| 15:13 | Reserved | Reserved                            | -                                                                                                                                                                                                                                                                                                  | -      |
| 12:8  | DTSEG1   | Data Time Before Sample Point       | Active setting: 0-31. The actual value used by hardware is the programmed value plus 1.                                                                                                                                                                                                            | RWA    |
| 7:4   | DTSEG2   | Data Time After Sample Point        | Active setting: 1-15. The actual value used by hardware is the programmed value plus 1.                                                                                                                                                                                                            | RWA    |
| 3:0   | DSJW     | Data (Re)synchronization Jump Width | Active setting: 0-15. The actual value used by hardware is the programmed value plus 1.                                                                                                                                                                                                            | RWA    |



**Note**

1. When the MCAN communication clock is 8 MHz, the reset value 0x00000A33 configures the MCAN data phase bit rate to 500 kbps.
2. The CAN FD data phase bit rate configured via DBTP must not be lower than the arbitration phase bit rate configured via NBTP.

### 37.4.4 MCAN Test Register (MCAN\_TEST)

Only when the MCAN\_CCCR.TEST bit is set to 1 can write access to the test register be enabled. When the MCAN\_CCCR.TEST bit is cleared to 0, all test register functions revert to their reset values.

Loopback mode and software control of the MCAN\_TX pin are considered hardware test modes. If the TX setting is configured to a value other than 0b00, it may interfere with message transmission on the CAN bus.

| Offset | Bit Position                                                                               |    |    |    |    |    |    |    |    |    |      |       |    |    |    |          |      |       |    |    |    |    |     |   |      |          |   |   |   |   |   |   |
|--------|--------------------------------------------------------------------------------------------|----|----|----|----|----|----|----|----|----|------|-------|----|----|----|----------|------|-------|----|----|----|----|-----|---|------|----------|---|---|---|---|---|---|
| 0x10   | 31                                                                                         | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21   | 20    | 19 | 18 | 17 | 16       | 15   | 14    | 13 | 12 | 11 | 10 | 9   | 8 | 7    | 6        | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x000000X0 (The reset value of Bit 7 is determined by the actual level on the MCAN_RX pin) |    |    |    |    |    |    |    |    |    |      |       |    |    |    |          |      |       |    |    |    |    |     |   |      |          |   |   |   |   |   |   |
| Name   | Reserved                                                                                   |    |    |    |    |    |    |    |    |    | SVAL | TXBNS |    |    |    | Reserved | PVAL | TXBNP |    |    |    | RX | TX  |   | LBCK | Reserved |   |   |   |   |   |   |
| Access |                                                                                            |    |    |    |    |    |    |    |    |    | RO   | RO    |    |    |    |          | RO   | RO    |    |    |    | RO | RWA |   | RWA  |          |   |   |   |   |   |   |

| Bit   | Name     | Function                                                       | Description                                                                                                                                                                                                                                                                                                                                      | Access |
|-------|----------|----------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:22 | Reserved | Reserved                                                       | -                                                                                                                                                                                                                                                                                                                                                | -      |
| 21    | SVAL     | Already active                                                 | <ul style="list-style-type: none"> <li>0b0: Value of TXBNS disabled</li> <li>0b1: Value of TXBNS enabled</li> </ul>                                                                                                                                                                                                                              | RO     |
| 20:16 | TXBNS    | The buffer number of the buffer that has started transmitting. | The buffer number of the last buffer that started transmitting.<br>When the SVAL bit is set to 1, the active setting range is 0-31.                                                                                                                                                                                                              | RO     |
| 15:14 | Reserved | Reserved                                                       | -                                                                                                                                                                                                                                                                                                                                                | -      |
| 13    | PVAL     | Ready to use                                                   | <ul style="list-style-type: none"> <li>0b0: Value of TXBNP disabled</li> <li>0b1: Value of TXBNP enabled</li> </ul>                                                                                                                                                                                                                              | RO     |
| 12:8  | TXBNP    | The number of buffers that are ready to send                   | Transmitting buffer number ready to send<br>When the PVAL bit is set to 1, the active setting range is 0-31.                                                                                                                                                                                                                                     | RO     |
| 7     | RX       | Receive Pin                                                    | The actual value of the monitored MCAN_RX pin <ul style="list-style-type: none"> <li>0b0: CAN bus is dominant (MCAN_RX=0)</li> <li>0b1: CAN bus is recessive (MCAN_RX=1)</li> </ul>                                                                                                                                                              | RO     |
| 6:5   | TX       | Transmit pin control                                           | <ul style="list-style-type: none"> <li>0b00: Reset value, MCAN_TX pin is controlled by the CAN core and updated at the end of the CAN bit time</li> <li>0b01: The sampling point can be monitored on the MCAN_TX pin</li> <li>0b10: MCAN_TX pin outputs a dominant level (0)</li> <li>0b11: MCAN_TX pin outputs a recessive level (1)</li> </ul> | RWA    |

| Bit | Name            | Function        | Description                                                                                                                                                                     | Access |
|-----|-----------------|-----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 4   | LBCK            | Loop mode       | <ul style="list-style-type: none"><li>● 0b0: Reset value, disable loopback mode</li><li>● 0b1: Enable loopback mode (see "<a href="#">Loopback Mode</a>" for details)</li></ul> | RWA    |
| 3:0 | <i>Reserved</i> | <i>Reserved</i> | -                                                                                                                                                                               | -      |

### 37.4.5 MCAN RAM Watchdog Register (MCAN\_RWD)

The RAM watchdog monitors the READY signal output of the message RAM. When the message RAM is accessed, the message RAM watchdog counter is started, and the starting value of the counter is configured by the MCAN\_RWD.WDC bit.

When the message RAM indicates that the access is successfully completed by activating its READY signal output, the counter is reloaded with the value of MCAN\_RWD.WDC. If the message RAM does not respond before the counter decrements to 0, the counter stops counting and sets the interrupt flag MCAN\_IR.WDI to 1. The RAM watchdog counter is counted by the MCAN control logic clock.

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |     |    |    |    |    |    |   |   |     |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-----|----|----|----|----|----|---|---|-----|---|---|---|---|---|---|---|
| 0x14   | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15  | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7   | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |     |    |    |    |    |    |   |   |     |   |   |   |   |   |   |   |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | WDV |    |    |    |    |    |   |   | WDC |   |   |   |   |   |   |   |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RO  |    |    |    |    |    |   |   | RWA |   |   |   |   |   |   |   |

| Bit   | Name     | Function               | Description                                                                                                     | Access |
|-------|----------|------------------------|-----------------------------------------------------------------------------------------------------------------|--------|
| 31:16 | Reserved | Reserved               | -                                                                                                               | -      |
| 15:8  | WDV      | Watchdog count value   | Actual value of the message RAM watchdog counter                                                                | RO     |
| 7:0   | WDC      | Watchdog configuration | Start value of the message RAM watchdog counter.<br>If the initial value 0x00 is used, the counter is disabled. | RWA    |

### 37.4.6 MCAN CC Control Register (MCAN\_CCCR)

For the setting conditions of each bit of the register, see "[Software Initialization](#)".

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |      |     |      |      |     |          |      |      |      |     |     |     |     |     |     |      |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|------|-----|------|------|-----|----------|------|------|------|-----|-----|-----|-----|-----|-----|------|
| 0x18   | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15   | 14  | 13   | 12   | 11  | 10       | 9    | 8    | 7    | 6   | 5   | 4   | 3   | 2   | 1   | 0    |
| Reset  | 0x00000001   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |      |     |      |      |     |          |      |      |      |     |     |     |     |     |     |      |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | NISO | TXP | EFBI | PXHD | WMM | Reserved | BRSE | FDOE | TEST | DAR | MON | CSR | CSA | ASM | CCE | INIT |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RWA  | RWA | RWA  | RWA  | RWA |          | RWA  | RWA  | RWA  | RWA | RW  | RO  | RWA | RWA | RW  |      |
|        |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |      |     |      |      |     |          |      |      |      |     |     |     |     |     |     |      |
|        |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |      |     |      |      |     |          |      |      |      |     |     |     |     |     |     |      |
|        |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |      |     |      |      |     |          |      |      |      |     |     |     |     |     |     |      |

| Bit   | Name     | Function                                  | Description                                                                                                                                                                                                                                                                                                                                                                                                                               | Access |
|-------|----------|-------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:16 | Reserved | Reserved                                  | -                                                                                                                                                                                                                                                                                                                                                                                                                                         | -      |
| 15    | NISO     | Non-ISO mode selection                    | <ul style="list-style-type: none"> <li>0b0: Use the CAN FD frame format specified by ISO 11898-1:2015</li> <li>0b1: Use the CAN FD frame format specified by Bosch CAN FD specification V1.0</li> </ul>                                                                                                                                                                                                                                   | RWA    |
| 14    | TXP      | Transmission pause                        | <ul style="list-style-type: none"> <li>0b0: Disable transmission pause</li> <li>0b1: Enable transmission pause</li> </ul> <div>  <b>Note</b><br/>           If this bit is set to 1, MCAN will pause for two bit times after successfully transmitting a frame of data before transmitting the next frame of data.         </div>                      | RWA    |
| 13    | EFBI     | Edge filtering during bus synchronization | <ul style="list-style-type: none"> <li>0b0: Edge filtering disabled</li> <li>0b1: Two consecutive dominant tq's are required to detect a hard sync edge</li> </ul>                                                                                                                                                                                                                                                                        | RWA    |
| 12    | PXHD     | Protocol exception handling disabled      | <ul style="list-style-type: none"> <li>0b0: Enable protocol exception processing</li> <li>0b1: Disable protocol exception processing</li> </ul> <div>  <b>Note</b><br/>           When protocol exception handling is disabled, MCAN will transmit error frames upon detection of a protocol exception condition.         </div>                       | RWA    |
| 11    | WMM      | Wide message marker                       | <ul style="list-style-type: none"> <li>0b0: Use 8-bit message marker</li> <li>0b1: Use 16-bit message marker, replace the 16-bit timestamp in the transmit event FIFO</li> </ul> <div>  <b>Note</b><br/>           When using 16-bit wide message marker (WMM=1), the 16-bit internal timestamp in the transmit event FIFO is disabled.         </div> | RWA    |
| 10    | Reserved | Reserved                                  | -                                                                                                                                                                                                                                                                                                                                                                                                                                         | -      |

| Bit | Name | Function                  | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | Access |
|-----|------|---------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 9   | BRSE | Bit rate switching        | <ul style="list-style-type: none"> <li>0b0: Disable bit rate switching when sending</li> <li>0b1: Enable bit rate switching when sending</li> </ul> <div>  <b>Note</b><br/>           When CAN FD mode is disabled (FDOE=0), the setting of BRSE has no effect.         </div>                                                                                                                   | RWA    |
| 8   | FDOE | FD mode enable            | <ul style="list-style-type: none"> <li>0b0: Disable FD mode</li> <li>0b1: Enable FD mode</li> </ul>                                                                                                                                                                                                                                                                                                                                                                               | RWA    |
| 7   | TEST | Test mode enabled         | Write 1 to enable test mode, write 0 has no effect<br>Read: <ul style="list-style-type: none"> <li>0b0: Normal mode, test register MCAN_TEST keeps reset value</li> <li>0b1: Test mode, enable write access to test register MCAN_TEST</li> </ul>                                                                                                                                                                                                                                 | RWA    |
| 6   | DAR  | Disable automatic resend  | <ul style="list-style-type: none"> <li>0b0: Enable automatic resend of unsuccessfully sent messages</li> <li>0b1: Disable automatic resend</li> </ul>                                                                                                                                                                                                                                                                                                                             | RWA    |
| 5   | MON  | Bus listening mode        | Write 1 to enable the bus monitoring mode, write 0 has no effect<br>Read: <ul style="list-style-type: none"> <li>0b0: Disables bus monitoring mode</li> <li>0b1: Enables bus monitoring mode</li> </ul> <div>  <b>Note</b><br/>           The MON bit can only be set to 1 by the host when both CCE and INIT bits are set to 1. This bit can be reset by the host at any time.         </div> | RWA    |
| 4   | CSR  | Clock stop request        | <ul style="list-style-type: none"> <li>0b0: No clock stop requested</li> <li>0b1: Clock stop requested</li> </ul> <div>  <b>Note</b><br/>           When clock stop is requested, after all pending transmission requests are completed and the CAN bus reaches the idle state, the INIT bit and CSA bit will be set to 1 sequentially.         </div>                                         | RW     |
| 3   | CSA  | Clock stop acknowledgment | <ul style="list-style-type: none"> <li>0b0: Clock stop request not acknowledged</li> <li>0b1: Clock stop request acknowledged, MCAN can be set to Power-down state by stopping the control logic clock and communication clock.</li> </ul>                                                                                                                                                                                                                                        | RO     |

| Bit | Name | Function                     | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | Access |
|-----|------|------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 2   | ASM  | Restricted working mode      | <p>Write 1 to enable restricted working mode, write 0 has no effect</p> <p>Read:</p> <ul style="list-style-type: none"> <li>● 0b0: Normal mode</li> <li>● 0b1: Restricted working mode</li> </ul> <div>  <b>Note</b><br/>           The host can set the ASM bit only when both CCE and INIT are set. This bit can be reset by the host at any time.<br/><br/>           For a description of restricted working mode, see "<a href="#">Restricted Working Mode</a>".         </div> | RWA    |
| 1   | CCE  | Configuration change enabled | <ul style="list-style-type: none"> <li>● 0b0: CPU has no write access to protected configuration registers</li> <li>● 0b1: CPU has write access to protected configuration registers (when INIT=1)</li> </ul>                                                                                                                                                                                                                                                                                                                                                         | RWA    |
| 0   | INIT | Initialization               | <ul style="list-style-type: none"> <li>● 0b0: Normal operation</li> <li>● 0b1: Start initialization</li> </ul> <div>  <b>Note</b><br/>           Due to the synchronization mechanism between the two clock domains, the value of INIT may be delayed after being written before it can be read correctly. Therefore, before resetting the INIT value, please confirm that the previous setting value has been written.         </div>                                               | RW     |

### 37.4.7 MCAN Nominal Bit Time and Prescaler Register (MCAN\_NBTP)

This register can be written only when both the MCAN\_CCCR.CCE and MCAN\_CCR.INIT bits are set. The CAN nominal bit time is programmable from 4 to 385 time slices (Time quanta, tq). The CAN time slice is programmable from 1 to 512 MCAN communication clock periods (Minimum time quantum, mtq).  
 $tq = (NBRP + 1) mtq$ .

NTSEG1 is the sum of Prop\_Seg and Phase\_Seg1. NTSEG2 is Phase\_Seg2.

Therefore, the length of the bit time is (programmed value) [NTSEG1 + NTSEG2 + 3] tq or (functional value) [Sync\_Seg + Prop\_Seg + Phase\_Seg1 + Phase\_Seg2] tq.

The information processing time (IPT) is zero, which means that the next bit of data is available on the first clock edge after the sample point.

| Offset | Bit Position |    |    |    |    |    |    |    |      |    |    |    |    |    |    |    |        |    |    |    |    |    |   |   |          |        |   |   |   |   |   |   |  |
|--------|--------------|----|----|----|----|----|----|----|------|----|----|----|----|----|----|----|--------|----|----|----|----|----|---|---|----------|--------|---|---|---|---|---|---|--|
| 0x1C   | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23   | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15     | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7        | 6      | 5 | 4 | 3 | 2 | 1 | 0 |  |
| Reset  | 0x06000A03   |    |    |    |    |    |    |    |      |    |    |    |    |    |    |    |        |    |    |    |    |    |   |   |          |        |   |   |   |   |   |   |  |
| Name   | NSJW         |    |    |    |    |    |    |    | NBRP |    |    |    |    |    |    |    | NTSEG1 |    |    |    |    |    |   |   | Reserved | NTSEG2 |   |   |   |   |   |   |  |
| Access |              |    |    |    |    |    |    |    |      |    |    |    |    |    |    |    |        |    |    |    |    |    |   |   | RWA      |        |   |   |   |   |   |   |  |

| Bit   | Name            | Function                                      | Description                                                                                                                                                                                                                                                   | Access |
|-------|-----------------|-----------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:25 | NSJW            | Nominal (re) synchronization jump width       | The active setting value is 0-127, and the actual value used by the hardware is the programmed value plus 1.                                                                                                                                                  | RWA    |
| 24:16 | NBRP            | Nominal bit rate prescaler                    | The MCAN communication clock prescaler value that generates a bit time slice, and the bit time is a multiple of the time slice. The active setting value of the prescaler is 0-511, and the actual value used by the hardware is the programmed value plus 1. | RWA    |
| 15:8  | NTSEG1          | Nominal time period before the sampling point | The active setting value is 1-255, and the actual value used by the hardware is the programmed value plus 1.                                                                                                                                                  | RWA    |
| 7     | <i>Reserved</i> | <i>Reserved</i>                               | -                                                                                                                                                                                                                                                             | -      |
| 6:0   | NTSEG2          | Nominal time period after the sampling point  | The active setting value is 1-127, and the actual value used by the hardware is the programmed value plus 1.                                                                                                                                                  | RWA    |



#### Note

When the MCAN clock is 8MHz, a reset value of 0x06000A03 will configure the MCAN bit rate to 500kbps.



### 37.4.8 MCAN Timestamp Counter Configuration Register (MCAN\_TSCC)

This register is used to configure the internal 16-bit timestamp counter. For an introduction to internal timestamp processing, see "[Timestamp Generation](#)".

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |     |    |          |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |     |  |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|-----|----|----------|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|-----|--|
| 0x20   | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19  | 18 | 17       | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |     |  |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |     |    |          |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |     |  |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    | TCP |    | Reserved |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   | TSS |  |
| Access |              |    |    |    |    |    |    |    |    |    |    |    | RWA |    |          |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   | RWA |  |

| Bit   | Name     | Function                      | Description                                                                                                                                                                                                                                                                                                        | Access |
|-------|----------|-------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:20 | Reserved | Reserved                      | -                                                                                                                                                                                                                                                                                                                  | -      |
| 19:16 | TCP      | Timestamp counter pre-scaling | Configure the timestamp and timeout counter time units to 1-16 times the CAN bit time. The active setting value is 0-15, and the actual value used by the hardware is the programmed value plus 1.                                                                                                                 | RWA    |
| 15:2  | Reserved | Reserved                      | -                                                                                                                                                                                                                                                                                                                  | -      |
| 1:0   | TSS      | Timestamp selection           | <ul style="list-style-type: none"> <li>0b00: The timestamp counter value is always 0x0000</li> <li>0b01: The timestamp counter value is incremented according to TCP</li> <li>0b10: Setting prohibited</li> <li>0b11: The timestamp counter value is always 0x0000 (equivalent to the set value "0b00")</li> </ul> | RWA    |

## 37.4.9 MCAN Timestamp Counter Value Register (MCAN\_TSCV)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |     |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0x24   | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15  | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |     |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | TSC |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RWC |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit   | Name     | Function          | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | Access |
|-------|----------|-------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:16 | Reserved | Reserved          | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | -      |
| 15:0  | TSC      | Timestamp counter | <p>The internal timestamp counter value captured when a SOF (Start Of Frame) is received or sent. When MCAN_TSCC.TSS=0b01, the timestamp counter increments by 1 to 16 times the CAN bit time, the specific multiple is determined by the setting value of MCAN_TSCC.TCP. When the count wraps around, the interrupt flag MCAN_IR.TSW will be set to 1. Write operations will clear the counter.</p> <div>  <b>Note</b> <ul style="list-style-type: none"> <li>● "Wraparound" means that the value of the timestamp counter changes from non-zero to zero, not due to a write operation of MCAN_TSCV.</li> <li>● When accessing bytes, writing any byte of this register will cause the timestamp register to be cleared.</li> </ul> </div> | RWC    |

### 37.4.10 MCAN Timeout Counter Configuration Register (MCAN\_TOCC)

For an introduction to the timeout counter, see "[Timeout Counter](#)".

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |          |    |    |    |    |    |   |   |   |   |     |      |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----------|----|----|----|----|----|---|---|---|---|-----|------|---|---|---|---|
| 0x28   | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15       | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5   | 4    | 3 | 2 | 1 | 0 |
| Reset  | 0xFFFF0000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |          |    |    |    |    |    |   |   |   |   |     |      |   |   |   |   |
| Name   | TOP          |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | Reserved |    |    |    |    |    |   |   |   |   | TOS | ETOC |   |   |   |   |
|        |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |          |    |    |    |    |    |   |   |   |   |     |      |   |   |   |   |
| Access | RW           |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |          |    |    |    |    |    |   |   |   |   | RWA | RWA  |   |   |   |   |
|        |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |          |    |    |    |    |    |   |   |   |   |     |      |   |   |   |   |

| Bit   | Name            | Function                | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | Access |
|-------|-----------------|-------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:16 | TOP             | Timeout period          | Start value of the timeout counter (down counter), configure the timeout period                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | RW     |
| 15:3  | <i>Reserved</i> | <i>Reserved</i>         | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | -      |
| 2:1   | TOS             | Timeout selection       | When operating in continuous mode, a write access to MCAN_TOCV will preset the counter to the value configured by MCAN_TOCC.TOP and continue to count down. When the timeout counter is controlled by one of the FIFOs, an empty FIFO will preset the counter to the value configured by MCAN_TOCC.TOP. When the first FIFO element is stored, the counter starts counting down: <ul style="list-style-type: none"> <li>● 0b00: Continuous mode</li> <li>● 0b01: Timeout controlled by the transmit event FIFO</li> <li>● 0b10: Timeout controlled by the receive FIFO0</li> <li>● 0b11: Timeout controlled by the receive FIFO1</li> </ul> | RWA    |
| 0     | ETOC            | Timeout counter enabled | <ul style="list-style-type: none"> <li>● 0b0: Disable</li> <li>● 0b1: Enable</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | RWA    |

## 37.4.11 MCAN Timeout Counter Value Register (MCAN\_TOCV)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |     |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0x2C   | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15  | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x0000FFFF   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |     |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | TOC |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |     |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit   | Name     | Function        | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | Access |
|-------|----------|-----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:16 | Reserved | Reserved        | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | -      |
| 15:0  | TOC      | Timeout counter | <p>The timeout counter decrements by 1 to 16 times the CAN bit time, the specific multiple is determined by the setting value of MCAN_TSCC.TCP. When the counter decrements to zero, the interrupt flag bit MCAN_IR.TOO is set to 1 and the counter stops counting. The start and reset (or re-start) conditions can be set by MCAN_TOCC.TOS.</p> <div>  <b>Note</b><br/>           For byte access, when MCAN_TOCC.TOS=0b00, writing any byte of this register will preset the timeout counter to the value configured by MCAN_TOCC.TOP.         </div> | RW     |

## 37.4.12 MCAN Error Counter Register (MCAN\_ECR)

| Offset | Bit Position |    |    |    |    |    |    |    |     |    |    |    |    |    |    |    |    |     |    |    |    |    |   |   |   |     |   |   |   |   |   |   |  |
|--------|--------------|----|----|----|----|----|----|----|-----|----|----|----|----|----|----|----|----|-----|----|----|----|----|---|---|---|-----|---|---|---|---|---|---|--|
| 0x40   | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23  | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14  | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6   | 5 | 4 | 3 | 2 | 1 | 0 |  |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |     |    |    |    |    |    |    |    |    |     |    |    |    |    |   |   |   |     |   |   |   |   |   |   |  |
| Name   | Reserved     |    |    |    |    |    |    |    | CEL |    |    |    |    |    |    |    | RP | REC |    |    |    |    |   |   |   | TEC |   |   |   |   |   |   |  |
| Access |              |    |    |    |    |    |    |    |     |    |    |    |    |    |    |    | RC |     |    |    |    |    |   |   |   |     |   |   |   |   |   |   |  |

| Bit   | Name     | Function               | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | Access |
|-------|----------|------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:24 | Reserved | Reserved               | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | -      |
| 23:16 | CEL      | CAN error log          | <p>When a CAN protocol error causes the transmit error counter TEC or the receive error counter REC to increment, the error log counter CEL will increment. When the counter value reaches 0xFF, it stops counting. When only the RP bit is set but REC does not change, CEL will not increment. CEL increments after REC and TEC increase.</p> <p>A read access to CEL resets the counter. When the counter value reaches 0xFF, it stops counting, and the interrupt flag MCAN_IR.ELO is set when TEC or REC increases next time.</p> <div>  <b>Note</b><br/>           When accessing by byte, reading byte 2 of the register will clear CEL, and reading byte 3, 1 or 0 has no effect.         </div> | RC     |
| 15    | RP       | Receive error passive  | <ul style="list-style-type: none"> <li>0b0: Receive error counter is below error passive level 128</li> <li>0b1: Receive error counter has reached error passive level 128</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | RO     |
| 14:8  | REC      | Receive error counter  | The actual status of the receive error counter, the count value range is 0-127                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | RO     |
| 7:0   | TEC      | Transmit error counter | The actual status of the send error counter, the count value range is 0-255                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | RO     |

 **Note**

If MCAN\_CCCR.ASM is set to 1, the CAN protocol controller will not increment TEC and REC when a CAN protocol error is detected, but CEL will still increment.

## 37.4.13 MCAN Protocol Status Register (MCAN\_PSR)

| Offset | Bit Position |    |    |    |    |    |    |    |      |    |    |    |    |    |    |    |          |     |      |      |      |      |    |    |    |    |    |    |    |     |    |     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |  |  |
|--------|--------------|----|----|----|----|----|----|----|------|----|----|----|----|----|----|----|----------|-----|------|------|------|------|----|----|----|----|----|----|----|-----|----|-----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|--|--|
| 0x44   | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23   | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15       | 14  | 13   | 12   | 11   | 10   | 9  | 8  | 7  | 6  | 5  | 4  | 3  | 2   | 1  | 0   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |  |  |
| Reset  | 0x00000707   |    |    |    |    |    |    |    |      |    |    |    |    |    |    |    |          |     |      |      |      |      |    |    |    |    |    |    |    |     |    |     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |  |  |
| Name   | Reserved     |    |    |    |    |    |    |    | TDCV |    |    |    |    |    |    |    | Reserved | PXE | RFDF | RBR5 | RESI | DLEC |    |    |    | BO |    | EW | EP | ACT |    | LEC |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |  |  |
| Access |              |    |    |    |    |    |    |    |      |    |    |    |    |    |    |    | RO       | RO  | RO   | RO   | RO   | RO   | RO | RO | RO | RO | RO | RO | RO | RO  | RO | RO  | RO | RO | RO | RO | RO | RO | RO | RO | RO | RO | RO | RO | RO | RO | RO |  |  |
|        |              |    |    |    |    |    |    |    |      |    |    |    |    |    |    |    |          |     |      |      |      |      |    |    |    |    |    |    |    |     |    |     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |  |  |
|        |              |    |    |    |    |    |    |    |      |    |    |    |    |    |    |    |          |     |      |      |      |      |    |    |    |    |    |    |    |     |    |     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |  |  |
|        |              |    |    |    |    |    |    |    |      |    |    |    |    |    |    |    |          |     |      |      |      |      |    |    |    |    |    |    |    |     |    |     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |  |  |

| Bit   | Name     | Function                          | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | Access |
|-------|----------|-----------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:23 | Reserved | Reserved                          | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | -      |
| 22:16 | TDCV     | Transmit delay compensation value | The actual transmit delay compensation value, the active setting value is 0-127 mtq.<br>The position of the second sampling point (Secondary Sample Point, SSP) is determined by the sum of the delay value measured between the MCAN_TX pin and the MCAN_RX pin and MCAN_TDCR.TDCO. In the data phase, the SSP position is the number of mtqs between the start point of the transmit bit and the second sampling point.                                                                                                 | RO     |
| 15    | Reserved | Reserved                          | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | -      |
| 14    | PXE      | Protocol exception event          | <ul style="list-style-type: none"> <li>0b0: No protocol exception event has occurred since the last read access</li> <li>0b1: A protocol exception event has occurred</li> </ul> <div>  <b>Note</b><br/>           During byte access, reading byte 0 of the register will clear this bit, reading bytes 3/2/1 has no effect.         </div>                                                                                           | RO     |
| 13    | RFDF     | Received CAN FD message           | This bit is set regardless of receive filtering. <ul style="list-style-type: none"> <li>0b0: No CAN FD formatted message has been received since the last time it was cleared by the CPU</li> <li>0b1: Received CAN FD formatted message with FDF flag set to 1</li> </ul> <div>  <b>Note</b><br/>           During byte access, reading byte 0 of the register will clear this bit, reading bytes 3/2/1 has no effect.         </div> | RO     |

| Bit  | Name | Function                                     | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | Access |
|------|------|----------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 12   | RBRS | BRS flag of the last received CAN FD message | <p>This bit is set together with RFDF and is independent of receive filtering.</p> <ul style="list-style-type: none"> <li>0b0: BRS flag of the last received CAN FD message is not set</li> <li>0b1: BRS flag of the last received CAN FD message is set</li> </ul> <div>  <b>Note</b><br/>           During byte access, reading byte 0 of the register will clear this bit, reading bytes 3/2/1 has no effect.         </div>                        | RO     |
| 11   | RESI | ESI flag of the last received CAN FD message | <p>This bit is set in conjunction with RFDF, regardless of the acceptance filter.</p> <ul style="list-style-type: none"> <li>0b0: ESI flag of the last received CAN FD message is not set</li> <li>0b1: ESI flag of the last received CAN FD message is set</li> </ul> <div>  <b>Note</b><br/>           During byte access, reading byte 0 of the register will clear this bit, reading bytes 3/2/1 has no effect.         </div>                     | RO     |
| 10:8 | DLEC | Data phase last error code                   | <p>The type of the last error that occurred in the data phase of a CAN FD frame with the BRS flag set. The error code is the same as LEC. DLEC will be cleared when there are no errors in the transmission (reception or transmission) of a CAN FD frame with the BRS flag set.</p> <div>  <b>Note</b><br/>           For byte access, reading byte 0 of the register will set this bit to 0b111, reading bytes 3/2/1 has no effect.         </div> | RO     |
| 7    | BO   | Bus_Off state                                | <ul style="list-style-type: none"> <li>0b0: MCAN is not in Bus_Off state</li> <li>0b1: MCAN is in Bus_Off state</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                              | RO     |
| 6    | EW   | Warning status                               | <ul style="list-style-type: none"> <li>0b0: Both error counters are less than the error warning (Error_Warning) upper limit of 96</li> <li>0b1: At least one error counter has reached the error warning (Error_Warning) upper limit of 96</li> </ul>                                                                                                                                                                                                                                                                                   | RO     |
| 5    | EP   | Error Passive                                | <ul style="list-style-type: none"> <li>0b0: MCAN is in Error_Active state. MCAN participates in bus communication normally and sends active error flag when an error is detected</li> <li>0b1: MCAN is in Error_Passive state</li> </ul>                                                                                                                                                                                                                                                                                                | RO     |

| Bit | Name | Function             | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | Access |
|-----|------|----------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 4:3 | ACT  | Communication status | <p>Monitor the communication status of MCAN:</p> <ul style="list-style-type: none"> <li>● 0b00: Synchronizing, the node is synchronizing CAN communication</li> <li>● 0b01: Idle, the node is neither a receiving node nor a sending node</li> <li>● 0b10: Receiving, the node works as a receiving node</li> <li>● 0b11: Sending, the node works as a sending node</li> </ul> <div>  <b>Note</b><br/>           Protocol abnormal events will set this bit to 0b00.         </div>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | RO     |
| 2:0 | LEC  | Last error code      | <ul style="list-style-type: none"> <li>● 0b000: No Error. No error has occurred since LEC was reset by successful reception or transmission</li> <li>● 0b001: Stuff Error. More than 5 identical bits appeared in a part of the received message, which is not disabled</li> <li>● 0b010: Form Error. The fixed format part of the received frame has a format error</li> <li>● 0b011: ACK Error. The message sent by MCAN is not acknowledged by other nodes</li> <li>● 0b100: Bit1 Error. During the message transmission process (except Arbitration field), the device sent a recessive level (logical value 1), but the detected bus level is dominant (logical value 0)</li> <li>● 0b101: Bit0 Error. During the message transmission process (including ACK bit, active error flag, overload flag), the device sent a dominant level (logical value 0 of data or ID), but the detected bus level is recessive (logical value 1). This state is set each time an 11-bit recessive bit sequence is detected during a Bus_Off recovery. This allows the CPU to monitor the progress of the bus recovery sequence (indicating that the bus is not stuck in a dominant state or continuously disturbed)</li> <li>● 0b110: CRC Error. The CRC of the received message is incorrect. The received CRC does not match the CRC calculated using the received data</li> <li>● 0b111: No Change. Any read access to the Protocol Status Register will reinitialize the LEC to 7. When the LEC shows a value of 7, it means that no CAN bus events have been detected since the last CPU read access to the Protocol Status Register.</li> </ul> | RS     |



| Bit | Name | Function        | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | Access |
|-----|------|-----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 2:0 | LEC  | Last error code |  <b>Note</b> <ol style="list-style-type: none"> <li>1. When a CAN FD frame with the BRS flag set has reached the Data phase, the next CAN event (error or active frame) will be shown in the DLEC instead of the LEC. An error with fixed stuff bits in the CAN FD CRC sequence will be shown as a Form Error instead of a Stuff Error.</li> <li>2. The Bus_Off recovery sequence (see ISO11898-1:2015) cannot be shortened by setting or clearing MCAN_CCCR.INIT. If the device enters the Bus_Off state, it will automatically set MCAN_CCCR.INIT, stopping all bus activity. Once MCAN_CCCR.INIT is cleared by the CPU, the device will wait for 129 bus idles (129*11 consecutive recessive bits) before resuming normal operation. At the end of the Bus_Off recovery sequence, the error management counter will be reset. During the waiting period after clearing MCAN_CCCR.INIT, each time an 11-bit recessive sequence is detected, a Bit0 error code is written to MCAN_PSR.LEC, allowing the CPU to check at any time whether the CAN bus is in a dominant state or is continuously disturbed, and monitor the Bus_Off recovery sequence. MCAN_ECR.REC is used to count these sequences.</li> <li>3. For byte operations, reading byte 0 of the register will set this bit to 0b111, and reading bytes 3/2/1 has no effect.</li> </ol> | RS     |

## 37.4.14 MCAN Transmit Delay Compensation Register (MCAN\_TDCR)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |      |    |    |    |          |      |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|------|----|----|----|----------|------|---|---|---|---|---|---|---|---|---|---|
| 0x48   | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15   | 14 | 13 | 12 | 11       | 10   | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |      |    |    |    |          |      |   |   |   |   |   |   |   |   |   |   |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | TDCO |    |    |    | Reserved | TDCF |   |   |   |   |   |   |   |   |   |   |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RWA  |    |    |    |          | RWA  |   |   |   |   |   |   |   |   |   |   |

| Bit   | Name     | Function                                        | Description                                                                                                                                                                                                                                                                                              | Access |
|-------|----------|-------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:15 | Reserved | Reserved                                        | -                                                                                                                                                                                                                                                                                                        | -      |
| 14:8  | TDCO     | Transmit delay compensation SSP offset          | The offset value refers to the distance between the delay measured from the MCAN_TX pin to the MCAN_RX pin and the second sampling point (SSP). The active setting value is 0-127mtq                                                                                                                     | RWA    |
| 7     | Reserved | Reserved                                        | -                                                                                                                                                                                                                                                                                                        | -      |
| 6:0   | TDCF     | Transmit delay-compensation filter window width | Defines the minimum value of the SSP position. For the measurement of transmit delay, the dominant edge on the MCAN_RX pin that would cause the SSP position to advance will be ignored. This function is enabled when the TDCF setting value is greater than TDCO. The active setting value is 0-127mtq | RWA    |

### 37.4.15 MCAN Interrupt Register (MCAN\_IR)

When one of the conditions in the following list is detected, the corresponding flag bit will be set. The flag bit remains set until the host clears it. **All flag bits write 1 to clear flag, write 0 has no effect.** A hard reset will clear the register. Whether an interrupt is generated is controlled by register MCAN\_IE; which interrupt line generates the interrupt is controlled by register MCAN\_ILS.

| Offset | Bit Position |     |     |     |     |    |    |    |     |     |     |     |     |      |     |      |      |      |      |     |     |    |     |      |      |      |      |      |      |      |      |    |
|--------|--------------|-----|-----|-----|-----|----|----|----|-----|-----|-----|-----|-----|------|-----|------|------|------|------|-----|-----|----|-----|------|------|------|------|------|------|------|------|----|
| 0x50   | 31           | 30  | 29  | 28  | 27  | 26 | 25 | 24 | 23  | 22  | 21  | 20  | 19  | 18   | 17  | 16   | 15   | 14   | 13   | 12  | 11  | 10 | 9   | 8    | 7    | 6    | 5    | 4    | 3    | 2    | 1    | 0  |
| Reset  | 0x00000000   |     |     |     |     |    |    |    |     |     |     |     |     |      |     |      |      |      |      |     |     |    |     |      |      |      |      |      |      |      |      |    |
| Name   | Reserved     | ARA | PED | PEA | WDI | BO | EW | EP | ELO | BEU | BEC | DRX | TOO | MRAF | TSW | TEFL | TEFF | TEFW | TEFN | TFE | TCF | TC | HPM | RF1L | RF1F | RF1W | RF1N | RF0L | RF0F | RF0W | RF0N |    |
| Access |              | RW  | RW  | RW  | RW  | RW | RW | RW | RW  | RW  | RW  | RW  | RW  | RW   | RW  | RW   | RW   | RW   | RW   | RW  | RW  | RW | RW  | RW   | RW   | RW   | RW   | RW   | RW   | RW   | RW   | RW |

| Bit   | Name     | Function                                                  | Description                                                                                                                                                                            | Access |
|-------|----------|-----------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:30 | Reserved | Reserved                                                  | -                                                                                                                                                                                      | -      |
| 29    | ARA      | Access to re-served address                               | <ul style="list-style-type: none"> <li>0b0: No access to reserved address</li> <li>0b1: Access to reserved address</li> </ul>                                                          | RW     |
| 28    | PED      | Data phase protocol error (using data bit time)           | <ul style="list-style-type: none"> <li>0b0: No data phase protocol error detected</li> <li>0b1: Data phase protocol error detected (MCAN_PSR.DLEC≠0b000/0b111)</li> </ul>              | RW     |
| 27    | PEA      | Arbitration phase protocol error (using nominal bit time) | <ul style="list-style-type: none"> <li>0b0: No arbitration phase protocol error detected</li> <li>0b1: Arbitration phase protocol error detected (MCAN_PSR.LEC≠0b000/0b111)</li> </ul> | RW     |
| 26    | WDI      | Watchdog interrupt                                        | <ul style="list-style-type: none"> <li>0b0: No message RAM watchdog event occurred</li> <li>0b1: A message RAM watchdog event occurred due to the loss of the READY signal</li> </ul>  | RW     |
| 25    | BO       | Bus_Off state                                             | <ul style="list-style-type: none"> <li>0b0: Bus_Off status has not changed</li> <li>0b1: Bus_Off status has changed</li> </ul>                                                         | RW     |
| 24    | EW       | Warning status                                            | <ul style="list-style-type: none"> <li>0b0: Error_Warning status has not changed</li> <li>0b1: Error_Warning status has changed</li> </ul>                                             | RW     |
| 23    | EP       | Error passive                                             | <ul style="list-style-type: none"> <li>0b0: Error_Passive state has not changed</li> <li>0b1: Error_Passive state has changed</li> </ul>                                               | RW     |
| 22    | ELO      | Error record overflow                                     | <ul style="list-style-type: none"> <li>0b0: CAN error record counter has not overflowed</li> <li>0b1: CAN error record counter has overflowed</li> </ul>                               | RW     |

| Bit | Name | Function                                   | Description                                                                                                                                                                                                                                                                                                                                                                                                   | Access |
|-----|------|--------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 21  | BEU  | Bit error uncorrected                      | <ul style="list-style-type: none"> <li>0b0: No bit error detected when reading from the message RAM</li> <li>0b1: Bit error detected and uncorrected when reading from the message RAM when MCANRAM_CKCR.ECCMOD is not 0</li> </ul> <div>  <b>Note</b><br/>           BEU is set while MCAN_CCCR.INIT is set.         </div> | RW     |
| 20  | BEC  | Bit error corrected                        | <ul style="list-style-type: none"> <li>0b0: No bit error detected when reading from the message RAM</li> <li>0b1: When MCANRAM_CKCR.ECCMOD is 0b10 or 0b11, a bit error was detected and corrected when reading from the message RAM</li> </ul>                                                                                                                                                               | RW     |
| 19  | DRX  | Message stored in dedicated receive buffer | <ul style="list-style-type: none"> <li>0b0: Receive buffer not updated</li> <li>0b1: At least one message has been stored in receive buffer</li> </ul> <div>  <b>Note</b><br/>           This flag is set whenever a received message has been stored in dedicated receive buffer.         </div>                            | RW     |
| 18  | TOO  | Timeout occurred                           | <ul style="list-style-type: none"> <li>0b0: No timeout occurred</li> <li>0b1: Timeout occurred</li> </ul>                                                                                                                                                                                                                                                                                                     | RW     |

| Bit | Name | Function                              | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | Access |
|-----|------|---------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 17  | MRAF | Message RAM access failure            | <ul style="list-style-type: none"> <li>● 0b0: No message RAM access failure occurred</li> <li>● 0b1: Message RAM access failure occurred</li> </ul> <p>Set during receive processing:</p> <ul style="list-style-type: none"> <li>● Receive filtering or storage of received message has not been completed before the arbitration field of the subsequent message is received. In this case, receive filtering or message storage is terminated and the receive processing unit starts processing the subsequent message.</li> <li>● Failed to write a message to the message RAM, message storage is terminated.</li> </ul> <div style="border: 1px solid black; padding: 5px; margin-top: 10px;">  <b>Note</b><br/> In both cases, the FIFO put index is not updated, the new data flag of the dedicated receive buffer is not set, and the partially stored message is overwritten when the next message is stored at this location. </div> <p>Set condition in the send processing:</p> <ul style="list-style-type: none"> <li>● The send processing unit cannot read the message from the message RAM in time. In this case, the message transmission will be terminated. In the case of a failure in the send processing unit access, the MCAN is switched to the restricted working mode (see "<a href="#">Restricted Working Mode</a>" for details). To exit restricted operating mode, the host CPU must reset MCAN_CCCR.ASM.</li> </ul> | RW     |
| 16  | TSW  | Timestamp wrap                        | <ul style="list-style-type: none"> <li>● 0b0: Timestamp counter has not wrapped</li> <li>● 0b1: Timestamp counter has wrapped</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | RW     |
| 15  | TEFL | Transmit event FIFO element lost      | <ul style="list-style-type: none"> <li>● 0b0: Transmit event FIFO element has not been lost</li> <li>● 0b1: Transmit event FIFO element has been lost; this bit is also set when writing to a transmit event FIFO with a space size of 0</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | RW     |
| 14  | TEFF | Transmit event FIFO is full           | <ul style="list-style-type: none"> <li>● 0b0: Transmit event FIFO is not full</li> <li>● 0b1: Transmit event FIFO is full</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | RW     |
| 13  | TEFW | Transmit event FIFO reaches watermark | <ul style="list-style-type: none"> <li>● 0b0: Transmit event FIFO fill level is below watermark</li> <li>● 0b1: Transmit event FIFO fill level reaches watermark</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | RW     |
| 12  | TEFN | Transmit event FIFO new entry         | <ul style="list-style-type: none"> <li>● 0b0: Transmit event FIFO unchanged</li> <li>● 0b1: Transmit processing unit writes to transmit event FIFO element</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | RW     |
| 11  | TFE  | Transmit FIFO empty                   | <ul style="list-style-type: none"> <li>● 0b0: Transmit FIFO is not empty</li> <li>● 0b1: Transmit FIFO is empty</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | RW     |

| Bit | Name | Function                        | Description                                                                                                                                                                                          | Access |
|-----|------|---------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 10  | TCF  | Send cancel completed           | <ul style="list-style-type: none"> <li>0b0: Send cancel not completed</li> <li>0b1: Send cancel completed</li> </ul>                                                                                 | RW     |
| 9   | TC   | Sending completed               | <ul style="list-style-type: none"> <li>0b0: Sending not completed</li> <li>0b1: Sending completed</li> </ul>                                                                                         | RW     |
| 8   | HPM  | High priority message           | <ul style="list-style-type: none"> <li>0b0: High priority message not received</li> <li>0b1: High priority message received</li> </ul>                                                               | RW     |
| 7   | RF1L | Receive FIFO1 message lost      | <ul style="list-style-type: none"> <li>0b0: Receive FIFO1 message not lost</li> <li>0b1: Receive FIFO1 message lost; this bit is also set when writing to receive FIFO1 with space size 0</li> </ul> | RW     |
| 6   | RF1F | Receive FIFO1 is full           | <ul style="list-style-type: none"> <li>0b0: Receive FIFO1 not full</li> <li>0b1: Receive FIFO1 full</li> </ul>                                                                                       | RW     |
| 5   | RF1W | Receive FIFO1 reaches watermark | <ul style="list-style-type: none"> <li>0b0: Receive FIFO1 fill level is below watermark</li> <li>0b1: Receive FIFO1 fill level reaches watermark</li> </ul>                                          | RW     |
| 4   | RF1N | Receive new message in FIFO1    | <ul style="list-style-type: none"> <li>0b0: No new message is written into FIFO1</li> <li>0b1: New message is written into FIFO1</li> </ul>                                                          | RW     |
| 3   | RF0L | Receive FIFO0 message lost      | <ul style="list-style-type: none"> <li>0b0: Receive FIFO0 message not lost</li> <li>0b1: Receive FIFO0 message lost; this bit is also set when writing to receive FIFO0 with space size 0</li> </ul> | RW     |
| 2   | RF0F | Receive FIFO0 is full           | <ul style="list-style-type: none"> <li>0b0: Receive FIFO0 not full</li> <li>0b1: Receive FIFO0 full</li> </ul>                                                                                       | RW     |
| 1   | RF0W | Receive FIFO0 reaches watermark | <ul style="list-style-type: none"> <li>0b0: Receive FIFO0 fill level is below watermark</li> <li>0b1: Receive FIFO0 fill level reaches watermark</li> </ul>                                          | RW     |
| 0   | RF0N | Receive new message in FIFO0    | <ul style="list-style-type: none"> <li>0b0: No new message is written into FIFO0</li> <li>0b1: New message is written into FIFO0</li> </ul>                                                          | RW     |

### 37.4.16 MCAN Interrupt Enable Register (MCAN\_IE)

The settings in the interrupt enable register determine which state changes in the interrupt register (MCAN\_IR) will be indicated on the interrupt line.

| Offset | Bit Position |      |      |      |      |     |     |     |      |      |      |      |      |       |      |       |       |       |       |      |      |     |      |       |       |       |       |       |       |       |       |    |
|--------|--------------|------|------|------|------|-----|-----|-----|------|------|------|------|------|-------|------|-------|-------|-------|-------|------|------|-----|------|-------|-------|-------|-------|-------|-------|-------|-------|----|
| 0x54   | 31           | 30   | 29   | 28   | 27   | 26  | 25  | 24  | 23   | 22   | 21   | 20   | 19   | 18    | 17   | 16    | 15    | 14    | 13    | 12   | 11   | 10  | 9    | 8     | 7     | 6     | 5     | 4     | 3     | 2     | 1     | 0  |
| Reset  | 0x00000000   |      |      |      |      |     |     |     |      |      |      |      |      |       |      |       |       |       |       |      |      |     |      |       |       |       |       |       |       |       |       |    |
| Name   | Reserved     | ARAE | PEDE | PEAE | WDIE | BOE | EWE | EPE | ELOE | BEUE | BECE | DRXE | TOOE | MRAFE | TSWE | TEFLE | TEFFE | TEFWE | TEFNE | TTEE | TCFE | TCE | HPME | RF1LE | RF1FE | RF1WE | RF1NE | RF0LE | RF0FE | RF0WE | RF0NE |    |
| Access |              | RW   | RW   | RW   | RW   | RW  | RW  | RW  | RW   | RW   | RW   | RW   | RW   | RW    | RW   | RW    | RW    | RW    | RW    | RW   | RW   | RW  | RW   | RW    | RW    | RW    | RW    | RW    | RW    | RW    | RW    | RW |

| Bit   | Name     | Function                                           | Description                                                                                             | Access |
|-------|----------|----------------------------------------------------|---------------------------------------------------------------------------------------------------------|--------|
| 31:30 | Reserved | Reserved                                           | -                                                                                                       | -      |
| 29    | ARAE     | Access re-served address interrupt enabled         | <ul style="list-style-type: none"> <li>0b0: Disable interrupt</li> <li>0b1: Enable interrupt</li> </ul> | RW     |
| 28    | PEDE     | Data phase protocol error interrupt enabled        | <ul style="list-style-type: none"> <li>0b0: Disable interrupt</li> <li>0b1: Enable interrupt</li> </ul> | RW     |
| 27    | PEAE     | Arbitration phase protocol error interrupt enabled | <ul style="list-style-type: none"> <li>0b0: Disable interrupt</li> <li>0b1: Enable interrupt</li> </ul> | RW     |
| 26    | WDIE     | Watchdog interrupt enable                          | <ul style="list-style-type: none"> <li>0b0: Disable interrupt</li> <li>0b1: Enable interrupt</li> </ul> | RW     |
| 25    | BOE      | Bus_Off status interrupt enabled                   | <ul style="list-style-type: none"> <li>0b0: Disable interrupt</li> <li>0b1: Enable interrupt</li> </ul> | RW     |
| 24    | EWE      | Warning status interrupt enabled                   | <ul style="list-style-type: none"> <li>0b0: Disable interrupt</li> <li>0b1: Enable interrupt</li> </ul> | RW     |
| 23    | EPE      | Error passive interrupt enabled                    | <ul style="list-style-type: none"> <li>0b0: Disable interrupt</li> <li>0b1: Enable interrupt</li> </ul> | RW     |
| 22    | ELOE     | Error record overflow interrupt enabled            | <ul style="list-style-type: none"> <li>0b0: Disable interrupt</li> <li>0b1: Enable interrupt</li> </ul> | RW     |
| 21    | BEUE     | Bit Error uncorrected interrupt enabled            | <ul style="list-style-type: none"> <li>0b0: Disable interrupt</li> <li>0b1: Enable interrupt</li> </ul> | RW     |

| Bit | Name  | Function                                                      | Description                                                                                                 | Access |
|-----|-------|---------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------|--------|
| 20  | BECE  | Bit error corrected interrupt enabled                         | <ul style="list-style-type: none"> <li>● 0b0: Disable interrupt</li> <li>● 0b1: Enable interrupt</li> </ul> | RW     |
| 19  | DRXE  | Message storage to dedicated receive buffer interrupt enabled | <ul style="list-style-type: none"> <li>● 0b0: Disable interrupt</li> <li>● 0b1: Enable interrupt</li> </ul> | RW     |
| 18  | TOOE  | Timeout interrupt enabled                                     | <ul style="list-style-type: none"> <li>● 0b0: Disable interrupt</li> <li>● 0b1: Enable interrupt</li> </ul> | RW     |
| 17  | MRAFE | Message RAM access failure interrupt enabled                  | <ul style="list-style-type: none"> <li>● 0b0: Disable interrupt</li> <li>● 0b1: Enable interrupt</li> </ul> | RW     |
| 16  | TSWE  | Timestamp wrap interrupt enabled                              | <ul style="list-style-type: none"> <li>● 0b0: Disable interrupt</li> <li>● 0b1: Enable interrupt</li> </ul> | RW     |
| 15  | TEFLE | Transmit event FIFO element lost interrupt enabled            | <ul style="list-style-type: none"> <li>● 0b0: Disable interrupt</li> <li>● 0b1: Enable interrupt</li> </ul> | RW     |
| 14  | TEFFE | Transmit event FIFO full interrupt enabled                    | <ul style="list-style-type: none"> <li>● 0b0: Disable interrupt</li> <li>● 0b1: Enable interrupt</li> </ul> | RW     |
| 13  | TEFWE | Transmit event FIFO reaches water level interrupt enabled     | <ul style="list-style-type: none"> <li>● 0b0: Disable interrupt</li> <li>● 0b1: Enable interrupt</li> </ul> | RW     |
| 12  | TEFNE | Transmit event FIFO new entry interrupt enabled               | <ul style="list-style-type: none"> <li>● 0b0: Disable interrupt</li> <li>● 0b1: Enable interrupt</li> </ul> | RW     |
| 11  | TFEE  | Transmit FIFO empty interrupt enabled                         | <ul style="list-style-type: none"> <li>● 0b0: Disable interrupt</li> <li>● 0b1: Enable interrupt</li> </ul> | RW     |
| 10  | TCFE  | Transmit cancel completed interrupt enabled                   | <ul style="list-style-type: none"> <li>● 0b0: Disable interrupt</li> <li>● 0b1: Enable interrupt</li> </ul> | RW     |
| 9   | TCE   | Transmit complete interrupt enabled                           | <ul style="list-style-type: none"> <li>● 0b0: Disable interrupt</li> <li>● 0b1: Enable interrupt</li> </ul> | RW     |
| 8   | HPME  | High priority message interrupt enabled                       | <ul style="list-style-type: none"> <li>● 0b0: Disable interrupt</li> <li>● 0b1: Enable interrupt</li> </ul> | RW     |



| Bit | Name  | Function                                            | Description                                                                                                 | Access |
|-----|-------|-----------------------------------------------------|-------------------------------------------------------------------------------------------------------------|--------|
| 7   | RF1LE | Receive FIFO1 message lost interrupt enabled        | <ul style="list-style-type: none"> <li>● 0b0: Disable interrupt</li> <li>● 0b1: Enable interrupt</li> </ul> | RW     |
| 6   | RF1FE | Receive FIFO1 full interrupt enabled                | <ul style="list-style-type: none"> <li>● 0b0: Disable interrupt</li> <li>● 0b1: Enable interrupt</li> </ul> | RW     |
| 5   | RF1WE | Receive FIFO1 reaches water level interrupt enabled | <ul style="list-style-type: none"> <li>● 0b0: Disable interrupt</li> <li>● 0b1: Enable interrupt</li> </ul> | RW     |
| 4   | RF1NE | Receive FIFO1 new message interrupt enabled         | <ul style="list-style-type: none"> <li>● 0b0: Disable interrupt</li> <li>● 0b1: Enable interrupt</li> </ul> | RW     |
| 3   | RF0LE | Receive FIFO0 message lost interrupt enabled        | <ul style="list-style-type: none"> <li>● 0b0: Disable interrupt</li> <li>● 0b1: Enable interrupt</li> </ul> | RW     |
| 2   | RF0FE | Receive FIFO0 full interrupt enabled                | <ul style="list-style-type: none"> <li>● 0b0: Disable interrupt</li> <li>● 0b1: Enable interrupt</li> </ul> | RW     |
| 1   | RF0WE | Receive FIFO0 reaches water level interrupt enabled | <ul style="list-style-type: none"> <li>● 0b0: Disable interrupt</li> <li>● 0b1: Enable interrupt</li> </ul> | RW     |
| 0   | RF0NE | Receive FIFO0 new message interrupt enabled         | <ul style="list-style-type: none"> <li>● 0b0: Disable interrupt</li> <li>● 0b1: Enable interrupt</li> </ul> | RW     |

### 37.4.17 MCAN Interrupt Line Selection Register (MCAN\_ILS)

The interrupt line select register assigns the interrupt generated by a specific interrupt flag in the interrupt register (MCAN\_IR) to one of the two interrupt lines. To generate an interrupt, the corresponding interrupt lines must be enabled via MCAN\_ILE.EINT0 and MCAN\_ILE.EINT1.

| Offset | Bit Position |      |      |      |      |     |     |     |      |      |      |      |      |       |      |       |       |       |       |      |      |     |      |       |       |       |       |       |       |       |       |    |
|--------|--------------|------|------|------|------|-----|-----|-----|------|------|------|------|------|-------|------|-------|-------|-------|-------|------|------|-----|------|-------|-------|-------|-------|-------|-------|-------|-------|----|
| 0x58   | 31           | 30   | 29   | 28   | 27   | 26  | 25  | 24  | 23   | 22   | 21   | 20   | 19   | 18    | 17   | 16    | 15    | 14    | 13    | 12   | 11   | 10  | 9    | 8     | 7     | 6     | 5     | 4     | 3     | 2     | 1     | 0  |
| Reset  | 0x00000000   |      |      |      |      |     |     |     |      |      |      |      |      |       |      |       |       |       |       |      |      |     |      |       |       |       |       |       |       |       |       |    |
| Name   | Reserved     | ARAL | PEDL | PEAL | WDIL | BOL | EWL | EPL | ELOL | BEUL | BECL | DRXL | TOOL | MRAFL | TSWL | TEFLL | TEFFL | TEFWL | TEFNL | TFEL | TCFL | TCL | HPML | RF1LL | RF1FL | RF1WL | RF1NL | RF0LL | RF0FL | RF0WL | RF0NL |    |
| Access |              | RW   | RW   | RW   | RW   | RW  | RW  | RW  | RW   | RW   | RW   | RW   | RW   | RW    | RW   | RW    | RW    | RW    | RW    | RW   | RW   | RW  | RW   | RW    | RW    | RW    | RW    | RW    | RW    | RW    | RW    | RW |

| Bit   | Name     | Function                                        | Description                                                                                                          | Access |
|-------|----------|-------------------------------------------------|----------------------------------------------------------------------------------------------------------------------|--------|
| 31:30 | Reserved | Reserved                                        | -                                                                                                                    | -      |
| 29    | ARAL     | Access re-served address interrupt line         | <ul style="list-style-type: none"> <li>0b0: Select interrupt line 0</li> <li>0b1: Select interrupt line 1</li> </ul> | RW     |
| 28    | PEDL     | Data phase protocol error interrupt line        | <ul style="list-style-type: none"> <li>0b0: Select interrupt line 0</li> <li>0b1: Select interrupt line 1</li> </ul> | RW     |
| 27    | PEAL     | Arbitration phase protocol error interrupt line | <ul style="list-style-type: none"> <li>0b0: Select interrupt line 0</li> <li>0b1: Select interrupt line 1</li> </ul> | RW     |
| 26    | WDIL     | Watchdog in-terrupt line                        | <ul style="list-style-type: none"> <li>0b0: Select interrupt line 0</li> <li>0b1: Select interrupt line 1</li> </ul> | RW     |
| 25    | BOL      | Bus_Off state interrupt line                    | <ul style="list-style-type: none"> <li>0b0: Select interrupt line 0</li> <li>0b1: Select interrupt line 1</li> </ul> | RW     |
| 24    | EWL      | Warning status interrupt line                   | <ul style="list-style-type: none"> <li>0b0: Select interrupt line 0</li> <li>0b1: Select interrupt line 1</li> </ul> | RW     |
| 23    | EPL      | Error passive interrupt line                    | <ul style="list-style-type: none"> <li>0b0: Select interrupt line 0</li> <li>0b1: Select interrupt line 1</li> </ul> | RW     |
| 22    | ELOL     | Error record overflow inter-rupt line           | <ul style="list-style-type: none"> <li>0b0: Select interrupt line 0</li> <li>0b1: Select interrupt line 1</li> </ul> | RW     |
| 21    | BEUL     | Bit error uncor-rected interrupt line           | <ul style="list-style-type: none"> <li>0b0: Select interrupt line 0</li> <li>0b1: Select interrupt line 1</li> </ul> | RW     |
| 20    | BECL     | BECL Bit Error Corrected In-terrupt Line        | <ul style="list-style-type: none"> <li>0b0: Select interrupt line 0</li> <li>0b1: Select interrupt line 1</li> </ul> | RW     |

| Bit | Name  | Function                                                         | Description                                                                                                              | Access |
|-----|-------|------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------|--------|
| 19  | DRXL  | Messages are stored in a dedicated receive buffer interrupt line | <ul style="list-style-type: none"> <li>● 0b0: Select interrupt line 0</li> <li>● 0b1: Select interrupt line 1</li> </ul> | RW     |
| 18  | TOOL  | Timeout interrupt line occurred.                                 | <ul style="list-style-type: none"> <li>● 0b0: Select interrupt line 0</li> <li>● 0b1: Select interrupt line 1</li> </ul> | RW     |
| 17  | MRAFL | Message RAM access failure interrupt line                        | <ul style="list-style-type: none"> <li>● 0b0: Select interrupt line 0</li> <li>● 0b1: Select interrupt line 1</li> </ul> | RW     |
| 16  | TSWL  | Timestamp wrap-around interrupt line                             | <ul style="list-style-type: none"> <li>● 0b0: Select interrupt line 0</li> <li>● 0b1: Select interrupt line 1</li> </ul> | RW     |
| 15  | TEFLL | Transmission Event FIFO element lost interrupt line              | <ul style="list-style-type: none"> <li>● 0b0: Select interrupt line 0</li> <li>● 0b1: Select interrupt line 1</li> </ul> | RW     |
| 14  | TEFFL | Transmission Event FIFO full interrupt line                      | <ul style="list-style-type: none"> <li>● 0b0: Select interrupt line 0</li> <li>● 0b1: Select interrupt line 1</li> </ul> | RW     |
| 13  | TEFWL | Transmission Event FIFO watermark reached interrupt line         | <ul style="list-style-type: none"> <li>● 0b0: Select interrupt line 0</li> <li>● 0b1: Select interrupt line 1</li> </ul> | RW     |
| 12  | TEFNL | Transmission event FIFO new entry interrupt line                 | <ul style="list-style-type: none"> <li>● 0b0: Select interrupt line 0</li> <li>● 0b1: Select interrupt line 1</li> </ul> | RW     |
| 11  | TFEL  | Transmission FIFO empty interrupt line                           | <ul style="list-style-type: none"> <li>● 0b0: Select interrupt line 0</li> <li>● 0b1: Select interrupt line 1</li> </ul> | RW     |
| 10  | TCFL  | Transmission cancellation finished interrupt line                | <ul style="list-style-type: none"> <li>● 0b0: Select interrupt line 0</li> <li>● 0b1: Select interrupt line 1</li> </ul> | RW     |
| 9   | TCL   | Transmission complete interrupt line                             | <ul style="list-style-type: none"> <li>● 0b0: Select interrupt line 0</li> <li>● 0b1: Select interrupt line 1</li> </ul> | RW     |
| 8   | HPML  | High priority message interrupt line                             | <ul style="list-style-type: none"> <li>● 0b0: Select interrupt line 0</li> <li>● 0b1: Select interrupt line 1</li> </ul> | RW     |
| 7   | RF1LL | Receive FIFO1 message lost interrupt line                        | <ul style="list-style-type: none"> <li>● 0b0: Select interrupt line 0</li> <li>● 0b1: Select interrupt line 1</li> </ul> | RW     |

| Bit | Name  | Function                                  | Description                                                                                                              | Access |
|-----|-------|-------------------------------------------|--------------------------------------------------------------------------------------------------------------------------|--------|
| 6   | RF1FL | Receive FIFO1 full interrupt line         | <ul style="list-style-type: none"> <li>● 0b0: Select interrupt line 0</li> <li>● 0b1: Select interrupt line 1</li> </ul> | RW     |
| 5   | RF1WL | Receive FIFO1 watermark interrupt line    | <ul style="list-style-type: none"> <li>● 0b0: Select interrupt line 0</li> <li>● 0b1: Select interrupt line 1</li> </ul> | RW     |
| 4   | RF1NL | Receive FIFO1 new message interrupt line  | <ul style="list-style-type: none"> <li>● 0b0: Select interrupt line 0</li> <li>● 0b1: Select interrupt line 1</li> </ul> | RW     |
| 3   | RF0LL | Receive FIFO0 message lost interrupt line | <ul style="list-style-type: none"> <li>● 0b0: Select interrupt line 0</li> <li>● 0b1: Select interrupt line 1</li> </ul> | RW     |
| 2   | RF0FL | Receive FIFO0 full interrupt line         | <ul style="list-style-type: none"> <li>● 0b0: Select interrupt line 0</li> <li>● 0b1: Select interrupt line 1</li> </ul> | RW     |
| 1   | RF0WL | Receive FIFO0 watermark interrupt line    | <ul style="list-style-type: none"> <li>● 0b0: Select interrupt line 0</li> <li>● 0b1: Select interrupt line 1</li> </ul> | RW     |
| 0   | RF0NL | Receive FIFO0 new message interrupt line  | <ul style="list-style-type: none"> <li>● 0b0: Select interrupt line 0</li> <li>● 0b1: Select interrupt line 1</li> </ul> | RW     |

### 37.4.18 MCAN Interrupt Line Enable Register (MCAN\_ILE)

Each of the two interrupt lines connected to the CPU can be enabled or disabled by setting EINT0 and EINT1, respectively.

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |       |       |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|-------|-------|
| 0x5C   | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1     | 0     |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |       |       |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   | EINT1 | EINT0 |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   | RW    | RW    |

| Bit  | Name     | Function                 | Description                                                                         | Access |
|------|----------|--------------------------|-------------------------------------------------------------------------------------|--------|
| 31:2 | Reserved | Reserved                 | -                                                                                   | -      |
| 1    | EINT1    | Interrupt line 1 enabled | <ul style="list-style-type: none"> <li>0b0: Disable</li> <li>0b1: Enable</li> </ul> | RW     |
| 0    | EINT0    | Interrupt line 0 enabled | <ul style="list-style-type: none"> <li>0b0: Disable</li> <li>0b1: Enable</li> </ul> | RW     |

### 37.4.19 MCAN Global Filter Configuration Register (MCAN\_GFC)

Global settings for message ID filtering. It controls the filtering paths for both standard and extended message IDs. For further details, refer to the sections on "[Standard Message ID Filtering](#)" and "[Extended Message ID Filtering](#)."

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |      |   |      |   |      |   |      |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|------|---|------|---|------|---|------|---|
| 0x80   | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7    | 6 | 5    | 4 | 3    | 2 | 1    | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |      |   |      |   |      |   |      |   |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   | ANFS |   | ANFE |   | RRFS |   | RRFE |   |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   | RWA  |   | RWA  |   | RWA  |   | RWA  |   |

| Bit  | Name     | Function                               | Description                                                                                                                                                                                                                                                                                                                                                                                | Access |
|------|----------|----------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:6 | Reserved | Reserved                               | -                                                                                                                                                                                                                                                                                                                                                                                          | -      |
| 5:4  | ANFS     | Accepting non-matching standard frames | Defines how to handle 11-bit standard ID messages that do not match any element in the filter list: <ul style="list-style-type: none"> <li>0b00: Accept non-matching standard frames in Receive FIFO0</li> <li>0b01: Accept non-matching standard frames in Receive FIFO1</li> <li>0b10: Reject non-matching standard frames</li> <li>0b11: Reject non-matching standard frames</li> </ul> | RWA    |
| 3:2  | ANFE     | Accepting non-matching extended frames | Defines how to handle 29-bit extended ID messages that do not match any element in the filter list: <ul style="list-style-type: none"> <li>0b00: Accept non-matching extended frames in Receive FIFO0</li> <li>0b01: Accept non-matching extended frames in Receive FIFO1</li> <li>0b10: Reject non-matching extended frames</li> <li>0b11: Reject non-matching extended frames</li> </ul> | RWA    |
| 1    | RRFS     | Reject Remote Standard Frames          | <ul style="list-style-type: none"> <li>0b0: Filter remote frames with 11-bit standard ID</li> <li>0b1: Reject all remote frames with 11-bit standard ID</li> </ul>                                                                                                                                                                                                                         | RWA    |
| 0    | RRFE     | Reject remote extension frames         | <ul style="list-style-type: none"> <li>0b0: Filter remote frames with 29-bit extension ID</li> <li>0b1: Reject all remote frames with 29-bit extension ID</li> </ul>                                                                                                                                                                                                                       | RWA    |

### 37.4.20 MCAN Standard ID Filter Configuration Register (MCAN\_SIDFC)

Settings for 11-bit standard ID filter. Standard ID filter configuration, control the filtering path of standard messages, see "[Standard Message ID Filtering](#)".

| Offset | Bit Position |    |    |    |    |    |    |    |     |    |    |    |    |    |    |    |             |    |    |    |    |    |   |   |          |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|-----|----|----|----|----|----|----|----|-------------|----|----|----|----|----|---|---|----------|---|---|---|---|---|---|---|
| 0x84   | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23  | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15          | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7        | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |     |    |    |    |    |    |    |    |             |    |    |    |    |    |   |   |          |   |   |   |   |   |   |   |
| Name   | Reserved     |    |    |    |    |    |    |    | LSS |    |    |    |    |    |    |    | FLSSA[15:2] |    |    |    |    |    |   |   | Reserved |   |   |   |   |   |   |   |
| Access |              |    |    |    |    |    |    |    | RWA |    |    |    |    |    |    |    | RWA         |    |    |    |    |    |   |   |          |   |   |   |   |   |   |   |

| Bit   | Name        | Function                              | Description                                                                                                                                                                                               | Access |
|-------|-------------|---------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:24 | Reserved    | Reserved                              | -                                                                                                                                                                                                         | -      |
| 23:16 | LSS         | Standard ID filter list size          | <ul style="list-style-type: none"> <li>0: No standard ID filter</li> <li>1-128: Number of standard ID filter elements</li> <li>&gt;128: Setting values greater than 128 are interpreted as 128</li> </ul> | RWA    |
| 15:2  | FLSSA[15:2] | Standard ID filter list start address | Set the start address of the standard ID filter list (32-bit word address, see " <a href="#">Message RAM Configuration</a> ")                                                                             | RWA    |
| 1:0   | Reserved    | Reserved                              | -                                                                                                                                                                                                         | -      |

### 37.4.21 MCAN Extended ID Filter Configuration Register (MCAN\_XIDFC)

Settings for 29-bit extended ID filter. Extension ID filter configuration, control the filtering path of extension messages, see "[Extended Message ID Filtering](#)".

| Offset | Bit Position |    |    |    |    |    |    |    |     |    |    |    |    |    |    |    |             |    |    |    |    |    |   |   |          |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|-----|----|----|----|----|----|----|----|-------------|----|----|----|----|----|---|---|----------|---|---|---|---|---|---|---|
| 0x88   | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23  | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15          | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7        | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |     |    |    |    |    |    |    |    |             |    |    |    |    |    |   |   |          |   |   |   |   |   |   |   |
| Name   | Reserved     |    |    |    |    |    |    |    | LES |    |    |    |    |    |    |    | FLESA[15:2] |    |    |    |    |    |   |   | Reserved |   |   |   |   |   |   |   |
| Access |              |    |    |    |    |    |    |    | RWA |    |    |    |    |    |    |    | RWA         |    |    |    |    |    |   |   |          |   |   |   |   |   |   |   |

| Bit   | Name        | Function                              | Description                                                                                                                                                                                           | Access |
|-------|-------------|---------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:23 | Reserved    | Reserved                              | -                                                                                                                                                                                                     | -      |
| 22:16 | LES         | Standard ID filter list size          | <ul style="list-style-type: none"> <li>0: No extended ID filter</li> <li>1-64: Number of extended ID filter elements</li> <li>&gt;64: Setting values greater than 64 are interpreted as 64</li> </ul> | RWA    |
| 15:2  | FLESA[15:2] | Extended ID filter list start address | Set the start address of the extended ID filter list (32-bit word address, see " <a href="#">Message RAM Configuration</a> ")                                                                         | RWA    |
| 1:0   | Reserved    | Reserved                              | -                                                                                                                                                                                                     | -      |



## 37.4.22 MCAN Extended ID and Mask Register (MCAN\_XIDAM)

| Offset | Bit Position |    |      |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |  |  |  |  |  |
|--------|--------------|----|------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|--|--|--|--|--|--|
| 0x90   | 31           | 30 | 29   | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |  |  |  |  |  |  |
| Reset  | 0x1FFFFFFF   |    |      |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |  |  |  |  |  |
| Name   | Reserved     |    | EIDM |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |  |  |  |  |  |
| Access |              |    | RWA  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |  |  |  |  |  |

| Bit   | Name     | Function         | Description                                                                                                                                                                                           | Access |
|-------|----------|------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:29 | Reserved | Reserved         | -                                                                                                                                                                                                     | -      |
| 28:0  | EIDM     | Extended ID mask | For extended frame reception filtering, EIDM is ANDed with the ID of the received frame. Used to mask the 29-bit ID in SAE J1939. The reset value of all bits is set to 1 and the mask has no effect. | RWA    |

### 37.4.23 MCAN High Priority Message Status Register (MCAN\_HPMS)

A message ID filter element configured to generate a priority event updates this register each time it is matched. This can be used to monitor the status of received high priority messages and provide quick access to these messages.

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |      |      |    |    |    |    |     |   |   |   |    |      |   |   |   |    |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|------|------|----|----|----|----|-----|---|---|---|----|------|---|---|---|----|
| 0x94   | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15   | 14   | 13 | 12 | 11 | 10 | 9   | 8 | 7 | 6 | 5  | 4    | 3 | 2 | 1 | 0  |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |      |      |    |    |    |    |     |   |   |   |    |      |   |   |   |    |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |      | FIDX |    |    |    |    | MSI |   |   |   |    | BIDX |   |   |   |    |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | FLST |      |    |    |    | RO |     |   |   |   | RO |      |   |   |   | RO |

| Bit   | Name     | Function                 | Description                                                                                                                                                                              | Access |
|-------|----------|--------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:16 | Reserved | Reserved                 | -                                                                                                                                                                                        | -      |
| 15    | FLST     | Filter list              | Indicates the matching filter list: <ul style="list-style-type: none"> <li>0b0: Standard filter list</li> <li>0b1: Extended filter list</li> </ul>                                       | RO     |
| 14:8  | FIDX     | Filter index             | The index of the matching filter element. The range is 0 to MCAN_SIDFC.LSS minus 1 or MCAN_XIDFC.LSE minus 1                                                                             | RO     |
| 7:6   | MSI      | Message memory indicator | <ul style="list-style-type: none"> <li>0b00: FIFO not selected</li> <li>0b01: FIFO message lost</li> <li>0b10: Message stored in FIFO0</li> <li>0b11: Message stored in FIFO1</li> </ul> | RO     |
| 5:0   | BIDX     | Buffer index             | The index of the receive FIFO element storing the message. Only active when MSI[1]=1                                                                                                     | RO     |

## 37.4.24 MCAN New Data 1 Register (MCAN\_NDAT1)

| Offset | Bit Position |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |     |     |     |     |     |     |     |     |     |     |
|--------|--------------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| 0x98   | 31           | 30   | 29   | 28   | 27   | 26   | 25   | 24   | 23   | 22   | 21   | 20   | 19   | 18   | 17   | 16   | 15   | 14   | 13   | 12   | 11   | 10   | 9   | 8   | 7   | 6   | 5   | 4   | 3   | 2   | 1   | 0   |
| Reset  | 0x00000000   |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |     |     |     |     |     |     |     |     |     |     |
| Name   | ND31         | ND30 | ND29 | ND28 | ND27 | ND26 | ND25 | ND24 | ND23 | ND22 | ND21 | ND20 | ND19 | ND18 | ND17 | ND16 | ND15 | ND14 | ND13 | ND12 | ND11 | ND10 | ND9 | ND8 | ND7 | ND6 | ND5 | ND4 | ND3 | ND2 | ND1 | ND0 |
| Access | RW           | RW   | RW   | RW   | RW   | RW   | RW   | RW   | RW   | RW   | RW   | RW   | RW   | RW   | RW   | RW   | RW   | RW   | RW   | RW   | RW   | RW   | RW  | RW  | RW  | RW  | RW  | RW  | RW  | RW  | RW  | RW  |

| Bit  | Name         | Function | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | Access |
|------|--------------|----------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:0 | NDn (n=31-0) | New data | <p>Corresponding to the new data flag of the receive buffer 31-0:</p> <ul style="list-style-type: none"> <li>● 0b0: The receive buffer is not updated</li> <li>● 0b1: The receive buffer is updated by a new message</li> </ul> <p>When the receive frame updates the receive buffer, the corresponding flag bit will be set to 1. The flag bit remains set to 1 until the host clears it. The flag bit is cleared by writing 1 to the corresponding bit, and write 0 has no effect. A hard reset will clear the register.</p> <div>  <b>Note</b><br/> Each bit corresponds to a buffer. </div> | RW     |

## 37.4.25 MCAN New Data 2 Register (MCAN\_NDAT2)

| Offset | Bit Position |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
|--------|--------------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| 0x9C   | 31           | 30   | 29   | 28   | 27   | 26   | 25   | 24   | 23   | 22   | 21   | 20   | 19   | 18   | 17   | 16   | 15   | 14   | 13   | 12   | 11   | 10   | 9    | 8    | 7    | 6    | 5    | 4    | 3    | 2    | 1    | 0    |
| Reset  | 0x00000000   |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
| Name   | ND63         | ND62 | ND61 | ND60 | ND59 | ND58 | ND57 | ND56 | ND55 | ND54 | ND53 | ND52 | ND51 | ND50 | ND49 | ND48 | ND47 | ND46 | ND45 | ND44 | ND43 | ND42 | ND41 | ND40 | ND39 | ND38 | ND37 | ND36 | ND35 | ND34 | ND33 | ND32 |
| Access | RW           | RW   | RW   | RW   | RW   | RW   | RW   | RW   | RW   | RW   | RW   | RW   | RW   | RW   | RW   | RW   | RW   | RW   | RW   | RW   | RW   | RW   | RW   | RW   | RW   | RW   | RW   | RW   | RW   | RW   | RW   | RW   |

| Bit  | Name          | Function | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | Access |
|------|---------------|----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:0 | NDn (n=63-32) | New data | <p>Corresponding to the new data flag of the receive buffer 63-32:</p> <ul style="list-style-type: none"> <li>● 0b0: The receive buffer is not updated</li> <li>● 0b1: The receive buffer is updated by a new message</li> </ul> <p>When the receive frame updates the receive buffer, the corresponding flag bit will be set to 1. The flag bit remains set to 1 until the host clears it. The flag bit is cleared by writing 1 to the corresponding bit, and write 0 has no effect. A hard reset will clear the register.</p> <div>  <b>Note</b><br/> Each bit corresponds to a buffer. </div> | RW     |

## 37.4.26 MCAN Receive FIFO0 Configuration Register (MCAN\_RXF0C)

| Offset | Bit Position |      |    |    |    |    |    |    |          |     |    |    |    |    |    |    |            |    |    |    |    |    |   |          |   |   |   |   |   |   |   |   |
|--------|--------------|------|----|----|----|----|----|----|----------|-----|----|----|----|----|----|----|------------|----|----|----|----|----|---|----------|---|---|---|---|---|---|---|---|
| 0xA0   | 31           | 30   | 29 | 28 | 27 | 26 | 25 | 24 | 23       | 22  | 21 | 20 | 19 | 18 | 17 | 16 | 15         | 14 | 13 | 12 | 11 | 10 | 9 | 8        | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |      |    |    |    |    |    |    |          |     |    |    |    |    |    |    |            |    |    |    |    |    |   |          |   |   |   |   |   |   |   |   |
| Name   | F0OM         | F0WM |    |    |    |    |    |    | Reserved | F0S |    |    |    |    |    |    | F0SA[15:2] |    |    |    |    |    |   | Reserved |   |   |   |   |   |   |   |   |
| Access | RWA          | RWA  |    |    |    |    |    |    |          | RWA |    |    |    |    |    |    | RWA        |    |    |    |    |    |   |          |   |   |   |   |   |   |   |   |

| Bit   | Name       | Function                    | Description                                                                                                                                                                                                                                                       | Access |
|-------|------------|-----------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31    | F0OM       | Receive FIFO0 Working Mode  | <ul style="list-style-type: none"> <li>0b0: Receive FIFO0 is in blocking mode</li> <li>0b1: Receive FIFO0 is in overwriting mode</li> </ul>                                                                                                                       | RWA    |
| 30:24 | F0WM       | Receive FIFO0 Water Level   | <ul style="list-style-type: none"> <li>0: Disable receive FIFO0 water level interrupt</li> <li>1-64: Level of receive FIFO0 waterline interrupt (MCAN_IR.RF0W)</li> <li>&gt;64: Disable receive FIFO0 water level interrupt</li> </ul>                            | RWA    |
| 23    | Reserved   | Reserved                    | -                                                                                                                                                                                                                                                                 | -      |
| 22:16 | F0S        | Receive FIFO0 Size          | <ul style="list-style-type: none"> <li>0: No receive FIFO0</li> <li>1-64: Number of receive FIFO0 elements</li> <li>&gt;64: Setting values greater than 64 are interpreted as 64</li> </ul> <p>The receive FIFO0 elements are indexed from 0 to F0 S minus 1.</p> | RWA    |
| 15:2  | F0SA[15:2] | Receive FIFO0 Start Address | The starting address of the receive FIFO0 in the message RAM (32-bit word address, see " <a href="#">Message RAM Configuration</a> ")                                                                                                                             | RWA    |
| 1:0   | Reserved   | Reserved                    | -                                                                                                                                                                                                                                                                 | -      |

## 37.4.27 MCAN Receive FIFO0 Status Register (MCAN\_RXF0S)

| Offset | Bit Position |    |    |    |    |    |    |      |    |    |     |          |    |    |    |    |    |      |          |    |    |    |   |   |      |          |   |    |   |   |   |   |  |      |
|--------|--------------|----|----|----|----|----|----|------|----|----|-----|----------|----|----|----|----|----|------|----------|----|----|----|---|---|------|----------|---|----|---|---|---|---|--|------|
| 0xA4   | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24   | 23 | 22 | 21  | 20       | 19 | 18 | 17 | 16 | 15 | 14   | 13       | 12 | 11 | 10 | 9 | 8 | 7    | 6        | 5 | 4  | 3 | 2 | 1 | 0 |  |      |
| Reset  | 0x00000000   |    |    |    |    |    |    |      |    |    |     |          |    |    |    |    |    |      |          |    |    |    |   |   |      |          |   |    |   |   |   |   |  |      |
| Name   | Reserved     |    |    |    |    |    | RO | RF0L |    | RO | F0F | Reserved |    | RO |    |    |    | F0PI | Reserved |    | RO |    |   |   | F0GI | Reserved |   | RO |   |   |   |   |  | F0FL |
| Access |              |    |    |    |    |    | RO |      |    | RO |     |          |    | RO |    |    |    |      |          |    | RO |    |   |   |      |          |   | RO |   |   |   |   |  |      |

| Bit   | Name     | Function                                                     | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | Access |
|-------|----------|--------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:26 | Reserved | Reserved                                                     | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | -      |
| 25    | RF0L     | Receive FIFO0 message lost                                   | <p>This bit is a copy of the interrupt flag MCAN_IR.RF0L. When MCAN_IR.RF0L is reset, this bit is also reset:</p> <ul style="list-style-type: none"> <li>0b0: No message lost in the receive FIFO0</li> <li>0b1: Message lost in the receive FIFO0; this bit is also set when writing to a receive FIFO0 with a space size of 0</li> </ul> <div>  <b>Note</b><br/>           When MCAN_RXF0C.F0OM=1, the oldest message is overwritten and this flag is not set.         </div> | RO     |
| 24    | F0F      | 0b0: Receive FIFO0 is not full<br>0b1: Receive FIFO0 is full | <ul style="list-style-type: none"> <li>0b0: Receive FIFO0 is not full</li> <li>0b1: Receive FIFO0 is full</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                              | RO     |
| 23:22 | Reserved | Reserved                                                     | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | -      |
| 21:16 | F0PI     | Receive FIFO0 puts index                                     | Receive FIFO0 write index pointer, the active range is 0-63.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | RO     |
| 15:14 | Reserved | Reserved                                                     | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | -      |
| 13:8  | F0GI     | Receive FIFO0 get index                                      | Receive FIFO0 read index pointer, the active range is 0-63.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | RO     |
| 7     | Reserved | Reserved                                                     | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | -      |
| 6:0   | F0FL     | Receive FIFO0 fill level                                     | The number of elements stored in the receive FIFO0. The active range is 0-64.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | RO     |

## 37.4.28 MCAN Receive FIFO0 Acknowledge Register (MCAN\_RXF0A)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |      |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|------|---|---|---|---|---|---|---|
| 0xA8   | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7    | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |      |   |   |   |   |   |   |   |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   | F0AI |   |   |   |   |   |   |   |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   | RW   |   |   |   |   |   |   |   |

| Bit  | Name     | Function                        | Description                                                                                                                                                                                                                                                                                            | Access |
|------|----------|---------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:6 | Reserved | Reserved                        | -                                                                                                                                                                                                                                                                                                      | -      |
| 5:0  | F0AI     | Receive FIFO0 acknowledge index | After the host reads a message or sequence of messages from the receive FIFO0, it must write the buffer index of the last element read from the receive FIFO0 to F0AI. This operation sets the receive FIFO0 get index MCAN_RXF0S.F0GI to F0AI plus 1 and updates the FIFO0 fill level MCAN_RXF0S.F0FL | RW     |

## 37.4.29 MCAN Receive Buffer Configuration Register (MCAN\_RXBC)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |            |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |          |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|------------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|----------|
| 0xAC   | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15         | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |          |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |            |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |          |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RBSA[15:2] |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   | Reserved |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RWA        |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |          |

| Bit   | Name       | Function                                   | Description                                                                                                                             | Access |
|-------|------------|--------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:16 | Reserved   | Reserved                                   | -                                                                                                                                       | -      |
| 15:2  | RBSA[15:2] | The starting address of the receive buffer | Configure the starting address of the receive buffer in the message RAM (32-bit address), also used to reference debug messages A, B, C | RWA    |
| 1:0   | Reserved   | Reserved                                   | -                                                                                                                                       | -      |

## 37.4.30 MCAN Receive FIFO1 Configuration Register (MCAN\_RXF1C)

| Offset | Bit Position |      |    |    |    |    |    |    |          |     |    |    |    |    |    |    |            |    |    |    |    |    |   |   |   |   |          |   |   |   |   |   |
|--------|--------------|------|----|----|----|----|----|----|----------|-----|----|----|----|----|----|----|------------|----|----|----|----|----|---|---|---|---|----------|---|---|---|---|---|
| 0xB0   | 31           | 30   | 29 | 28 | 27 | 26 | 25 | 24 | 23       | 22  | 21 | 20 | 19 | 18 | 17 | 16 | 15         | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5        | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |      |    |    |    |    |    |    |          |     |    |    |    |    |    |    |            |    |    |    |    |    |   |   |   |   |          |   |   |   |   |   |
| Name   | F1OM         | F1WM |    |    |    |    |    |    | Reserved | F1S |    |    |    |    |    |    | F1SA[15:2] |    |    |    |    |    |   |   |   |   | Reserved |   |   |   |   |   |
|        |              |      |    |    |    |    |    |    |          |     |    |    |    |    |    |    |            |    |    |    |    |    |   |   |   |   |          |   |   |   |   |   |
| Access | RWA          | RWA  |    |    |    |    |    |    |          | RWA |    |    |    |    |    |    | RWA        |    |    |    |    |    |   |   |   |   |          |   |   |   |   |   |

| Bit   | Name       | Function                    | Description                                                                                                                                                                                                                                                      | Access |
|-------|------------|-----------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31    | F1OM       | Receive FIFO1 working mode  | <ul style="list-style-type: none"> <li>0b0: Receive FIFO1 is in blocking mode</li> <li>0b1: Receive FIFO1 is in overwriting mode</li> </ul>                                                                                                                      | RWA    |
| 30:24 | F1WM       | Receive FIFO1 water level   | <ul style="list-style-type: none"> <li>0: Disable receive FIFO1 water level interrupt</li> <li>1-64: Level of receive FIFO1 waterline interrupt (MCAN_IR.RF1W)</li> <li>&gt;64: Disable receive FIFO1 water level interrupt</li> </ul>                           | RWA    |
| 23    | Reserved   | Reserved                    | -                                                                                                                                                                                                                                                                | -      |
| 22:16 | F1S        | Receive FIFO1 size          | <ul style="list-style-type: none"> <li>0: No receive FIFO1</li> <li>1-64: Number of receive FIFO1 elements</li> <li>&gt;64: Setting values greater than 64 are interpreted as 64</li> </ul> <p>The receive FIFO1 elements are indexed from 0 to F1S minus 1.</p> | RWA    |
| 15:2  | F1SA[15:2] | Receive FIFO1 start address | The starting address of the receive FIFO1 in the message RAM (32-bit word address, see " <a href="#">Message RAM Configuration</a> ")                                                                                                                            | RWA    |
| 1:0   | Reserved   | Reserved                    | -                                                                                                                                                                                                                                                                | -      |



## 37.4.31 MCAN Receive FIFO1 Status Register (MCAN\_RXF1S)

| Offset | Bit Position |          |    |    |    |    |      |    |     |    |          |      |    |    |    |    |          |      |    |    |    |    |          |      |   |    |   |   |   |   |   |   |  |
|--------|--------------|----------|----|----|----|----|------|----|-----|----|----------|------|----|----|----|----|----------|------|----|----|----|----|----------|------|---|----|---|---|---|---|---|---|--|
| 0xB4   | 31           | 30       | 29 | 28 | 27 | 26 | 25   | 24 | 23  | 22 | 21       | 20   | 19 | 18 | 17 | 16 | 15       | 14   | 13 | 12 | 11 | 10 | 9        | 8    | 7 | 6  | 5 | 4 | 3 | 2 | 1 | 0 |  |
| Reset  | 0x00000000   |          |    |    |    |    |      |    |     |    |          |      |    |    |    |    |          |      |    |    |    |    |          |      |   |    |   |   |   |   |   |   |  |
| Name   | DMS          | Reserved |    |    |    |    | RF1L |    | F1F |    | Reserved | F1PI |    |    |    |    | Reserved | F1GI |    |    |    |    | Reserved | F1FL |   |    |   |   |   |   |   |   |  |
| Access |              | RO       |    |    |    |    | RO   |    | RO  |    |          | RO   |    |    |    |    |          |      | RO |    |    |    |          |      |   | RO |   |   |   |   |   |   |  |

| Bit   | Name     | Function                   | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | Access |
|-------|----------|----------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:30 | DMS      | Debug message status       | <ul style="list-style-type: none"> <li>0b00: Idle state, waiting to receive debug message</li> <li>0b01: Received debug message A</li> <li>0b10: Received debug messages A and B</li> <li>0b11: Received debug messages A, B, and C</li> </ul>                                                                                                                                                                                                                                                                                                                     | RO     |
| 29:26 | Reserved | Reserved                   | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | -      |
| 25    | RF1L     | Receive FIFO1 message lost | <p>This bit is a copy of the interrupt flag MCAN_IR.RF1L. When MCAN_IR.RF1L is reset, this bit is also reset:</p> <ul style="list-style-type: none"> <li>0b0: No message lost in the receive FIFO1</li> <li>0b1: Message lost in the receive FIFO1; this bit is also set when writing to a receive FIFO1 with a space size of 0</li> </ul> <div>  <b>Note</b><br/>           When MCAN_RXF1C.F1OM=1, the oldest message is overwritten and this flag is not set.         </div> | RO     |
| 24    | F1F      | 0b0: Receive FIFO1 is full | <ul style="list-style-type: none"> <li>0b0: Receive FIFO1 is not full</li> <li>0b1: Receive FIFO1 is full</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                               | RO     |
| 23:22 | Reserved | Reserved                   | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | -      |
| 21:16 | F1PI     | Receive FIFO1 puts index   | Receive FIFO1 write index pointer, the active range is 0-63                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | RO     |
| 15:14 | Reserved | Reserved                   | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | -      |
| 13:8  | F1GI     | Receive FIFO1 get index    | Receive FIFO1 read index pointer, the active range is 0-63                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | RO     |
| 7     | Reserved | Reserved                   | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | -      |
| 6:0   | F1FL     | Receive FIFO1 fill level   | The number of elements stored in the receive FIFO1. The active range is 0-64                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | RO     |

## 37.4.32 MCAN Receive FIFO1 Acknowledge Register (MCAN\_RXF1A)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |      |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|------|---|---|---|---|---|---|---|
| 0xB8   | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7    | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |      |   |   |   |   |   |   |   |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   | F1AI |   |   |   |   |   |   |   |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   | RW   |   |   |   |   |   |   |   |

| Bit  | Name     | Function                        | Description                                                                                                                                                                                                                                                                                            | Access |
|------|----------|---------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:6 | Reserved | Reserved                        | -                                                                                                                                                                                                                                                                                                      | -      |
| 5:0  | F1AI     | Receive FIFO1 acknowledge index | After the host reads a message or sequence of messages from the receive FIFO1, it must write the buffer index of the last element read from the receive FIFO1 to F1AI. This operation sets the receive FIFO1 get index MCAN_RXF1S.F1GI to F1AI plus 1 and updates the FIFO1 fill level MCAN_RXF1S.F1FL | RW     |

### 37.4.33 MCAN Receive Buffer and FIFO Element Size Configuration Registers (MCAN\_RXESC)

Configure the receive buffer and the number of data bytes for the receive FIFO elements. Data fields larger than 8 bytes are only applicable in CAN FD mode.

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |      |   |          |      |   |          |      |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|------|---|----------|------|---|----------|------|---|---|---|
| 0xBC   | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9    | 8 | 7        | 6    | 5 | 4        | 3    | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |      |   |          |      |   |          |      |   |   |   |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RBDs |   | Reserved | F1Ds |   | Reserved | F0Ds |   |   |   |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RWA  |   |          | RWA  |   |          | RWA  |   |   |   |

| Bit   | Name     | Function                       | Description                                                                                                                                                                                                                                                     | Access |
|-------|----------|--------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:11 | Reserved | Reserved                       | -                                                                                                                                                                                                                                                               | -      |
| 10:8  | RBDS     | Receive buffer data field size | <ul style="list-style-type: none"> <li>● 0b000: 8 bytes</li> <li>● 0b001: 12 bytes</li> <li>● 0b010: 16 bytes</li> <li>● 0b011: 20 bytes</li> <li>● 0b100: 24 bytes</li> <li>● 0b101: 32 bytes</li> <li>● 0b110: 48 bytes</li> <li>● 0b111: 64 bytes</li> </ul> | RWA    |
| 7     | Reserved | Reserved                       | -                                                                                                                                                                                                                                                               | -      |
| 6:4   | F1DS     | Receive FIFO1 data field size  | <ul style="list-style-type: none"> <li>● 0b000: 8 bytes</li> <li>● 0b001: 12 bytes</li> <li>● 0b010: 16 bytes</li> <li>● 0b011: 20 bytes</li> <li>● 0b100: 24 bytes</li> <li>● 0b101: 32 bytes</li> <li>● 0b110: 48 bytes</li> <li>● 0b111: 64 bytes</li> </ul> | RWA    |
| 3     | Reserved | Reserved                       | -                                                                                                                                                                                                                                                               | -      |
| 2:0   | F0DS     | Receive FIFO0 data field size  | <ul style="list-style-type: none"> <li>● 0b000: 8 bytes</li> <li>● 0b001: 12 bytes</li> <li>● 0b010: 16 bytes</li> <li>● 0b011: 20 bytes</li> <li>● 0b100: 24 bytes</li> <li>● 0b101: 32 bytes</li> <li>● 0b110: 48 bytes</li> <li>● 0b111: 64 bytes</li> </ul> | RWA    |

### 37.4.34 MCAN Transmit Buffer Configuration Register (MCAN\_TXBC)

Write access to this register is possible only when both the CCE bit and the INIT bit of the MCAN\_CCCR register are set to 1.

| Offset | Bit Position |      |      |    |    |    |    |          |      |    |    |    |    |            |    |    |    |    |    |    |    |    |   |   |   |   |   |   |          |   |   |   |
|--------|--------------|------|------|----|----|----|----|----------|------|----|----|----|----|------------|----|----|----|----|----|----|----|----|---|---|---|---|---|---|----------|---|---|---|
| 0xC0   | 31           | 30   | 29   | 28 | 27 | 26 | 25 | 24       | 23   | 22 | 21 | 20 | 19 | 18         | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3        | 2 | 1 | 0 |
| Reset  | 0x00000000   |      |      |    |    |    |    |          |      |    |    |    |    |            |    |    |    |    |    |    |    |    |   |   |   |   |   |   |          |   |   |   |
| Name   | Reserved     | TFQM | TFQS |    |    |    |    | Reserved | NDTB |    |    |    |    | TBSA[15:2] |    |    |    |    |    |    |    |    |   |   |   |   |   |   | Reserved |   |   |   |
| Access |              | RWA  | RWA  |    |    |    |    |          | RWA  |    |    |    |    | RWA        |    |    |    |    |    |    |    |    |   |   |   |   |   |   |          |   |   |   |

| Bit   | Name       | Function                       | Description                                                                                                                                                                                           | Access |
|-------|------------|--------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31    | Reserved   | Reserved                       | -                                                                                                                                                                                                     | -      |
| 30    | TFQM       | Transmit FIFO/queue mode       | <ul style="list-style-type: none"> <li>0b0: Transmit FIFO mode</li> <li>0b1: Transmit queue mode</li> </ul>                                                                                           | RWA    |
| 29:24 | TFQS       | Transmit FIFO/queue size       | <ul style="list-style-type: none"> <li>0: No transmit FIFO or queue</li> <li>1-32: Transmit FIFO queue buffer size</li> <li>&gt;32: Setting values greater than 32 are interpreted as 32</li> </ul>   | RWA    |
| 23:22 | Reserved   | Reserved                       | -                                                                                                                                                                                                     | -      |
| 21:16 | NDTB       | Dedicated transmit buffer size | <ul style="list-style-type: none"> <li>0: No dedicated transmit buffer</li> <li>1-32: Dedicated transmit buffer size</li> <li>&gt;32: Setting values greater than 32 are interpreted as 32</li> </ul> | RWA    |
| 15:2  | TBSA[15:2] | Tx Buffer start address        | The starting address of the transmit buffer portion of the Message RAM (32-bit word address, see <a href="#">"Message RAM Configuration"</a> )                                                        | RWA    |
| 1:0   | Reserved   | Reserved                       | -                                                                                                                                                                                                     | -      |



#### Note

The sum of TFQS and NDTB shall not be greater than 32. MCAN does not monitor this configuration error. The transmit buffer portion of the message RAM starts with a dedicated transmit buffer.

### 37.4.35 MCAN Transmit FIFO/Queue Status Register (MCAN\_TXFQS)

The status of the transmit FIFO or queue is related to the pending transmit request in register MCAN\_TXBRP. Therefore, the effect of adding or canceling a request may be delayed due to a transmit scan that is running (TXBRP has not yet been updated).

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |      |       |    |    |    |          |    |    |    |      |    |    |   |          |   |      |   |   |   |   |   |   |  |
|--------|--------------|----|----|----|----|----|----|----|----|----|------|-------|----|----|----|----------|----|----|----|------|----|----|---|----------|---|------|---|---|---|---|---|---|--|
| 0xC4   | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21   | 20    | 19 | 18 | 17 | 16       | 15 | 14 | 13 | 12   | 11 | 10 | 9 | 8        | 7 | 6    | 5 | 4 | 3 | 2 | 1 | 0 |  |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |      |       |    |    |    |          |    |    |    |      |    |    |   |          |   |      |   |   |   |   |   |   |  |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    | TFQF | TFQPI |    |    |    | Reserved |    |    |    | TFGI |    |    |   | Reserved |   | TFFL |   |   |   |   |   |   |  |
| Access |              |    |    |    |    |    |    |    |    |    | RO   | RO    |    |    |    |          |    |    |    | RO   |    |    |   |          |   | RO   |   |   |   |   |   |   |  |

| Bit   | Name     | Function                      | Description                                                                                                                                                                                                  | Access |
|-------|----------|-------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:22 | Reserved | Reserved                      | -                                                                                                                                                                                                            | -      |
| 21    | TFQF     | Transmit FIFO/queue full      | <ul style="list-style-type: none"> <li>0b0: Transmit FIFO or queue not full</li> <li>0b1: Transmit FIFO or queue full</li> </ul>                                                                             | RO     |
| 20:16 | TFQPI    | Transmit FIFO/queue put index | The put index pointer of the transmit FIFO or queue, the active range is 0-31                                                                                                                                | RO     |
| 15:13 | Reserved | Reserved                      | -                                                                                                                                                                                                            | -      |
| 12:8  | TFGI     | Transmit FIFO get index       | The read index pointer of the transmit FIFO, the active range is 0-31. If it has been configured as the transmit queue mode (MCAN_TXBC.TFQM=1), the read value of this bit is 0                              | RO     |
| 7:6   | Reserved | Reserved                      | -                                                                                                                                                                                                            | -      |
| 5:0   | TFFL     | Transmit FIFO idle level      | The number of consecutive idle transmit FIFO elements starting from TFGI, the active range is 0-32. If it has been configured as the transmit queue mode (MCAN_TXBC.TFQM=1), the read value of this bit is 0 | RO     |



#### Note

In the case where a dedicated transmit buffer is configured in combination with a transmit FIFO or a transmit queue, the put index and get index refer to the transmit buffer number starting from the first dedicated transmit buffer.

For example: For a combined configuration with 12 dedicated transmit buffers and 20 transmit FIFO buffers, put index 15 to point to the 4th buffer of the transmit FIFO.

### 37.4.36 MCAN Transmit Buffer Element Size Configuration Register (MCAN\_TXESC)

Configure the number of data bytes for a transmit buffer element. Data fields larger than 8 bytes are only applicable in CAN FD mode.

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |      |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|------|
| 0xC8   | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0    |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |      |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   | TBDS |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   | RW   |

| Bit  | Name     | Function                        | Description                                                                                                                                                                                                                                                     | Access |
|------|----------|---------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:3 | Reserved | Reserved                        | -                                                                                                                                                                                                                                                               | -      |
| 2:0  | TBDS     | Transmit buffer data field size | <ul style="list-style-type: none"> <li>● 0b000: 8 bytes</li> <li>● 0b001: 12 bytes</li> <li>● 0b010: 16 bytes</li> <li>● 0b011: 20 bytes</li> <li>● 0b100: 24 bytes</li> <li>● 0b101: 32 bytes</li> <li>● 0b110: 48 bytes</li> <li>● 0b111: 64 bytes</li> </ul> | RW     |



#### Note

If the data length of the transmit buffer element (configured by DLC) is greater than the data length configured by TBDS, the undefined data bytes in the transmit buffer will be sent as 0xCC (filling bytes).

## 37.4.37 MCAN Transmit Buffer Request Pending Register (MCAN\_TXBRP)

| Offset | Bit Position |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |      |      |      |      |      |      |      |      |      |      |    |
|--------|--------------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|------|------|------|------|------|------|------|------|------|------|----|
| 0xCC   | 31           | 30    | 29    | 28    | 27    | 26    | 25    | 24    | 23    | 22    | 21    | 20    | 19    | 18    | 17    | 16    | 15    | 14    | 13    | 12    | 11    | 10    | 9    | 8    | 7    | 6    | 5    | 4    | 3    | 2    | 1    | 0    |    |
| Reset  | 0x00000000   |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |      |      |      |      |      |      |      |      |      |      |    |
| Name   | TRP31        | TRP30 | TRP29 | TRP28 | TRP27 | TRP26 | TRP25 | TRP24 | TRP23 | TRP22 | TRP21 | TRP20 | TRP19 | TRP18 | TRP17 | TRP16 | TRP15 | TRP14 | TRP13 | TRP12 | TRP11 | TRP10 | TRP9 | TRP8 | TRP7 | TRP6 | TRP5 | TRP4 | TRP3 | TRP2 | TRP1 | TRP0 |    |
| Access | RO           | RO    | RO    | RO    | RO    | RO    | RO    | RO    | RO    | RO    | RO    | RO    | RO    | RO    | RO    | RO    | RO    | RO    | RO    | RO    | RO    | RO    | RO   | RO   | RO   | RO   | RO   | RO   | RO   | RO   | RO   | RO   | RO |

| Bit  | Name             | Functional Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | Access |
|------|------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:0 | TRPn<br>(n=31-0) | <p>Transmit request pending flag</p> <p>Corresponding to the transmit request pending bits of transmit buffers 31-0:</p> <ul style="list-style-type: none"> <li>0b0: No transmit request pending</li> <li>0b1: Transmit request pending</li> </ul> <p>These bits are set by register MCAN_TXBAR; reset after the requested transmit is completed or the requested transmit is canceled by register MCAN_TXBCR.</p> <p>The TXBRP bit corresponding to the transmit buffer configured by MCAN_TXBC is set. When the TXBRP bit is set, a transmit scan (see "<a href="#">Transmission Processing</a>") is started to check the highest priority pending transmit request (i.e. the transmit buffer with the smallest message ID).</p> <p>Canceling a transmit request resets the corresponding TXBRP bit. If the transmit has already started when the request is canceled, the TXBRP bit will be reset at the end of the transmission regardless of whether the transmit is successful or not. After the TXBRP bit is reset, the corresponding cancel request bit will also be reset immediately.</p> <p>After requesting cancellation, the following situations indicate cancellation completion through register MCAN_TXBCF:</p> <ul style="list-style-type: none"> <li>After successful transmission and corresponding MCAN_TXBTO bit is set</li> <li>Transmission has not started at the time of cancellation</li> <li>Transmission is aborted due to arbitration failure</li> <li>Error occurs during frame transmission</li> </ul> <p>In DAR mode, all transmissions are automatically canceled after failure. For all failed transmissions, the corresponding MCAN_TXBCF bit is set.</p> <div>  <b>Note</b><br/> Each bit corresponds to a buffer.<br/><br/> TXBRP bits set during the transmission scan are ignored during this transmission scan. During this period, if cancellation of these transmissions is requested, the transmission is immediately canceled and the corresponding TXBRP bits are reset. </div> | RO     |

## 37.4.38 MCAN Transmit Buffer Add Request Register (MCAN\_TXBAR)

| Offset | Bit Position |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |     |     |     |     |     |     |     |     |     |     |    |
|--------|--------------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|----|
| 0xD0   | 31           | 30   | 29   | 28   | 27   | 26   | 25   | 24   | 23   | 22   | 21   | 20   | 19   | 18   | 17   | 16   | 15   | 14   | 13   | 12   | 11   | 10   | 9   | 8   | 7   | 6   | 5   | 4   | 3   | 2   | 1   | 0   |    |
| Reset  | 0x00000000   |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |     |     |     |     |     |     |     |     |     |     |    |
| Name   | AR31         | AR30 | AR29 | AR28 | AR27 | AR26 | AR25 | AR24 | AR23 | AR22 | AR21 | AR20 | AR19 | AR18 | AR17 | AR16 | AR15 | AR14 | AR13 | AR12 | AR11 | AR10 | AR9 | AR8 | AR7 | AR6 | AR5 | AR4 | AR3 | AR2 | AR1 | AR0 |    |
| Access | RW           | RW   | RW   | RW   | RW   | RW   | RW   | RW   | RW   | RW   | RW   | RW   | RW   | RW   | RW   | RW   | RW   | RW   | RW   | RW   | RW   | RW   | RW  | RW  | RW  | RW  | RW  | RW  | RW  | RW  | RW  | RW  | RW |

| Bit  | Name            | Function    | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | Access |
|------|-----------------|-------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:0 | ARn<br>(n=31-0) | Add request | <p>Corresponding to the add request bit of the transmit buffer 31-0:</p> <p>Write 1 to set the corresponding add request bit, write 0 has no effect</p> <p>Read:</p> <ul style="list-style-type: none"> <li>● 0b0: No transmit request added</li> <li>● 0b1: Transmit request added</li> </ul> <p>The host can set the transmit request of multiple transmit buffers at a time. Only the TXBAR bits corresponding to the transmit buffers configured by MCAN_TXBC will be set. When the transmit scan is not running, these bits are reset immediately, otherwise they will remain set until the transmit scan is completed.</p> <div>  <b>Note</b><br/> Each bit corresponds to a buffer.<br/><br/> If the transmit buffer to which the add request is sent already has a transmit request pending (the corresponding bit of the MCAN_TXBRP register is set), the add request will be ignored. </div> | RW     |



## 37.4.39 MCAN Transmit Buffer Cancel Request Register (MCAN\_TXBCR)

| Offset | Bit Position |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |     |     |     |     |     |     |     |     |     |     |    |
|--------|--------------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|----|
| 0xD4   | 31           | 30   | 29   | 28   | 27   | 26   | 25   | 24   | 23   | 22   | 21   | 20   | 19   | 18   | 17   | 16   | 15   | 14   | 13   | 12   | 11   | 10   | 9   | 8   | 7   | 6   | 5   | 4   | 3   | 2   | 1   | 0   |    |
| Reset  | 0x00000000   |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |     |     |     |     |     |     |     |     |     |     |    |
| Name   | CR31         | CR30 | CR29 | CR28 | CR27 | CR26 | CR25 | CR24 | CR23 | CR22 | CR21 | CR20 | CR19 | CR18 | CR17 | CR16 | CR15 | CR14 | CR13 | CR12 | CR11 | CR10 | CR9 | CR8 | CR7 | CR6 | CR5 | CR4 | CR3 | CR2 | CR1 | CR0 |    |
| Access | RW           | RW   | RW   | RW   | RW   | RW   | RW   | RW   | RW   | RW   | RW   | RW   | RW   | RW   | RW   | RW   | RW   | RW   | RW   | RW   | RW   | RW   | RW  | RW  | RW  | RW  | RW  | RW  | RW  | RW  | RW  | RW  | RW |

| Bit  | Name            | Function       | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | Access |
|------|-----------------|----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:0 | CRn<br>(n=31-0) | Cancel request | <p>Corresponding to the cancel request bit of the transmit buffer 31-0:</p> <p>Write 1 to set the corresponding cancel request bit, write 0 has no effect</p> <p>Read:</p> <ul style="list-style-type: none"> <li>● 0b0: No cancel request pending</li> <li>● 0b1: Cancel request pending</li> </ul> <p>The host can set the cancel request for multiple transmit buffers at a time. Only the TXBCR bits corresponding to the transmit buffers configured by MCAN_TXBC will be set. These bits remain set until the corresponding MCAN_TXBRP bits are re-set.</p> <div>  <b>Note</b><br/> Each bit corresponds to a buffer. </div> | RW     |

## 37.4.40 MCAN Transmit Buffer Transmit Occurred Register (MCAN\_TXBTO)

| Offset | Bit Position |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |     |     |     |     |     |     |     |     |     |     |    |
|--------|--------------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|----|
| 0xD8   | 31           | 30   | 29   | 28   | 27   | 26   | 25   | 24   | 23   | 22   | 21   | 20   | 19   | 18   | 17   | 16   | 15   | 14   | 13   | 12   | 11   | 10   | 9   | 8   | 7   | 6   | 5   | 4   | 3   | 2   | 1   | 0   |    |
| Reset  | 0x00000000   |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |     |     |     |     |     |     |     |     |     |     |    |
| Name   | T031         | T030 | T029 | T028 | T027 | T026 | T025 | T024 | T023 | T022 | T021 | T020 | T019 | T018 | T017 | T016 | T015 | T014 | T013 | T012 | T011 | T010 | T09 | T08 | T07 | T06 | T05 | T04 | T03 | T02 | T01 | T00 |    |
| Access | RO           | RO   | RO   | RO   | RO   | RO   | RO   | RO   | RO   | RO   | RO   | RO   | RO   | RO   | RO   | RO   | RO   | RO   | RO   | RO   | RO   | RO   | RO  | RO  | RO  | RO  | RO  | RO  | RO  | RO  | RO  | RO  | RO |

| Bit  | Name            | Function              | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | Access |
|------|-----------------|-----------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:0 | TOn<br>(n=31-0) | Transmit has occurred | <p>The corresponding transmit buffer 31-0 has transmit flag bit:</p> <ul style="list-style-type: none"> <li>● 0b0: Not transmit</li> <li>● 0b1: Transmit</li> </ul> <p>When the transmit buffer is completed and the corresponding bit of MCAN_TXBRP is cleared, the corresponding transmit has occurred flag is set. When writing 1 to the corresponding bit of MCAN_TXBAR to request a new transmission, the corresponding transmission has occurred flag is reset.</p> <div>  <b>Note</b><br/>           Each bit corresponds to a buffer.         </div> | RO     |

## 37.4.41 MCAN Transmit Buffer Cancel Completed Register (MCAN\_TXBCF)

| Offset | Bit Position |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |     |     |     |     |     |     |     |     |     |     |    |
|--------|--------------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|----|
| 0xDC   | 31           | 30   | 29   | 28   | 27   | 26   | 25   | 24   | 23   | 22   | 21   | 20   | 19   | 18   | 17   | 16   | 15   | 14   | 13   | 12   | 11   | 10   | 9   | 8   | 7   | 6   | 5   | 4   | 3   | 2   | 1   | 0   |    |
| Reset  | 0x00000000   |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |     |     |     |     |     |     |     |     |     |     |    |
| Name   | CF31         | CF30 | CF29 | CF28 | CF27 | CF26 | CF25 | CF24 | CF23 | CF22 | CF21 | CF20 | CF19 | CF18 | CF17 | CF16 | CF15 | CF14 | CF13 | CF12 | CF11 | CF10 | CF9 | CF8 | CF7 | CF6 | CF5 | CF4 | CF3 | CF2 | CF1 | CF0 |    |
| Access | RO           | RO   | RO   | RO   | RO   | RO   | RO   | RO   | RO   | RO   | RO   | RO   | RO   | RO   | RO   | RO   | RO   | RO   | RO   | RO   | RO   | RO   | RO  | RO  | RO  | RO  | RO  | RO  | RO  | RO  | RO  | RO  | RO |

| Bit  | Name            | Function               | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | Access |
|------|-----------------|------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:0 | CFn<br>(n=31-0) | Cancellation completed | <p>Corresponding to the cancellation completed flag of the transmit buffer 31-0:</p> <ul style="list-style-type: none"> <li>0b0: No transmit buffer cancellation</li> <li>0b1: Transmit buffer canceled</li> </ul> <p>When the cancellation requested by MCAN_TXBCR is completed and the corresponding bit of MCAN_TXBRP is cleared, the corresponding cancellation completed flag is set. If the corresponding bit of MCAN_TXBRP is not set at the time of cancellation, the cancellation completed flag is set immediately. When writing 1 to the corresponding bit of MCAN_TXBAR to request a new transmission, the corresponding cancellation completed flag is reset.</p> <div>  <b>Note</b><br/> Each bit corresponds to a buffer. </div> | RO     |

## 37.4.42 MCAN Transmit Buffer Transmit Interrupt Enable Register (MCAN\_TXBTIE)

| Offset | Bit Position |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |      |      |      |      |      |      |      |      |      |      |    |
|--------|--------------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|------|------|------|------|------|------|------|------|------|------|----|
| 0xE0   | 31           | 30    | 29    | 28    | 27    | 26    | 25    | 24    | 23    | 22    | 21    | 20    | 19    | 18    | 17    | 16    | 15    | 14    | 13    | 12    | 11    | 10    | 9    | 8    | 7    | 6    | 5    | 4    | 3    | 2    | 1    | 0    |    |
| Reset  | 0x00000000   |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |      |      |      |      |      |      |      |      |      |      |    |
| Name   | TIE31        | TIE30 | TIE29 | TIE28 | TIE27 | TIE26 | TIE25 | TIE24 | TIE23 | TIE22 | TIE21 | TIE20 | TIE19 | TIE18 | TIE17 | TIE16 | TIE15 | TIE14 | TIE13 | TIE12 | TIE11 | TIE10 | TIE9 | TIE8 | TIE7 | TIE6 | TIE5 | TIE4 | TIE3 | TIE2 | TIE1 | TIE0 |    |
| Access | RW           | RW    | RW    | RW    | RW    | RW    | RW    | RW    | RW    | RW    | RW    | RW    | RW    | RW    | RW    | RW    | RW    | RW    | RW    | RW    | RW    | RW    | RW   | RW   | RW   | RW   | RW   | RW   | RW   | RW   | RW   | RW   | RW |

| Bit  | Name                         | Function                   | Description                                                                                                                                                                                                                                                                                                                                                                   | Access |
|------|------------------------------|----------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:0 | TIE <sub>n</sub><br>(n=31-0) | Transmit interrupt enabled | Corresponding to the transmit interrupt enable bit of the transmit buffer 31-0: <ul style="list-style-type: none"> <li>0b0: Disable transmit interrupt</li> <li>0b1: Enable transmit interrupt</li> </ul> <div>  <b>Note</b><br/>           Each bit corresponds to a buffer.         </div> | RW     |

## 37.4.43 MCAN Transmit Buffer Cancel Completed Interrupt Enable Register (MCAN\_TXBCIE)

| Offset | Bit Position |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |       |       |       |       |       |       |       |       |       |       |    |
|--------|--------------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|----|
| 0xE4   | 31           | 30     | 29     | 28     | 27     | 26     | 25     | 24     | 23     | 22     | 21     | 20     | 19     | 18     | 17     | 16     | 15     | 14     | 13     | 12     | 11     | 10     | 9     | 8     | 7     | 6     | 5     | 4     | 3     | 2     | 1     | 0     |    |
| Reset  | 0x00000000   |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |       |       |       |       |       |       |       |       |       |       |    |
| Name   | CFIE31       | CFIE30 | CFIE29 | CFIE28 | CFIE27 | CFIE26 | CFIE25 | CFIE24 | CFIE23 | CFIE22 | CFIE21 | CFIE20 | CFIE19 | CFIE18 | CFIE17 | CFIE16 | CFIE15 | CFIE14 | CFIE13 | CFIE12 | CFIE11 | CFIE10 | CFIE9 | CFIE8 | CFIE7 | CFIE6 | CFIE5 | CFIE4 | CFIE3 | CFIE2 | CFIE1 | CFIE0 |    |
| Access | RW           | RW     | RW     | RW     | RW     | RW     | RW     | RW     | RW     | RW     | RW     | RW     | RW     | RW     | RW     | RW     | RW     | RW     | RW     | RW     | RW     | RW     | RW    | RW    | RW    | RW    | RW    | RW    | RW    | RW    | RW    | RW    | RW |

| Bit  | Name                          | Function                               | Description                                                                                                                                                                                                                                                                                                                                                                                                                 | Access |
|------|-------------------------------|----------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:0 | CFIE <sub>n</sub><br>(n=31-0) | Cancel the completed interrupt enabled | Corresponding to the cancel completed interrupt enable bit of the transmit buffer 31-0: <ul style="list-style-type: none"> <li>0b0: Disable cancellation of completed interrupts</li> <li>0b1: Enable cancellation of completed interrupts</li> </ul> <div>  <b>Note</b><br/>           Each bit corresponds to a buffer.         </div> | RW     |

## 37.4.44 MCAN Transmit Event FIFO Configuration Register (MCAN\_TXEFC)

| Offset | Bit Position |      |    |    |    |    |    |    |          |     |    |    |    |    |    |    |            |    |    |    |    |    |   |   |   |   |   |   |   |   |   |          |
|--------|--------------|------|----|----|----|----|----|----|----------|-----|----|----|----|----|----|----|------------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|----------|
| 0xF0   | 31           | 30   | 29 | 28 | 27 | 26 | 25 | 24 | 23       | 22  | 21 | 20 | 19 | 18 | 17 | 16 | 15         | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0        |
| Reset  | 0x00000000   |      |    |    |    |    |    |    |          |     |    |    |    |    |    |    |            |    |    |    |    |    |   |   |   |   |   |   |   |   |   |          |
| Name   | Reserved     | EFWM |    |    |    |    |    |    | Reserved | EFS |    |    |    |    |    |    | EFSA[15:2] |    |    |    |    |    |   |   |   |   |   |   |   |   |   | Reserved |
| Access |              | RWA  |    |    |    |    |    |    |          | RWA |    |    |    |    |    |    | RWA        |    |    |    |    |    |   |   |   |   |   |   |   |   |   |          |

| Bit   | Name       | Function                          | Description                                                                                                                                                                                                                                                                             | Access |
|-------|------------|-----------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:30 | Reserved   | Reserved                          | -                                                                                                                                                                                                                                                                                       | -      |
| 29:24 | EFWM       | Transmit event FIFO water level   | <ul style="list-style-type: none"> <li>0: Disable water level interrupt</li> <li>1-32: Level of transmit event FIFO water level interrupt (MCAN_IR.TEFW)</li> <li>&gt;32: Disable water level interrupt</li> </ul>                                                                      | RWA    |
| 23:22 | Reserved   | Reserved                          | -                                                                                                                                                                                                                                                                                       | -      |
| 21:16 | EFS        | Transmit event FIFO size          | <ul style="list-style-type: none"> <li>0: Disable transmit event FIFO</li> <li>1-32: Number of elements in transmit event FIFO</li> <li>&gt;32: Setting values greater than 32 are interpreted as 32</li> </ul> <p>The index of the transmit event FIFO element is 0 to EFS minus 1</p> | RWA    |
| 15:2  | EFSA[15:2] | Transmit event FIFO start address | The starting address of the transmit event FIFO in the message RAM (32-bit word address, see " <a href="#">Message RAM Configuration</a> ")                                                                                                                                             | RWA    |
| 1:0   | Reserved   | Reserved                          | -                                                                                                                                                                                                                                                                                       | -      |

## 37.4.45 MCAN Transmit Event FIFO Status Register (MCAN\_TXEFS)

| Offset | Bit Position |    |    |    |    |    |      |     |          |    |    |    |      |    |    |    |          |    |    |    |      |    |   |   |          |   |      |   |   |   |   |   |  |  |
|--------|--------------|----|----|----|----|----|------|-----|----------|----|----|----|------|----|----|----|----------|----|----|----|------|----|---|---|----------|---|------|---|---|---|---|---|--|--|
| 0xF4   | 31           | 30 | 29 | 28 | 27 | 26 | 25   | 24  | 23       | 22 | 21 | 20 | 19   | 18 | 17 | 16 | 15       | 14 | 13 | 12 | 11   | 10 | 9 | 8 | 7        | 6 | 5    | 4 | 3 | 2 | 1 | 0 |  |  |
| Reset  | 0x00000000   |    |    |    |    |    |      |     |          |    |    |    |      |    |    |    |          |    |    |    |      |    |   |   |          |   |      |   |   |   |   |   |  |  |
| Name   | Reserved     |    |    |    |    |    | TEFL | EFF | Reserved |    |    |    | EFPI |    |    |    | Reserved |    |    |    | EFGI |    |   |   | Reserved |   | EFFL |   |   |   |   |   |  |  |
| Access |              |    |    |    |    |    | RO   | RO  |          |    |    |    | RO   |    |    |    |          |    |    |    | RO   |    |   |   |          |   | RO   |   |   |   |   |   |  |  |

| Bit   | Name     | Function                           | Description                                                                                                                                                                                                                                                                                                                                                    | Access |
|-------|----------|------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:26 | Reserved | Reserved                           | -                                                                                                                                                                                                                                                                                                                                                              | -      |
| 25    | TEFL     | Transmit event FIFO element lost   | This bit is a copy of the interrupt flag MCAN_IR.TEFL. When MCAN_IR.TEFL is reset, this bit is also reset: <ul style="list-style-type: none"> <li>0b0: No element is lost in the transmit event FIFO</li> <li>0b1: An element is lost in the transmit event FIFO; this bit is also set when writing to a transmit event FIFO with a space size of 0</li> </ul> | RO     |
| 24    | EFF      | Transmission event FIFO is full    | <ul style="list-style-type: none"> <li>0: Transmission event FIFO is not full</li> <li>1: Transmission event FIFO is full</li> </ul>                                                                                                                                                                                                                           | RO     |
| 23:21 | Reserved | Reserved                           | -                                                                                                                                                                                                                                                                                                                                                              | -      |
| 20:16 | EFPI     | Transmission event FIFO puts index | Transmission event FIFO writes index pointer, the active range is 0-31                                                                                                                                                                                                                                                                                         | RO     |
| 15:13 | Reserved | Reserved                           | -                                                                                                                                                                                                                                                                                                                                                              | -      |
| 12:8  | EFGI     | Transmission event FIFO get index  | Transmission event FIFO read index pointer, the active range is 0-31                                                                                                                                                                                                                                                                                           | RO     |
| 7:6   | Reserved | Reserved                           | -                                                                                                                                                                                                                                                                                                                                                              | -      |
| 5:0   | EFFL     | Transmission event FIFO fill level | The number of elements stored in the transmit event FIFO. The active range is 0-32.                                                                                                                                                                                                                                                                            | RO     |

## 37.4.46 MCAN Transmit Event FIFO Acknowledge Register (MCAN\_TXEFA)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |      |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|------|---|---|---|
| 0xF8   | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3    | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |      |   |   |   |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   | EFAI |   |   |   |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   | RW   |   |   |   |

| Bit  | Name     | Function                              | Description                                                                                                                                                                                                                                                                                                    | Access |
|------|----------|---------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:5 | Reserved | Reserved                              | -                                                                                                                                                                                                                                                                                                              | -      |
| 4:0  | EFAI     | Transmit event FIFO acknowledge index | After the host reads an element or sequence of elements from the transmit event FIFO, it must write the index of the last element read from the event FIFO to EFAI. This operation sets the transmit event FIFO get index MCAN_TXEFS.EFGI to EFAI plus 1 and updates the event FIFO fill level MCAN_TXEFS.EFFL | RW     |

## 37.4.47 MCAN Message RAM Check Control Register (MCANRAM\_CKCR)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |        |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|--------|
| 0x00   | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0      |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |        |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   | ECCMOD |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   | RW     |

| Bit  | Name     | Function                            | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | Access |
|------|----------|-------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:2 | Reserved | Reserved                            | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | -      |
| 1:0  | ECCMOD   | ECC check enable bit of message RAM | <ul style="list-style-type: none"> <li>● 0b00: Disable ECC check function.</li> <li>● 0b01: <ul style="list-style-type: none"> <li>▶ If a single-bit error occurs, ECC corrects it without generating a single-bit error flag or interrupt.</li> <li>▶ If a double-bit error occurs, ECC detects the error, sets the double-bit error flag, and generates an interrupt.</li> </ul> </li> <li>● 0b10: <ul style="list-style-type: none"> <li>▶ If a single-bit error occurs, ECC corrects the error, sets the single-bit error flag, and no interrupt is generated.</li> <li>▶ If a double-bit error occurs, ECC detects the error, sets the double-bit error flag, and generates an interrupt.</li> </ul> </li> <li>● 0b11: <ul style="list-style-type: none"> <li>▶ If a single-bit error occurs, ECC corrects the error, sets the single-bit error flag, and generates an interrupt.</li> <li>▶ If a double-bit error occurs, ECC detects the error, sets the double-bit error flag, and generates an interrupt.</li> </ul> </li> </ul> | RW     |

**Note**

1. When using the ECC function of the message RAM, you must configure ECCMOD before reading or writing accesses to the message RAM, and then initialize the used message RAM space in units of words.
2. When ECCMOD is 0b10 or 0b11, if MCAN1/MCAN2 generates a 1-bit error when accessing the message RAM, MCAN\_IR.BEC of MCAN1/MCAN2 is set, and if the corresponding interrupt is enabled, MCAN1/MCAN2 generates an interrupt.
3. When ECCMOD is 0b01, 0b10 or 0b11, if MCAN1/MCAN2 generates a 2-bit error when accessing the message RAM, MCAN\_IR.BEU of MCAN1/MCAN2 is set, and if the corresponding interrupt is enabled, MCAN1/MCAN2 generates an interrupt.



## 37.4.48 MCAN Message RAM Check Status Register (MCANRAM\_CKSR)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |          |          |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|----------|----------|
| 0x04   | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1        | 0        |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |          |          |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   | RAM_2ERR | RAM_1ERR |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   | RW       | RW       |

| Bit  | Name     | Function                          | Description                                                                                                                                                                                | Access |
|------|----------|-----------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:2 | Reserved | Reserved                          | -                                                                                                                                                                                          | -      |
| 1    | RAM_2ERR | Message RAM ECC 2-bit error flag. | Write 1 to clear the flag, write 0 has no effect<br>Read:<br><ul style="list-style-type: none"> <li>0b0: No 2-bit ECC error occurred.</li> <li>0b1: A 2-bit ECC error occurred.</li> </ul> | RW     |
| 0    | RAM_1ERR | Message RAM ECC 1-bit error flag  | Write 1 to clear the flag, write 0 has no effect<br>Read:<br><ul style="list-style-type: none"> <li>0b0: No 1-bit ECC error occurred.</li> <li>0b1: A 1-bit ECC error occurred.</li> </ul> | RW     |

## 37.4.49 MCAN Message RAM Error Injection Enable Register (MCANRAM\_EIEN)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |      |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|------|
| 0x08   | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0    |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |      |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   | EIEN |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   | RW   |

| Bit  | Name     | Function               | Description                                                                         | Access |
|------|----------|------------------------|-------------------------------------------------------------------------------------|--------|
| 31:1 | Reserved | Reserved               | -                                                                                   | -      |
| 0    | EIEN     | Error injection enable | <ul style="list-style-type: none"> <li>0b0: Disable</li> <li>0b1: Enable</li> </ul> | RW     |

## 37.4.50 MCAN Message RAM Error Injection Bit Enable Register 0 (MCANRAM\_EIBIT0)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0x0C   | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | EIBIT[31:0]  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access | RW           |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit  | Name        | Function                    | Description                                                                                                                                                                                                                                                                                                                                                                                           | Access |
|------|-------------|-----------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:0 | EIBIT[31:0] | Error injection bit enabled | Controls whether the 31-0 bits of the message RAM output data are inverted: <ul style="list-style-type: none"> <li>0b0: No inversion, no ECC error injection for this bit.</li> <li>0b1: Invert, inject ECC error for this bit.</li> </ul> <div>  <b>Note</b><br/>Each bit corresponds to an output data bit. </div> | RW     |

## 37.4.51 MCAN Message RAM Error Injection Bit Enable Register 1 (MCANRAM\_EIBIT1)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |              |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|--------------|---|---|---|---|---|---|---|
| 0x10   | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7            | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |              |   |   |   |   |   |   |   |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   | EIBIT[38:32] |   |   |   |   |   |   |   |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   | RW           |   |   |   |   |   |   |   |

| Bit  | Name         | Function                    | Description                                                                                                                                                                                                                                                                                                                                                                                              | Access |
|------|--------------|-----------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:7 | Reserved     | Reserved                    | -                                                                                                                                                                                                                                                                                                                                                                                                        | -      |
| 6:0  | EIBIT[38:32] | Error injection bit enabled | Controls whether the 38-32 bits of the message RAM output data are inverted: <ul style="list-style-type: none"> <li>0b0: No inversion, no ECC error injection for this bit.</li> <li>0b1: Invert, inject ECC error for this bit.</li> </ul> <div>  <b>Note</b><br/>Each bit corresponds to an output data bit. </div> | RW     |

### 37.4.52 MCAN Message RAM Check Error Address Register (MCANRAM\_ECCERRADDR)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |            |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|------------|----|---|---|---|---|---|---|---|---|---|---|
| 0x14   | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11         | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |            |    |   |   |   |   |   |   |   |   |   |   |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | ECCERRADDR |    |   |   |   |   |   |   |   |   |   |   |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |            |    |   |   |   |   |   |   |   |   |   |   |

| Bit   | Name       | Function            | Description                                                                                                                                        | Access |
|-------|------------|---------------------|----------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:12 | Reserved   | Reserved            | -                                                                                                                                                  | -      |
| 11:0  | ECCERRADDR | Check Error Address | Records the access address when a 1-bit or 2-bit ECC error occurs in the message RAM (the offset address relative to the message RAM base address) | RO     |

## 37.5 Precautions

### 37.5.1 CAN Bus Anti-Interference Measures

CAN bus is widely used in the automotive, industrial control and other industries. If the electromagnetic environment at the CAN application site is relatively bad, with factors such as circuit imbalance, spatial electromagnetic field, and power grid incoming lines, the CAN bus will generate a lot of communication noise due to radiation and conduction interference, resulting in an increase in bus error frames, frequent retransmissions, and the failure of correct data to arrive in time. It may also lead to undefined errors in the CAN protocol, bus anomalies and other unexpected situations, seriously affecting the quality of data communication. Therefore, in practical applications, efforts should be made to eliminate noise interference and ensure stable operation of the CAN bus network.

The following are several commonly used CAN bus anti-interference measures (including but not limited to) :

1. Add electrical isolation for the CAN bus interface
2. Common signal ground for transceivers
3. Use shielded twisted pair cables and properly ground them
4. Increase the degree of twisting of CAN transmission lines
5. Add signal protector
6. Improve network topology
7. Anti-interference mechanisms at the application layer software

### 37.5.2 CAN Controller Noise Constraints

In the CAN bus network, it should be ensured that the bit time of the communication meets the requirements of the standard protocol. If the noise interference that does not meet the bit time width is introduced, it may cause the abnormal operation of the MCAN controller.

## 38 Ethernet MAC Controller (ETHMAC)

### 38.1 Introduction

Ethernet MAC controller (ETHMAC) is used to send and receive data in Ethernet network according to IEEE802.3-2002 standard, and it has many application fields, such as switches and network interface cards. The MAC controller supports two industrial standard interfaces connected with external physical layer (PHY) : Media Independent Interface (MII) (defined in IEEE802.3 specification) and Simplified Media Independent Interface (RMII).

Mainly follow the following protocol specifications:

- IEEE802.3-2002, used for Ethernet MAC
- IEEE1588-2008 standard, used to specify network clock synchronization
- AMBA2.0, for AHB master/slave ports
- RMII interface specification

### 38.2 Features

#### 38.2.1 ETH\_MAC Features

The basic functional features of the ETH\_MAC module are as follows:

- Support for external PHY interface to achieve 10/100Mbps data transfer rate
- Support for MII or RMII interface to communicate with external fast Ethernet PHY
- Provide separate transmit, receive, and control interfaces for applications
- Use SMI interface to configure and manage up to 32 PHY devices
- Support for detecting remote wakeup frames and AMD Magic Packet™ frames
- Support for LoopBack mode with internal loopback via MII
- Support for full-duplex and half-duplex operations:
  - ▶ Support for CSMA/CD protocol suitable for half-duplex operation
  - ▶ Support for backpressure flow control suitable for half-duplex operation
  - ▶ Support for IEEE802.3x flow control suitable for full-duplex operation
  - ▶ In full-duplex operation, the received pause control frame can be forwarded to the application



#### Note

The "application" referred to in this section means that the MAC transmits the received frame to the system memory through RxDMA for use by the upper-layer network layer and transport layer; or it transmits the data from the upper-layer network layer and transport layer from the system controller to the MAC through TxDMA. The upper-layer network layer and transport layer are called applications.

- ▶ In full-duplex operation, if the flow control input signal disappears, a zero-time pause frame is sent automatically
- Ethernet frame processing:
  - ▶ When receiving an Ethernet frame, automatically remove the PAD and FCS fields of frames with length fields less than 1536
  - ▶ When receiving an Ethernet frame, automatically remove the FCS field of the type frame
  - ▶ When receiving an Ethernet frame, automatically perform CRC calculation of the received frame

- ▶ When sending an Ethernet frame, insert and replace the SA
- ▶ When sending an Ethernet frame, insert and replace the CRC
- ▶ When sending an Ethernet frame, insert, replace, and delete the VLAN frame identifier
- ▶ When sending an Ethernet frame, automatically generate PAD filling for frames smaller than 60 bytes
- ▶ When sending an Ethernet frame, automatically retransmit conflict frames
- ▶ When sending an Ethernet frame, support programmable frame intervals (40-96 bit times, with 8 as the step length)
- ▶ When receiving or sending an Ethernet frame, the frame length can be programmed, supporting giant frames up to 16KB
- Ethernet frame address filtering:
  - ▶ Support up to five 48-bit perfect (DA) address filters, performing mask operations on each byte
  - ▶ Support up to four 48-bit perfect (SA) address filters, performing mask operations on each byte
  - ▶ Support for 64-bit Hash filter, suitable for unicast and multicast target address filtering
  - ▶ Can transmit all multicast address frames
  - ▶ Support for mixed mode, can transmit all frames, no need to filter for network monitoring
  - ▶ A status report is attached when transmitting all incoming data packets (each time filtered)
- VLAN identifier filtering:
  - ▶ Supports a VLAN tag filtering set
  - ▶ Support for 12-bit/16-bit VLAN tag selection
  - ▶ Support for VLAN mark Hash filter
- LAY3 IP message address filtering:
  - ▶ Support for a set of LAY3 IP message address filters, performing mask operations on each byte
  - ▶ Support for address filtering selection of IPv4 messages and IPv6 messages
- LAY4 TCP/UDP message port filtering:
  - ▶ Support for a set of LAY4 TCP/UDP message port filters
- Remote wake-up frame filtering mode
  - ▶ Support for four sets of remote wakeup frame filters, controlling four sets of mask operations
- COE engine:
  - ▶ When receiving frames, you can choose to discard frames without TCP/UDP fields
  - ▶ Check the Header Checksum field of the received IPv4 message
  - ▶ Check the Checksum field of the received TCP messages, UDP messages, and ICMP messages
  - ▶ Perform Checksum calculation for the sent IPv4 message and insert the result into the Header Checksum field
  - ▶ Perform Checksum calculation for the sent TCP messages, UDP messages, and ICMP messages, and insert the result into the Checksum field
- Support for MMC counters to perform network statistics
- FIFO processing:
  - ▶ A 2KB transmit FIFO (Tx FIFO) with a programmable threshold and a 2KB receive FIFO (Rx FIFO) with a configurable threshold
  - ▶ Both Rx FIFO and Tx FIFO support store-and-forward mode
  - ▶ Software-controlled Tx FIFO flush

- ▶ In store-and-forward mode, Rx FIFO can choose to filter out all error frames and not forward these error frames to the application
- ▶ Rx FIFO can automatically generate pause control frames or backpressure signals to be sent to the MAC core based on the fill level (configurable threshold)
- ▶ Frames lost or corrupted in the Rx FIFO can be counted
- ▶ When the Rx FIFO stores multiple frames, by inserting a receive status vector into the Rx FIFO after the EOF transfer, the Rx FIFO does not need to store the receive status of these frames

### 38.2.2 ETH\_PTP Features

The basic functional features of the ETH\_PTP module are as follows:

- Support for taking timestamp snapshots of received and transmitted frames and inserting time-stamps
- Support for coarse and fine calibration of system time
- Support for 2 sets of target time settings, interrupts are triggered when the system time is greater than the target time
- Support for 2 sets of PPS outputs, which can be used as event outputs to control the actions of other modules

### 38.2.3 ETH\_DMA Features

The basic functional features of the ETH\_DMA are as follows:

- Support for software setting of the AHB burst type of the AHB Master interface (fixed or undefined burst)
- Support for all AHB burst types in the AHB Slave interface
- The AHB Master interface can choose address-aligned burst transfers
- Support for RxDMA and TxDMA engines to program burst lengths separately to fully utilize the bus
- Support for byte-aligned addressing of the data buffer
- Descriptor features:
  - ▶ Support for two types of descriptor linking methods: ring and chain
  - ▶ The descriptor architecture can transmit large data blocks with almost no CPU intervention
  - ▶ Each descriptor can transmit up to 8KB of data
- Both TxDMA and RxDMA support store-and-forward mode
- Report the comprehensive status of normal operation and transmission errors
- Programmable interrupt options, choose different interrupts according to different application scenarios
- Frame-controlled send/receive complete interrupt
- Use circular scheduling arbitration or fixed priority arbitration between the receive engine and the transmit engine

## 38.3 Functional Description

The Ethernet MAC controller follows the IEEE802.3 international standard for local area networks (LAN), using the CSMA/CD (Carrier Sense Multiple Access with Collision Detection) method as the basic communication method. It includes a MAC802.3 (Media Access Control) controller with MII and RMII interfaces and a dedicated DMA controller.

### 38.3.1 Basic Block Diagram

The figure below is the architecture diagram of the Ethernet MAC controller.

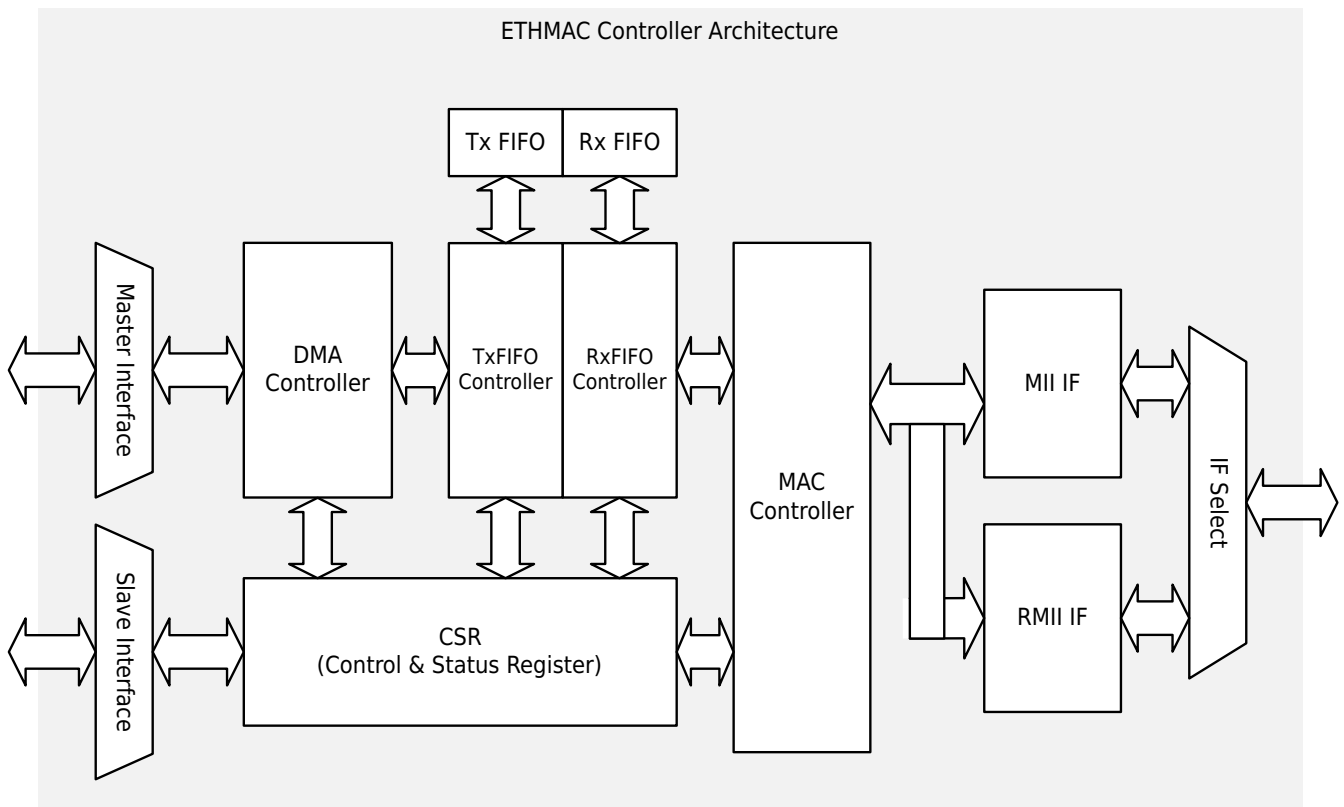


Figure 38-1 Ethernet MAC Controller Architecture Diagram

From the figure, it can be seen that the core part of the Ethernet MAC controller includes the AHB Master interface, AHB Slave interface, Rx FIFO, Tx FIFO, DMA controller, MAC controller, interface control logic, etc. The DMA controller connects the MAC core to the system memory through the AHB master-slave interface.

The AHB Master interface is used for data transfer, and the AHB Slave interface is used for basic register access configuration of the controller.

When transmitting data, first send the data from the system memory to the Tx FIFO for buffering through the DMA controller, then send it to the external PHY through the MII interface or the RMII interface after a series of processing by the MAC controller (COE control, PTP control, MMC control, etc.).

When receiving data, the data received through the MII interface or the RMII interface is sent to the Rx FIFO after a series of processing by the MAC controller (COE control, PTP control, MMC control, etc.), and then transmit the data from the Rx FIFO to the system memory by DMA controller.

### 38.3.2 Interface Description

The Ethernet MAC controller supports MII, RMII, and SMI interface types. Users can choose one of the MII and RMII interfaces according to actual needs for data exchange. The SMI interface exists independently and is used for basic configuration-related data exchange with the PHY.

By setting the IFSEL bit of the interface control register (ETH\_MAC\_IFCONFR), you can choose whether the Ethernet uses the MII interface or the RMII interface. This bit must be set first before configuring other features of ETHMAC.

#### 38.3.2.1 MII Interface

The Medium Independent Interface (MII) defines the interconnection between the MAC sublayer and the PHY at data transfer rates of 10Mbps and 100Mbps. The figure below shows the basic connection diagram of the MII interface.

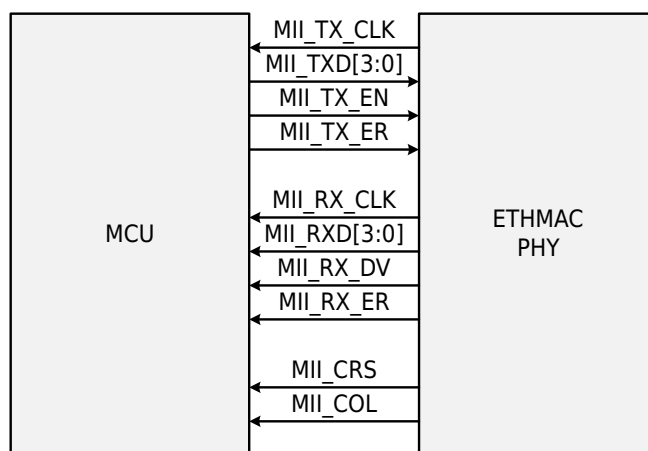


Figure 38-2 MII Interface Connection Diagram

The following table introduces the functions of each signal in the MII interface.

Table 38-1 MII Interface Signal Description

| Interface Signal Name | Description                                                                                                                                                                                                                                                                                                                                |
|-----------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| MII_TX_CLK            | Sends Clock Signal.<br>This signal provides the reference timing for TX data transmission. It is 2.5MHz at a rate of 10Mbps; it is 25MHz at a rate of 100Mbps.                                                                                                                                                                             |
| MII_RX_CLK            | Receive Clock Signal.<br>This signal provides the reference timing for RX data transmission. It is 2.5MHz at a rate of 10Mbps; it is 25MHz at a rate of 100Mbps.                                                                                                                                                                           |
| MII_TX_EN             | Send Enable Signal.<br>This signal indicates that the MAC is currently transmitting data to the MII.                                                                                                                                                                                                                                       |
| MII_TX_ER             | Send Error Signal.                                                                                                                                                                                                                                                                                                                         |
| MII_TXD[3:0]          | Send Data Signal.<br>This signal is a group of four data signals, synchronously driven by the MAC sublayer, and is only active when the MII_TX_EN signal is enable. MII_TXD[0] is the least significant bit, and MII_TXD[3] is the most significant bit. When the MII_TX_EN signal is disabled, transmitting data does not affect the PHY. |



| Interface Signal Name | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|-----------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| MII_RX_DV             | Receive Data Active Signal.<br>This signal indicates that the PHY is currently sending data to the MII. To correctly receive a frame, this signal must cover the frame to be received in the time range, and its start time must not be later than the time when the SFD field appears.                                                                                                                                                                                                                            |
| MII_RX_ER             | Receive Error Signal.<br>This signal must be maintained for one or more transmit clock periods to indicate to the MAC sublayer that an error is detected somewhere in the frame. This error condition must be verified by the RX_DV signal (see <a href="#">Table 38-3 : Signal State Description when Receiving Data</a> ).                                                                                                                                                                                       |
| MII_RXD[3:0]          | Receive Data Signal.<br>This signal is a group of 4 data signals, driven synchronously by the external PHY, and is a active signal only when the MII_RX_DV signal is enable. MII_RXD[0] is the least significant bit, and MII_RXD[3] is the most significant bit. When the MII_RX_DV signal is disabled and the MII_RX_ER signal is enabled, specific MII_RXD[3:0] values are used to transmit specific information from the PHY (see <a href="#">Table 38-3 : Signal State Description when Receiving Data</a> ). |
| MII_CRS               | Carrier Sense Signal.<br>When the transmit or receive medium is not in an idle state, this signal is enabled by the PHY; when both the transmit and receive media are in an idle state, this signal is disabled by the PHY. The PHY must ensure that the MII_CRS signal remains active under conflict conditions. This signal does not need to be synchronized with the TX and RX clocks. This signal is meaningless in full-duplex mode.                                                                          |
| MII_COL               | Conflict Detection Signal.<br>After detecting a conflict on the medium, the PHY must immediately enable the conflict detection signal, and as long as the conflict condition exists, the conflict detection signal must remain active. This signal does not need to be synchronized with the TX and RX clocks. This signal is meaningless in full-duplex mode.                                                                                                                                                     |

The following table introduces the signal encoding states when sending data.

**Table 38-2 Signal State Description When Sending Data**

| MII_TX_EN | MII_TX_ER | MII_TXD[3:0]  | State Description                           |
|-----------|-----------|---------------|---------------------------------------------|
| 0b0       | 0b0       | 0b0000-0b1111 | Normal frame interval, no data transmission |
| 0b0       | 0b1       | 0b0000-0b1111 | Meaningless                                 |
| 0b1       | 0b0       | 0b0000-0b1111 | Normal data transmission                    |
| 0b1       | 0b1       | 0b0000-0b1111 | Error in data transmission                  |

The following table introduces the signal encoding states when receiving data.

**Table 38-3 Signal State Description when Receiving Data**

| MII_RX_DV | MII_RX_ER | MII_RXD[3:0]  | State Description                        |
|-----------|-----------|---------------|------------------------------------------|
| 0b0       | 0b0       | 0b0000-0b1111 | Normal frame interval, no data reception |

| MII_RX_DV | MII_RX_ER | MII_RXD[3:0]  | State Description       |
|-----------|-----------|---------------|-------------------------|
| 0b0       | 0b1       | 0b0000        | Normal frame interval   |
|           |           | 0b0001-0b1101 | Meaningless             |
|           |           | 0b1110        | Error carrier detection |
|           |           | 0b1111        | Meaningless             |
| 0b1       | 0b0       | 0b0000-0b1111 | Normal data reception   |
| 0b1       | 0b1       | 0b0000-0b1111 | Errors in received data |

### 38.3.2.2 RMII Interface

The Reduced Medium Independent Interface (RMII) specification reduces the number of ports between the microcontroller Ethernet MAC controller and the external PHY at 10/100Mbps. The number of ports it uses is reduced from 16 of MII to 7, as shown in the figure below.

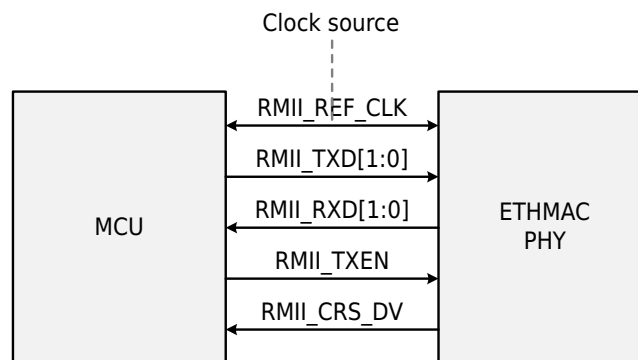


Figure 38-3 RMII Interface Connection Diagram

From the connection diagram, the following main differences can be seen between the RMII interface and the MII interface:

1. The transmit clock and receive clock are unified as the reference clock RMII\_REF\_CLK, which must be a 50MHz clock
2. The data lines are reduced from 4 to 2
3. The MII\_COL signal is discarded, and the MII\_RX\_DV and MII\_CRD signals are combined into the RMII\_CRD\_DV signal
4. MII\_TX\_ER and MII\_RX\_ER signals are discarded

### 38.3.2.3 SMI Interface

The SMI interface can implement station management of the PHY through the Ethernet MAC controller. The MAC controller accesses any PHY register through two lines of clock and data. This interface supports access to up to 32 PHYs. The figure below is the connection diagram of the SMI interface.



Figure 38-4 SMI Interface Connection Diagram

The functions of each signal within the SMI are described in the following table.

Table 38-4 SMI Interface Signal Description

| Interface Signal Name | Description                                                                                                                                                                                                                                                                                                  |
|-----------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| SMI_MDC               | Periodic clock, providing reference timing for data transmission at a maximum frequency of 2.5MHz. The shortest high level and the shortest low level of MDC must both be 160ns. The minimum period for the MDC signal must be 400ns. In the idle state, the SMI drives the MDC clock signal to a low level. |
| SMI_MDIO              | Data input/output bit stream, used for data information transfer with the PHY device through the MDC clock signal.                                                                                                                                                                                           |

The MAC controller can select one PHY from 32 PHYs, and then select one register from the 32 registers contained in the selected PHY to transmit control data or receive status information. Only one register in one PHY can be addressed at any given time.

The frame format of SMI is shown in the following table, and the following format is followed during data exchange with PHY:

Table 38-5 SMI Frame Format

|       | Detailed description of the SMI frame format |       |           |        |        |      |               |      |
|-------|----------------------------------------------|-------|-----------|--------|--------|------|---------------|------|
|       | Head (32-bit)                                | Start | Operation | PhyAdd | RegAdd | TA   | Data (16-bit) | Idle |
| Read  | 0b1...1                                      | 0b01  | 0b10      | 0bxxxx | 0bxxxx | Z0   | 0bxx...xx     | Z    |
| Write | 0b1...1                                      | 0b01  | 0b01      | 0bxxxx | 0bxxxx | 0b10 | 0bxx...xx     | Z    |

### 38.3.2.4 Ethernet Port Configuration

Based on the above three types of data interfaces, MII, RMII, and SMI, the correspondence between the port function allocation of the Ethernet MAC controller and each type of interface is organized in the following table:

Table 38-6 ETHMAC Port Function Allocation

| ETH Port Name       | Interface Function Signal |
|---------------------|---------------------------|
| ETH_MII_RMII_RXCLK  | MII_RX_CLK/RMII_REF_CLK   |
| ETH_MII_TXCLK       | MII_TX_CLK                |
| ETH_MII_RMII_TXEN   | MII_TX_EN/RMII_TX_EN      |
| ETH_MII_TXER        | MII_TX_ER                 |
| ETH_MII_RMII_TXD1-0 | MII_TXD[0]/RMII_TXD[0]    |
|                     | MII_TXD[1]/RMII_TXD[1]    |
| ETH_MII_TXD3-2      | MII_TXD[2]                |
|                     | MII_TXD[3]                |
| ETH_MII_RMII_RXDV   | MII_RX_DV/RMII_CRS_DV     |
| ETH_MII_RXER        | MII_RX_ER                 |
| ETH_MII_RMII_RXD1-0 | MII_RXD[0]/RMII_RXD[0]    |
|                     | MII_RXD[1]/RMII_RXD[1]    |

| ETH Port Name  | Interface Function Signal |
|----------------|---------------------------|
| ETH_MII_RXD3-2 | MII_RXD[2]                |
|                | MII_RXD[3]                |
| ETH_MII_CRS    | MII_CRS                   |
| ETH_MII_COL    | MII_COL                   |
| ETH_SMI_MDC    | SMI_MDC                   |
| ETH_SMI_MDIO   | SMI_MDIO                  |

### 38.3.3 ETH\_MAC Function

#### 38.3.3.1 ETH\_MAC Function Overview

The MAC sublayer performs the following functions related to data link control steps:

- Data package (transmit and receive)
  - ▶ Frame formation (frame boundary definition, frame synchronization)
  - ▶ Addressing (processing source and destination addresses)
  - ▶ Error detection
- Medium access management
  - ▶ Medium allocation (collision avoidance)
  - ▶ Contention resolution (collision handling)

The MAC sublayer has two main operating modes:

- Half-duplex mode: Stations use the CSMA/CD algorithm to contend for the physical medium
- Full-duplex mode: When the following conditions are met, competition issues do not need to be resolved (the CSMA/CD algorithm is not required) to send and receive data simultaneously:
  - ▶ The physical medium can support synchronous sending and receiving
  - ▶ Exactly 2 stations are connected to the LAN
  - ▶ Both stations are configured for full-duplex operating mode

#### 38.3.3.2 MAC 802.3 Frame Format

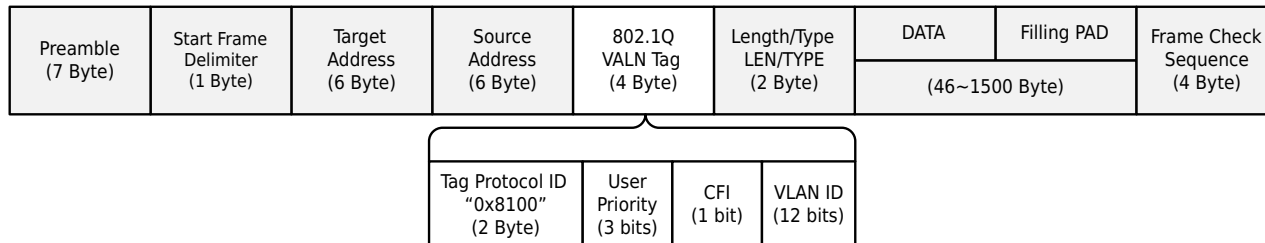
MAC data communication can use two frame formats:

- Basic MAC frame structure
- MAC frame format with VLAN tag (an extension of the basic MAC frame format)

The figure below shows the specific structures of the two frame formats.

*Basic MAC Frame Format*

|                      |                                      |                               |                               |                                     |                |             |                                     |
|----------------------|--------------------------------------|-------------------------------|-------------------------------|-------------------------------------|----------------|-------------|-------------------------------------|
| Preamble<br>(7 Byte) | Start Frame<br>Delimiter<br>(1 Byte) | Target<br>Address<br>(6 Byte) | Source<br>Address<br>(6 Byte) | Length/Type<br>LEN/TYPE<br>(2 Byte) | DATA           | Filling PAD | Frame Check<br>Sequence<br>(4 Byte) |
|                      |                                      |                               |                               |                                     | (46~1500 Byte) |             |                                     |

*MAC Frame Format with VLAN Flag***Figure 38-5 MAC Frame Structure**

Except for the frame check sequence (FCS), the Ethernet MAC controller sends each byte in the order of least significant bit first.

CRC calculation includes all bytes of frame data except for the preamble and start frame delimiter fields. The 32-bit CRC generator polynomial for Ethernet frames is 0x04C11DB7, and this polynomial is used for all 32-bit CRC calculations in the Ethernet module, as shown in the following formula:

$$G(x)=x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$$

A frame is defined as an inactive MAC frame by one of the following conditions:

- The frame length does not match the expected value specified by the length/type field. If the length/type field contains a type value, the frame length is considered consistent with this field (no inactive frames)
- The frame length is not an integer multiple of bytes (extra bits)
- The CRC value calculated based on the incoming frame does not match the included FCS

**38.3.3.3 MAC Frame Transmission Action**

TxDMA controls all transactions on the transmit path. The Ethernet frame read from the system memory is sent to the Tx FIFO by TxDMA and then transmitted to the MAC core. At the end of frame transmission, the transmit status is obtained from the MAC core and sent back to TxDMA.

The Ethernet MAC controller is equipped with a 2KB Tx FIFO. The Tx FIFO fill level is indicated to the DMA so that TxDMA can initiate data transfer in the required system memory via the AHB interface. Data from the AHB Master interface is sent to the Tx FIFO.

When SOF is detected, the MAC receives data and begins to send it to the MII. After the application starts sending, the time required to send frame data to the MII is variable, depending on delay factors such as IFG delay, time to send the header/SFD, and any backoff delay in half-duplex mode. After the EOF is transmitted to the MAC core, the core completes the normal transmission and then return the transmission status to the TxDMA. If a normal collision occurs during transmission (in half-duplex mode), the MAC core activeates the transmission status, then accept and discard all subsequent data until the next SOF is received. When a retry request from the MAC is detected (in the status), the same frame should be retransmitted starting from the SOF.

If data is not continuously provided during transmission, the MAC issues an underflow status. During the normal transmission of a frame, if the MAC receives an SOF without obtaining the EOF of the previous frame, it ignores the SOF and treat the new frame as a continuation of the previous frame.

There are two operating modes for delivering data to the MAC core:

- In threshold mode, once the number of bytes in the FIFO exceeds the configured threshold (or the end of the frame is written before exceeding the threshold), the data is ready and forwarded to the MAC core. This threshold can be configured using the TTC bit in ETH\_DMA\_OPRMODR.
- In store-and-forward mode, frames are delivered to the MAC core only after a complete frame is stored in the FIFO. If the size of the Tx FIFO is smaller than the Ethernet frame to be sent, the frame is delivered to the MAC core when the Tx FIFO is nearly full.

The application can empty all contents of the Tx FIFO by setting the FTF bit (ETH\_DMA\_OPRMODR register[20]) to 1. This bit self-clears and initializes the FIFO pointer to the default state. If this bit is set to 1 during frame transmission to the MAC core, the transmission stops, the Tx FIFO is considered empty, an underflow event occurs in the MAC transmitter, and the corresponding status word is forwarded to the TxDMA.

### Basic Protocol Transmission

The MAC controls the transmission of Ethernet frames. It performs the following functions to meet the IEEE802.3/802.3z specifications. Includes:

- Generates the header and SFD
- Generates a Jam signal in half-duplex mode
- Controls Jabber timeout
- Controls traffic (backpressure) in half-duplex mode
- Transmit frame status generation
- Includes IEEE1588-compliant timestamp snapshot logic

When a new frame is requested for transmission, the MAC transmits the header and SFD, followed by the transmit data. The header is defined as 7 bytes of "10101010" pattern, and the SFD is defined as 1 byte of "10101011" pattern.

The collision window is defined as 1 slot (512 bit times for 10/100Mbps Ethernet). Jam signal generation is only applicable in half-duplex mode and not in full-duplex mode. In MII mode, if a collision occurs at any time between the start of frame transmission and the end of the CRC field, the MAC transmits a 32-bit Jam signal of 0x55555555 on the MII to notify all other stations of the collision. If a collision occurs during the header transmission phase, the MAC completes the transmission of the header and SFD, then send the Jam signal.

The MAC core uses a Jabber timer to cut off Ethernet frame transmission when the transmitted bytes exceed 2048 bytes. In half-duplex mode, the MAC uses a deferral mechanism for traffic control (backpressure). When the application requests to stop receiving frames, if transmit flow control is enabled, the MAC transmits a 32-byte Jam signal whenever it detects a received frame. This causes a collision and makes the remote station back off. The application requests flow control by setting the BPA bit in the ETH\_MAC\_FLOCTLR register to 1. If the application requests to send a frame, it is transmitted as scheduled even if the backpressure function is active. If the backpressure function remains active for a long time (more than 16 consecutive collision events), the remote station aborts transmission due to excessive collisions. If the IEEE1588 timestamp function is enabled for a transmitted frame, a snapshot of the system time is captured when the SFD is placed on the transmit MII bus.

## Scheduling Algorithm Implementation

The MAC is responsible for scheduling frame transmission on the MII. The inter-frame gap between two transmitted frames is maintained by IFG settings or the truncated binary exponential backoff algorithm. After the IFG and backoff condition are met, the MAC enables transmission.

The IFG setting ensures an idle period between two transmitted frames, i.e., the configured inter-frame gap (IFG bit in the ETH\_MAC\_CONFIGR register). If a frame to be transmitted arrives before the configured IFG time, the MII waits for the enable signal from the MAC before starting transmission. Whenever the carrier signal of the MII becomes inactive, the MAC starts the IFG counter. When the IFG count equals the set value, the MAC enables transmission in full-duplex mode.

In half-duplex mode, when the IFG is configured as 96 bit times, the MAC follows the compliance rules specified in the IEEE802.3 standard. If a carrier is detected during the first two-thirds of the IFG interval (64 bit times for all IFG values), the MAC resets its IFG counter. If a carrier is detected during the last third of the IFG interval, the MAC continues the IFG count and enables the transmitter after the IFG interval ends.

In half-duplex mode, the truncated binary exponential backoff algorithm is implemented, controlled by the BL bit setting in the MAC configuration register (ETH\_MAC\_CONFIGR).

## Retransmission During Collision

In half-duplex mode, a collision event may occur on the MAC line interface during frame transmission to the MAC. The MAC may indicate a retry even before receiving the end of frame status, then enable retransmission and send the frame from the Tx FIFO again.

After more than 96 bytes are delivered to the MAC core, the Tx FIFO controller releases the space, allowing the DMA to deliver more data. This means retransmission is not possible after exceeding the threshold or when the MAC core indicates a delayed collision event.

## TxFIFO Flush

The MAC controller can empty the Tx FIFO by setting the FTF bit in the Operation Mode Register (ETH\_DMA\_OPRMODR). The emptying operation is immediate; even if the Tx FIFO is transmitting a frame to the MAC core, the Tx FIFO and its pointers is cleared to the initial state. This can cause an underflow event in the MAC transmitter, aborting frame transmission, and the frame status marks both underflow and flush events (FFF bit and UDE bit in TDES0). During the emptying operation, no data is transferred from the TxDMA to the Tx FIFO. The TxDMA transmits the corresponding number of transmission status words to the application based on the number of flushed frames (including partial frames). The frame flush status bit (FFF bit in TDES0) is set to 1 for fully flushed frames. The emptying operation is completed when the TxDMA accepts the status words of all flushed frames, and the FTF bit then is cleared. At this point, new frames from the TxDMA is accepted. After the emptying operation is completed, all submitted data is discarded unless the data starts with an SOF mark.

## CRC Processing and PAD Generation

When the number of bytes received from the application is less than 60 (DA+SA+LT+data), zeros (PAD) can be appended to the transmitted frame by resetting the DPAD bit in the transmit descriptor TDES0, making the data length exactly 46 bytes to meet the IEEE802.3 minimum data field requirement, or no padding can be added.

The MAC also calculates the cyclic redundancy check (CRC) for the frame check sequence (FCS) field. Depending on the settings of the DCRC bit and CRCR bit in the transmit descriptor TDES0, the checksum



value can be inserted or replaced into the transmitted data. For details, refer to "Normal Tx Descriptor" and "Enhanced Tx Descriptor".

**Note**

When the CRC replacement function is enabled, the enhanced Tx descriptor must be enabled.

If the MAC is programmed not to append the CRC value to the end of the Ethernet frame, the calculated CRC is not sent. However, when the DPAD bit in TDES0 is 0, frames smaller than 60 bytes (DA+SA+LT+data) have padding PAD appended after the data, and the CRC result is also automatically appended to the end of the padded frame.

### Source Address (SA) Processing

If the data from the application does not contain a source address field, the MAC can insert this field; if it contains a source address field, the MAC can choose to replace it with the values set in MAC Address Registers 0-1. The specific handling of the SA field can be controlled by the SAIRC bit in the MAC configuration register (ETH\_MAC\_CONFIGR) or the SAIRC bit in descriptor TDES1.

**Note**

When this function is enabled via the descriptor method, the enhanced Tx descriptor must be enabled.

### VLAN Tag Handling

The MAC can also insert, replace, or delete the identifier field of VLAN frames from the application. The specific handling can be controlled by the VLANC bit in the VLAN mark transmit control register (ETH\_MAC\_VTACTLR) or the VLANC bit in descriptor TDES0.

**Note**

When this function is enabled via the descriptor method, the enhanced Tx descriptor must be enabled.

### Transmit Status Word

At the end of Ethernet frame transmission to the MAC core, and after the core completes frame transmission, the transmission status is provided to the application via the transmit descriptor. For details on the transmission status, refer to bit [17:0] in TDES0. If the IEEE1588 timestamp function is enabled, a 64-bit timestamp for the specific frame is returned along with the transmission status.

### Pause Frame Transmission

In full-duplex mode, when the transmit flow control enable bit (TFE bit in ETH\_MAC\_FLOCTL) is set to 1, the MAC generates pause frames and send them as needed. Pause frames are sent along with the calculated CRC. Pause frame generation can be initiated in two ways.

- By setting the FCA bit in the ETH\_MAC\_FLOCTL register to 1 to request flow control, the MAC generates and send a single pause frame. The pause time value in the generated frame is the programmed pause time value in ETH\_MAC\_FLOCTL. To extend or end the pause time before the time specified in the previously sent pause frame, the user must program the PAUSET pause time value in the ETH\_MAC\_FLOCTL register with an appropriate value in the application, then request another pause frame transmission.
- When the Rx FIFO is full and the application has requested flow control, the MAC generates and send pause frames. The pause time value in the generated frame is the programmed pause time value in ETH\_MAC\_FLOCTL. If the Rx FIFO remains full during the configurable number of slots (PLT bit in ETH\_MAC\_FLOCTL) before this pause time ends, a second pause frame is sent. This process repeats



as long as the Rx FIFO remains full. If this condition is no longer met before the pause time ends, the MAC transmits a pause frame with a pause time of zero to indicate to the remote port that the Rx FIFO is ready to receive new data frames.

## COE Engine

In network interconnection, basic communication protocols (e.g., TCP and UDP) implement checksums and include checksum fields, which help determine the integrity of data sent over the network. Since the most common use of Ethernet is to package TCP and UDP in IP datagrams, the Ethernet MAC controller has a checksum offload function (COE engine), which supports checksum calculation and insertion in the transmit path, and checksum calculation and detection in the receive path. This section describes the operation of the COE engine in transmission; Section 30.4.1.3.6 describes its operation in reception.

The COE engine calculates the checksum for TCP, UDP, or ICMP over the complete frame and inserts it into the checksum field of the header. Due to this requirement, this function is enabled only when the Tx FIFO is configured in store-and-forward mode (TSF bit in ETH\_DMA\_OPRMODR register set to 1). If the core is configured in threshold mode, the COE engine is bypassed.

The Tx FIFO must be deep enough to store a complete frame for transmission to the MAC core Transmitter. If the FIFO depth is smaller than the input Ethernet frame size, the TCP/UDP/ICMP checksum insertion function is bypassed, and only the IPv4 header checksum insertion function is supported, even in store-and-forward mode.

Control of the transmit checksum offload function can be achieved by setting the CIC bits (TDES0[23:22]) for checksum control per frame.

### ● IP Header Checksum

In IPv4 datagrams, the integrity of the header fields is indicated by a 16-bit header checksum field (bytes 11 and 12 of the IPv4 datagram). The COE engine detects an IPv4 datagram when the Ethernet frame's type field is 0x0800 and the IP datagram's version field is 0x4. During calculation, the checksum field of the input frame is ignored and replaced with the calculated value. IPv6 headers do not have a checksum field, so the COE does not modify IPv6 header fields. The result of the IPv4 header checksum calculation is indicated by the IP header error status bit (IHE bit in TDES0) in the transmission status. This status bit is set to 1 whenever the Ethernet type field value and IP header version field value are inconsistent, or when the Ethernet frame does not have enough data (as indicated by the IP header length field).

Therefore, in summary, IP header errors occur under the following conditions:

1. For IPv4 datagrams:
  - ▶ Ethernet type is 0x0800, but the IP header version field is not equal to 0x4.
  - ▶ The IPv4 header length field indicates a value less than 0x5 (20 bytes).
  - ▶ The total frame length is less than the value given by the IPv4 header length field.
2. For IPv6 datagrams:
  - ▶ The Ethernet type is 0x86DD, but the IP header version field is not equal to 0x6
  - ▶ The frame ends before the IPv6 header (40 bytes), or the extension header is fully received (as given by the corresponding header length field in the extension header).

If the Ethernet type field indicates an IPv4 payload, the IPv4 header checksum is inserted even if the COE engine detects an IP header error.

### ● TCP/UDP/ICMP Checksum

The TCP/UDP/ICMP checksum function processes IPv4 or IPv6 headers (including extension headers) and determines whether the encapsulated payload is TCP, UDP, or ICMP.

The COE engine is disabled in the following two cases:

1. Non-TCP, non-UDP or non-ICMP/ICMPv6 payloads.
2. Fragmented IP frames (IPv4 or IPv6), IP frames with safety functions (e.g., authentication headers or encapsulated safety payloads), and IPv6 frames with routing headers.

When the COE engine is enabled, it calculates the checksum for TCP, UDP, or ICMP payloads and inserts it into the corresponding field in the header. It can operate in the following two ways:

1. The TCP, UDP or ICMPv6 pseudo-header is not included in the checksum calculation and is assumed to exist in the checksum field of the input frame. The checksum field is included in the checksum calculation and then replaced with the final calculated checksum.
2. The TCP, UDP, or ICMPv6 pseudo-header is included in the checksum calculation, the checksum field is ignored, and the checksum field is overwritten with the final calculated value.



#### Note

For ICMP-over-IPv4 packets, since no pseudo-header is defined for them, the checksum field in the ICMP packet must always be 0x0000 in both modes. If it is not equal to 0x0000, an incorrect checksum may be inserted into the packet.

The result of this operation is indicated by the payload error status bit (TPCE bit in TDES0) in the transmit descriptor. The TPCE bit is set to 1 when one of the following is detected:

1. In store-and-forward mode, the frame is forwarded to the MAC transmitter, but the end of the frame is written to the Rx FIFO.
2. The packet ends before receiving the number of bytes indicated by the payload length field in the IP header.



#### Note

When the packet length is greater than the indicated payload length, the extra bytes are ignored as padding, and no error is reported. When the first type of error is detected, the TCP, UDP, or ICMP headers are not modified. For the second type of error, the calculated checksum is still inserted into the corresponding header field.

### Transmission Timing

In the RMII interface, each nibble from the MII is sent on the RMII as two bits at a time, with the transmission order of the two bits as shown in the figure below. The lower-order bits (D0 and D1) are sent first, followed by the higher-order bits (D2 and D3).

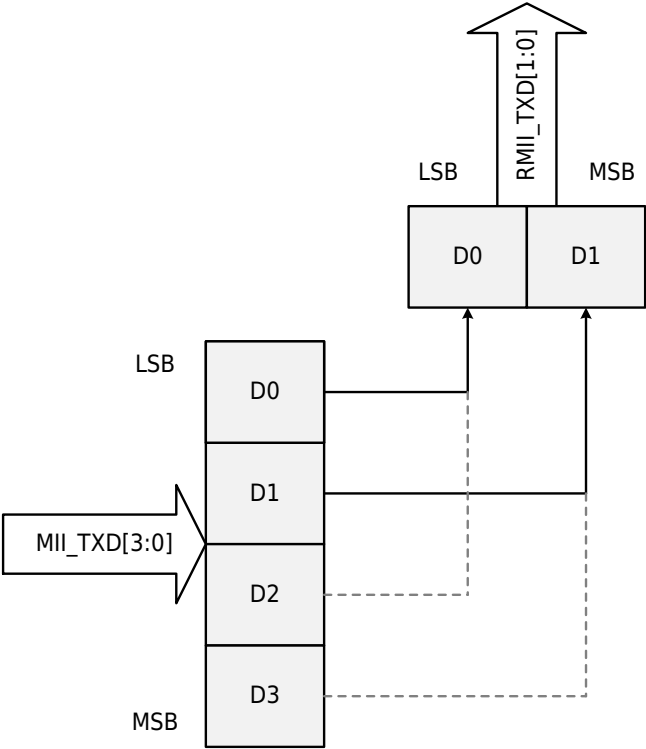


Figure 38-6 MII/RMII Transmission Bit Order

The following shows the basic transmission timing diagram.

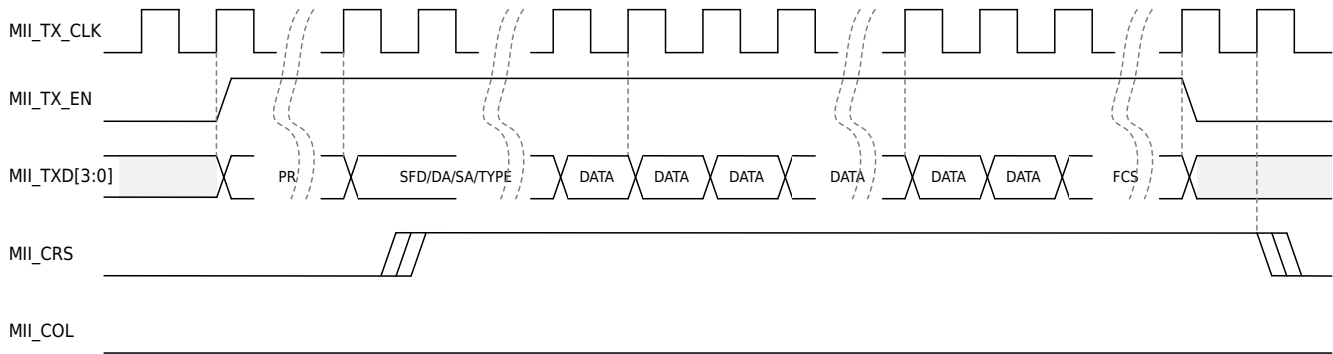


Figure 38-7 Collision-Free Transmission Diagram

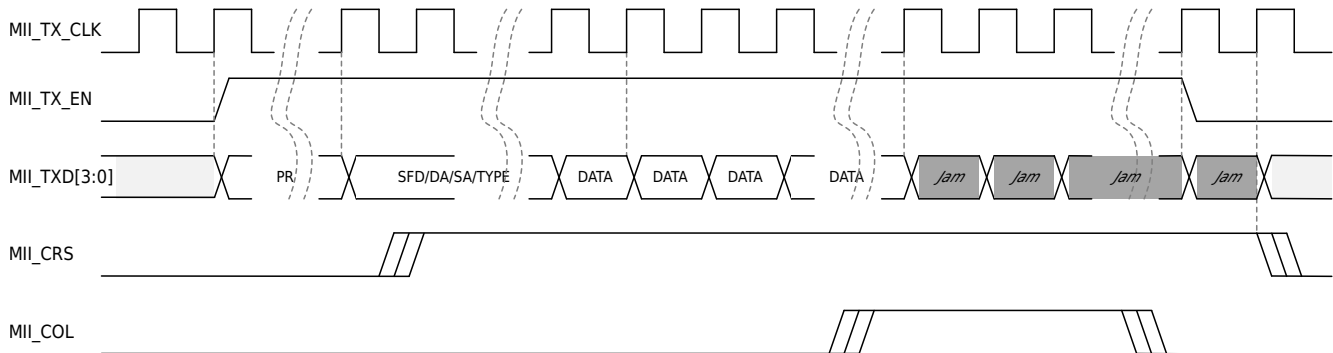


Figure 38-8 Collision Transmission Diagram

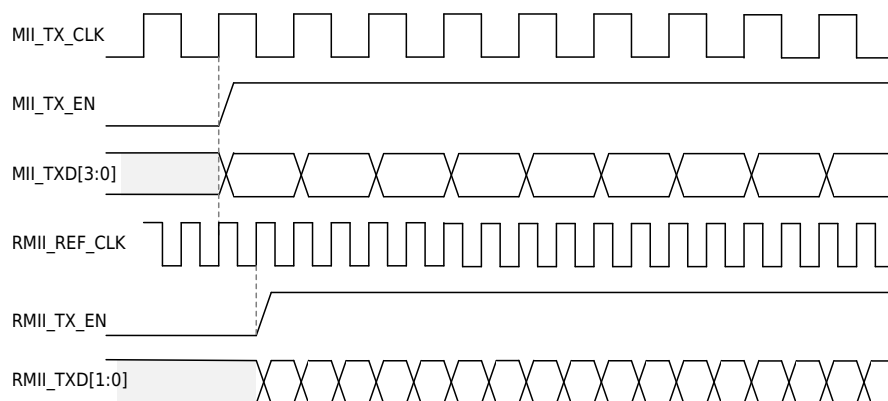


Figure 38-9 Transmission Diagram in MII/RMII Mode

### 38.3.3.4 MAC Frame Reception Action

Frames received by the MAC controller are sent to the Rx FIFO. When reception meets certain conditions, the RxDMA is instructed to perform data transfer, and the RxDMA can initiate preconfigured burst transfers on the AHB interface.

The Ethernet MAC controller is equipped with a 2KB Rx FIFO. RxDMA has two operating modes for transferring data from Rx FIFO to system memory:

In threshold mode, when the number of bytes received by the Rx FIFO exceeds the configured receive threshold (configured using the RTC bit in the ETH\_DMA\_OPRMODR register) or a complete packet is received, data transfer is initiated, and its availability is notified to the RxDMA. After the RxDMA initiates a transfer on the AHB interface, data transfer continues from the Rx FIFO until the entire packet is transmitted. After the transmission of the EOF frame is completed, the receive status word is returned and sent to the RxDMA controller. In this mode, some error frames are not discarded because the error status is received at the end of the frame, by which time the data is already transferred from the Rx FIFO to the system memory.

In store-and-forward mode, frames can only be read after they are completely written to the Rx FIFO. In this mode, all error frames are discarded (if the core is configured to do so), so only active frames are read and forwarded to the application.

The status is immediately available after receive data, so frames can be stored in the Rx FIFO as long as it is not full.

The MAC starts the receive operation when it detects the SFD on the MII. The MAC core removes the header and SFD before further processing the frame. Header fields are checked for filtering, and the FCS field is used to verify the frame's CRC. If the frame does not pass the address filter, it is discarded in the core.

### Basic Protocol Reception

When the MAC receives receive data, it first removes the header and SFD from the received frame. After detecting the SFD, it starts sending Ethernet frame data to the Rx FIFO, beginning with the first byte after the SFD (destination address). If the IEEE1588 timestamp function is enabled, a snapshot of the system time is captured when the SFD of any frame is detected on the MII. Unless the MAC filters out and discards the frame, this timestamp is passed to the application.

If the received frame's length/type field is less than 0x600 and the MAC is programmed with the automatic CRC/PAD removal option, the MAC transmits frame data to the Rx FIFO (no more than the amount specified in the length/type field), then start discarding bytes (including the FCS field). If the

length/type field is greater than or equal to 0x600, the MAC transmits all received Ethernet frame data to the Rx FIFO, regardless of the programmed automatic CRC removal option.

By default, the MAC watchdog timer is enabled, meaning frames exceeding 2048 bytes (DA+SA+LT+data+PAD+FCS) is cut off. This function can be disabled by programming the watchdog disable (MWD) bit in the MAC configuration register (ETH\_MAC\_CONFIGR). However, even if the watchdog timer is disabled, frames larger than 16KB is still cut off, and a watchdog timeout status is given.

### Reception Error Handling

If the Rx FIFO becomes full before receiving EOF data from the MAC, an overflow is declared, and the entire frame is discarded, while the overflow counter in the Frame Loss Statistics Register (ETH\_DMA\_RFR CNTR) increases. Due to the overflow, the status indicates that this is a partial frame. If the corresponding functions are enabled (FEF and FUF bits in ETH\_DMA\_OPRMODR), the Rx FIFO can filter error frames and undersized frames.

In threshold mode, when reading the frame SOF from the Rx FIFO, basic information about the current frame can be obtained. By comparing it with the basic settings of the Rx descriptor, it can be determined whether the frame is an error frame, and the entire error frame can be discarded. In store-and-forward mode, all error frames can be filtered and discarded before data transfer.

### CRC Calculation and PAD Removal

The MAC core calculates the 32-bit CRC of the received frame (including the destination address field to the FCS field) and check for any CRC errors in the received frame. Regardless of whether automatic CRC/PAD removal is enabled (controlled by the ACS bit in the ETH\_MAC\_CONFIGR register), the MAC receives the entire frame to calculate the CRC check of the received frame and feed the result back to the application via the CRE bit in the receive descriptor RDES0.

### Reception Status Word

At the end of Ethernet frame reception, the MAC outputs the receive status, which is sent via the RxDMA to the receive descriptor RDES0 for application reference.

### Pause Frame Reception

During frame transmission, the MAC detects received pause frames and pause frame transmission for the delay specified in the received pause frame (full-duplex mode only). Pause frame detection function can be enabled or disabled via the RFE bit in ETH\_MAC\_FLOCTL R. After receive flow control is enabled, monitoring starts to check whether the destination address of the received frame matches the multicast address of control frames (0x0180C2000001). If a match is detected (the destination address of the received frame matches the reserved control frame's destination address), the MAC decides whether to transmit the received control frame to the application based on the PCF bit in ETH\_MAC\_FLTCTLR.

The MAC also decodes the type, opcode, and pause timer fields of the received control frame. If the status byte counting indicates 64 bytes and there are no CRC errors, the MAC transmitter pauses any data frame transmission for the pause time value decoded multiplied by the slot time (64 byte times for 10/100Mbps modes). At the same time, if another pause frame with a zero pause time value is detected, the MAC resets the pause time and manage the new pause request. If the received control frame does not match the type field (0x8808), opcode (0x00001), and byte length (64 bytes), or if there is a CRC error, the MAC does not generate a pause.

For pause frames with multicast destination addresses, the MAC filters frames based on address matching. For pause frames with unicast destination addresses, the MAC filters based on whether the DA field matches the content of MAC Address Register 0 and whether the UNP bit in ETH\_MAC\_FLOCTL R is set

to 1 (detecting pause frames with unicast destination addresses). The PCF bit in the ETH\_MAC\_FLTCTLR register controls filtering of control frames and address filtering.

### COE Engine

The Ethernet MAC controller supports detection and processing of IPv4 and IPv6 frames in received Ethernet frames to ensure data integrity. The receive COE engine is enabled by setting the IPCO bit in the ETH\_MAC\_CONFIGR register to 1. The MAC identifies IPv4 or IPv6 frames by checking whether the type field of the received Ethernet frame contains the value 0x0800 or 0x86DD. This identification method also applies to frames with VLAN marks.

The receive COE engine can calculate the IPv4 header checksum and check whether it matches the received IPv4 header checksum. If there is any mismatch between the indicated payload type (Ethernet type field) and the IP header version, or if the number of bytes in the received frame is less than the amount indicated in the IPv4 header's length field (fewer than 20 bytes available in the IPv4 or IPv6 header), the IP header error bit is set to 1.

The receive COE engine can also identify TCP, UDP, or ICMP payloads in received IP datagrams (IPv4 or IPv6) and correctly calculate the checksum fields for such payloads, including TCP/UDP/ICMPv6 pseudo-header bytes for checksum calculation, and check whether the received checksum field matches the calculated value. The result of this operation is given by the IP payload error bit in receive descriptor RDES4. If the length of the TCP, UDP, or ICMP payload does not match the expected payload length given in the IP header, this status bit is also set to 1.

The COE engine bypasses payloads of fragmented IP datagrams, IP datagrams with safety functions, IPv6 routing headers, and payloads other than TCP, UDP, or ICMP.

### Reception Timing

In the RMII interface, each nibble inside the MAC is sent from the two bits received from the RMII to the MII, with the nibble transmission order as shown in the figure below. The lower-order bits (D0 and D1) are received first, followed by the higher-order bits (D2 and D3).

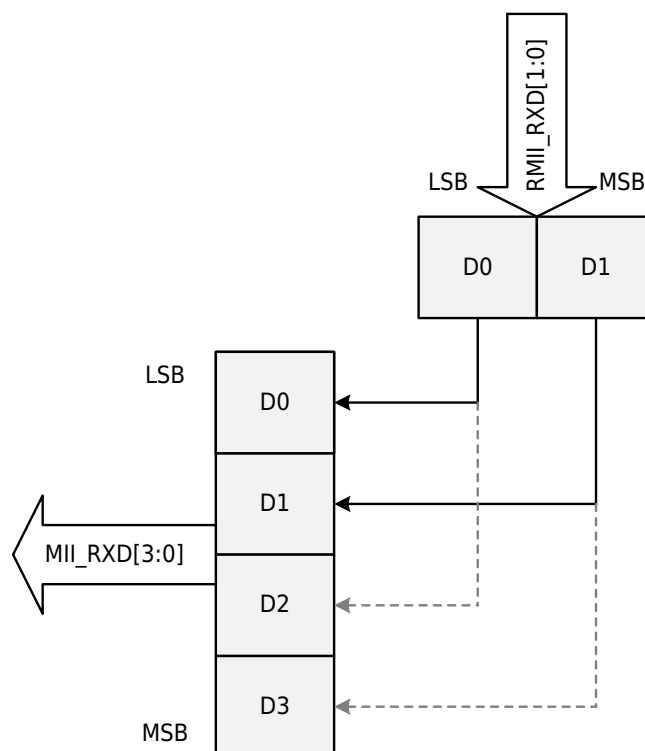


Figure 38-10 MII/RMII Reception Bit Order

The following shows the basic transmission timing diagram.

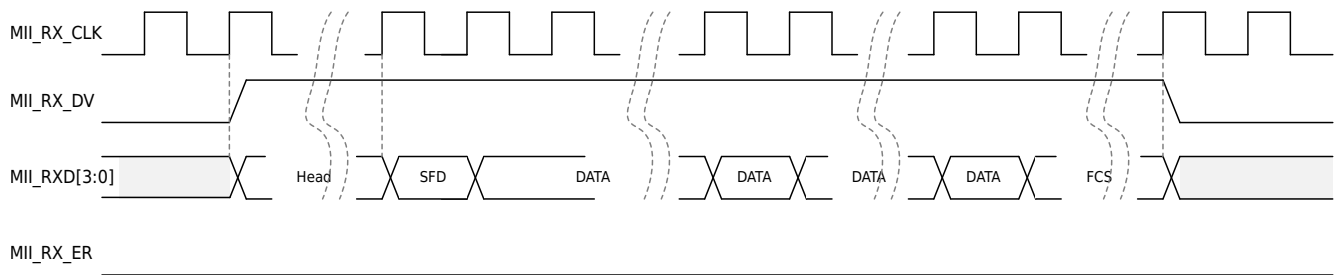


Figure 38-11 Error-Free Transmission Diagram

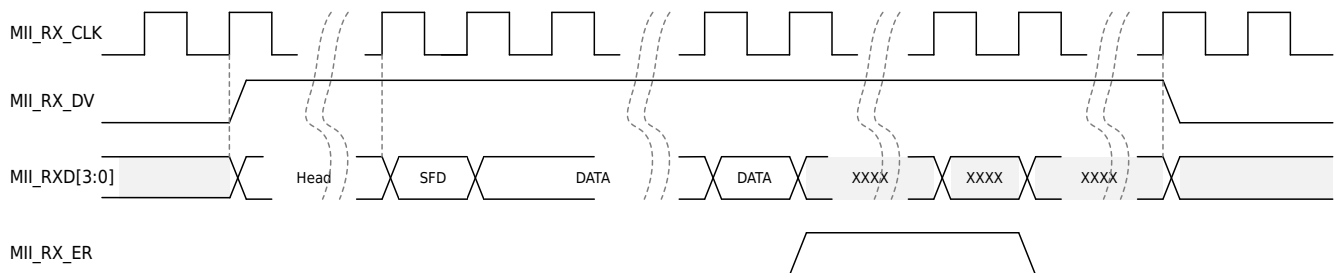


Figure 38-12 Error Transmission Diagram

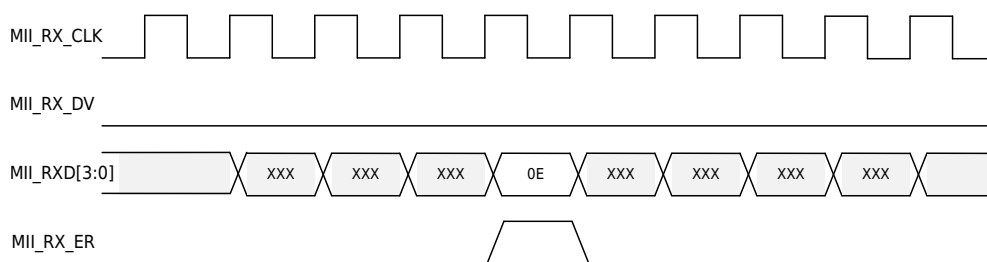


Figure 38-13 Reception Diagram Under False Carrier Indication

### 38.3.3.5 MAC Frame Filtering

The MAC can filter various received frames to select those needed by the application. Frame filtering methods include the following:

1. Filtering by destination address and source address of MAC frames.
2. Filtering by identifier of VLAN frames.
3. Filtering by source address and destination address of LAY3 IP packets.
4. Filtering by source port and destination port of LAY4 TCP and UDP packets.

The basic descriptions of each filtering method are introduced in the following sections:

#### MAC Source Address Filtering

The MAC can perform perfect filtering based on the source address field of the received frame. By default, the MAC compares the SA field with the values programmed in the MAC Address Register. MAC Address Registers 1-4 can be configured to compare the SA field by setting bit 30 (SA) in the MAC Address Register to 1. If the SAF bit in the Frame Filter Control Register (ETH\_MAC\_FLTCTLR) is set to 1, the MAC discards frames that fail SA filtering. Otherwise, the result of SA filtering is given by the status bit in the receive status word (see receive descriptor word RDES0).

Note: When the SAF bit is set to 1, the results of SA filtering and DA filtering are performed AND operations to determine whether to forward the frame. This means that if either filtering fails, the frame is discarded. Both filtering processes must pass for the frame to be forwarded to the application.

When the ETHMAC controller compares each byte of the address value in MAC Address Registers 1-4 with the corresponding SA byte of the received frame, it can mask the byte by setting the corresponding control bit in the register to 1, thereby achieving group address filtering for the SA.

For source address filtering, the filtering match result can be inverted at the final output, meaning that a match in the SA address comparison is considered a filtering failure. This function is controlled by the SAIF bit in the Frame Filter Control Register (ETH\_MAC\_FLTCTLR).

### MAC Destination Address Filtering-Unicast

The MAC supports up to 5 MAC destination addresses for unicast filtering. If perfect filtering is selected (HUC bit in the Frame Filter Control Register set to 0), the MAC compares all 48 bits of the received unicast destination address with the programmed MAC address to determine a match. By default, MAC Address Register 0 is always enabled, while other MAC Address Registers 1-4 are selected via individual enable bits.

When the ETHMAC controller compares each byte of MAC Address Registers 1-4 with the corresponding DA byte of the received frame, it can mask the byte by setting the corresponding control bit in the register, thereby achieving group address filtering for the DA.

### MAC Destination Address Filtering-Multicast

By setting the PMF bit in the Frame Filter Control Register to 1, the MAC is programmed to pass all multicast frames. If the PAM bit is reset, the MAC performs filtering of multicast addresses based on the HMC bit in the Frame Filter Register. In perfect filtering mode, multicast addresses are compared with the programmed MAC Address Registers 1-4. This filtering method also supports group address filtering.

### MAC Destination Address Filtering-Broadcast

In default mode, the MAC does not filter any broadcast frames. However, if the DBF bit in the Frame Filter Register is set to 1 to program the MAC to reject all broadcast frames, any broadcast frame is discarded.

For destination address filtering (unicast, multicast, broadcast), the filtering match result can also be inverted at the final output, meaning that a match in the DA address comparison is considered a filtering failure. This function is controlled by the DAIF bit in the Frame Filter Control Register (ETH\_MAC\_FLTCTLR) and also applies to Hash filtering results.

### MAC Destination Address Hash Filtering

Hash filtering is also supported for unicast and multicast destination address filtering. In Hash filtering mode (HUC bit/HMC bit set to 1), the MAC uses a 64-bit Hash table to perform imperfect filtering of unicast or multicast addresses, i.e., Hash filtering. For Hash filtering, the MAC uses the 6 high-order CRC bits of the received destination address to index the contents of the Hash table. A value of 000000 selects bit 0 in the selected register, and a value of 111111 selects bit 63 in the Hash table register. If the corresponding bit (indicated by the 6-bit CRC) is set to 1, the unicast or multicast frame is considered to have passed Hash filtering; otherwise, the frame is considered to have failed Hash filtering.



#### Note

CRC is a 32-bit value encoded using the following polynomial:

$$G(x)=x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$$



## VLAN Identifier Filtering

When the received frame is a VLAN frame, the MAC can enable filtering of the VLAN identifier field by setting the VTFE bit in the Frame Filter Control Register (ETH\_MAC\_FLTCTLR). The comparison value for filtering is set in the VLFLT bit of the VLAN Tag Acceptance Filter Register (ETH\_MAC\_VTAFLTR), and the VTAL bit in this register can be set to choose whether to filter the entire VLAN identifier field or only the lower 12 bits.

For VLAN identifier field filtering, the filtering match result can also be inverted at the final output, meaning that a match in the identifier field comparison is considered a filtering failure. This function is controlled by the VTIM bit in the VLAN Tag Acceptance Filter Register (ETH\_MAC\_VTAFLTR) and also applies to Hash filtering results.

## VLAN Identifier Hash Filtering

Hash filtering can also be selected for VLAN identifier filtering. In Hash filtering mode (VTHM bit in ETH\_MAC\_VTAFLTR set to 1), the MAC uses a 16-bit Hash table to perform imperfect filtering of VLAN identifiers, i.e., Hash filtering. For Hash filtering, the MAC uses the CRC bits of the VLAN identifier to index the contents of the Hash table. A value of 1 indicates passing VLAN identifier filtering; a value of 0 indicates failing VLAN identifier filtering.



### Note

CRC is a 32-bit value encoded using the following polynomial:

$$G(x) = x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$$

## L3 Source Address & Destination Address Filtering

Filtering by source address and destination address is also supported for received LAY3 IPv4 and IPv6 packets. The L3SAM bit and L3DAM bit in the LAY3LAY4 Control Register can enable source address filtering and destination address filtering, respectively. The comparison values for the source address or destination address are set in the LAY3 Address Registers (L3ADDR0-3). For details, refer to the register description.

LAY3 IP address filtering can optionally mask high-order bits to achieve group address filtering. The high-order bit masking filtering control for source address and destination address is controlled by the HSBM and HDBM bits in the LAY3LAY4 Control Register (ETH\_MAC\_L34CTLR), respectively.

For LAY3 IP packet address filtering, the filtering match result can also be inverted at the final output, meaning that a match in the address field comparison is considered a filtering failure. This function is controlled by the L3SAIM and L3DAIM bits in the LAY3LAY4 Control Register (ETH\_MAC\_L34CTLR).

## L4 Source Port & Destination Port Filtering

Filtering by source port and destination port is also supported for received LAY4 TCP and UDP packets. The L4SPM bit and L4DPM bit in the LAY3LAY4 Control Register can enable source port filtering and destination port filtering, respectively. The comparison values for the source port or destination port are set in the LAY4 Port Register (L4PORTR). For details, refer to the register description.

For LAY4 TCP and UDP packet port filtering, the filtering match result can also be inverted at the final output, meaning that a match in the port field comparison is considered a filtering failure. This function is controlled by the L4SPIM and L3DPIM bits in the LAY3LAY4 Control Register (ETH\_MAC\_L34CTLR).

### 38.3.3.6 MAC Loopback Mode

The MAC supports looping back frames sent to its receiver. By default, the MAC loopback function is disabled. It can be enabled by programming the LM bit in the MAC control register (ETH\_MAC\_CONFIGR).

### 38.3.3.7 MAC Management Counter (MMC)

The MAC Management Counter (MMC) is equipped with a set of Registers to collect statistics about received and transmitted frames. This includes a Control Register (ETH\_MMC\_MMCTLR) to control the behavior of each register, two 32-bit registers (ETH\_MMC\_REVSTSR and ETH\_MMC\_TRSSTSR) containing current receive and transmit statistics, two 32-bit registers (ETH\_MMC\_RITCTLR and ETH\_MMC\_TITCTLR) containing receive and transmit interrupt control, and statistical registers for various frame types. These statistical registers can be accessed from the application. The purpose and significance of these statistical registers are described in detail in the register chapter. Please refer to the ETH\_MMC register chapter.

The MAC management counter (MMC) updates the corresponding statistical registers for received frames that pass address filtering. Statistics for discarded frames are not updated unless the discarded frame is an undersized frame smaller than 6 bytes (DA bytes cannot be fully received).

If a frame is transmitted successfully, it is considered a "Good Frame". That is, a transmitted frame is considered a "Good Frame" if the transmission process is not aborted due to the following errors:

- Jabber timeout
- No carrier/carrier loss
- Late collision
- Frame underrun
- Excessive delay
- Excessive collisions

If a frame is received successfully, it is considered a "Good Frame." That is, a received frame is considered a "Good Frame" if it does not have the following errors:

- CRC error
- Undersized frame (shorter than 64 bytes)
- Alignment error (only for 10/100 Mb/s)
- Length error (non-type frames only)
- Out of range (only for non-type frames, exceeding maximum size)
- MII\_RXER input error



#### Note

The maximum frame size depends on the frame type, as follows:

- Maximum bytes for non-identifier frames: 1518
- Maximum bytes for VLAN frames: 1522

### 38.3.3.8 MAC Power Management (PMT)

The ETHMAC controller supports the Power Management (PMT) mechanism. The PMT module can be enabled via the remote wake-up frame enable bit and the magic packet enable bit. These enable bits (WKEN and MPEN) are located in the ETH\_MAC\_PMTCTLR register and can be programmed by the application. When Power-down mode is enabled (ETH\_MAC\_PMTCTLR.PWDN=1), the MAC discards all received frames and not forward them to the application. The MAC controller exits Power-down mode

only when the corresponding detection bit is enabled and the corresponding magic packet or remote wake-up frame is received.

### Remote Wake-Up Frame Filtering

The ETHMAC controller has 8 wake-up frame filter registers. To perform a write operation on each register, the Remote Wake-Up Frame Filter Register (ETH\_MAC\_RTWKFFR) must be loaded with values one by one. By loading the Remote Wake-Up Frame Filter Register eight times consecutively, the desired values can be loaded into the register. The read operation is similar to the write operation. To read eight values, the Wake-Up Frame Filter Register must be read eight times to reach the last register. Each read/write operation points the Wake-Up Frame Filter Register pointer to the next filter register. The current read/write operation's filter register can be determined by reading the RTWKPT bit in the PMT Control Status Register (ETH\_MAC\_PMTCTLR). The internal pointer can be reset by setting the RTWKFR bit in this register to 1, after which the Remote Wake-Up Frame Filter Register can be operated again. The figure below shows the 8 internal Remote Wake-Up Frame Filter Registers of the ETHMAC.

|                          |                   |                 |                |                 |                |                 |                |                 |
|--------------------------|-------------------|-----------------|----------------|-----------------|----------------|-----------------|----------------|-----------------|
| <i>ETH_MAC_RTWKFFR 0</i> | FLT0<br>Byte Mask |                 |                |                 |                |                 |                |                 |
| <i>ETH_MAC_RTWKFFR 1</i> | FLT1<br>Byte Mask |                 |                |                 |                |                 |                |                 |
| <i>ETH_MAC_RTWKFFR 2</i> | FLT2<br>Byte Mask |                 |                |                 |                |                 |                |                 |
| <i>ETH_MAC_RTWKFFR 3</i> | FLT3<br>Byte Mask |                 |                |                 |                |                 |                |                 |
| <i>ETH_MAC_RTWKFFR 4</i> | -<br>Command      | FLT3<br>Command | -<br>Command   | FLT2<br>Command | -<br>Command   | FLT1<br>Command | -<br>Command   | FLT0<br>Command |
| <i>ETH_MAC_RTWKFFR 5</i> | FLT3<br>Offset    |                 | FLT2<br>Offset |                 | FLT1<br>Offset |                 | FLT0<br>Offset |                 |
| <i>ETH_MAC_RTWKFFR 6</i> | FLT1<br>CRC16     |                 |                |                 | FLT0<br>CRC16  |                 |                |                 |
| <i>ETH_MAC_RTWKFFR 7</i> | FLT3<br>CRC16     |                 |                |                 | FLT2<br>CRC16  |                 |                |                 |

Figure 38-14 Remote Wake-Up Frame Filter Registers

The Function descriptions of each part in the above figure are as follows:

- Filter i Byte Mask

This register defines which bytes of the frame filter i checks to determine whether the frame is a wake-up frame (i=0-3). The MSB (bit 31) must be zero, and bits j[30:0] are the byte mask. If bit j (byte count) of the byte mask is set to 1, the filter i offset + j of the incoming frame is processed by the CRC module; otherwise, filter i offset + j is ignored.

- Filter i Command

This 4-bit command controls filter i operation (i=0-3). Bit3 is the address type selection: when set, only multicast frames are detected; otherwise, only unicast frames are detected. Bit2 and Bit1 are reserved bits. Bit0 is the enable bit for filter i: when set, filter i is enabled; otherwise, it is disabled.

- Filter i Offset

This register defines the offset of the frame to be detected by filter i (i=0-3). This 8-bit pattern offset is the offset of the first byte to be detected by filter i. The minimum value allowed is 12, representing the 13th byte of the frame (offset value 0 represents the first byte of the frame).

- Filter i CRC-16

This register contains the pre-written CRC-16 value (i=0-3) for comparison with the CRC-16 value calculated from the frame data (after filter i Offset and corresponding Byte Mask).

The generator polynomial of CRC-16 is:  $G(x)=x^{16} + x^{15} + x^2 + 1$ .

### Remote Wake-Up Frame Detection

When the MAC is in Power-down mode and the remote wake-up bit WKEN in the PMT Control Status Register (ETH\_MAC\_PMTCTLR) is set to 1, the MAC can resume normal operation after receiving a remote wake-up frame. The PMT supports four programmable filters, which can be configured by the application for different receive frame patterns. A wake-up frame can be received if the incoming frame passes the filter's address filtering and the filter's CRC-16 matches the incoming detection pattern. Only the wake-up frame needs to be checked for length errors, FCS errors, frame tail errors, MII errors, and collisions to ensure it is not an undersized frame. Even if the wake-up frame exceeds 512 bytes, the frame is enable as long as its CRC value is enable.

The wake-up frame detection status bit (WKFR) in the ETH\_MAC\_PMTCTLR register is updated for each received remote wake-up frame. Additionally, a PMT Interrupt is generated (if enabled) to indicate that a remote wake-up frame is received.

### Magic Packet Detection

The magic packet is based on AMD's magic packet technology to power on devices on the network that are in Sleep mode. The MAC receives a specific packet called a magic packet, whose destination address is a node on the network. The MAC controller only checks magic packets with destination addresses matching this device or multicast addresses to determine whether these packets meet the wake-up requirements. Magic packets that pass address filtering (unicast or multicast) are checked to determine whether they conform to the remote wake-up data format: a packet with 6 bytes of all "1" bits plus the MAC address repeated 16 times.

The application enables magic packet wake-up by writing 1 to the MPEN bit in the ETH\_MAC\_PMTCTLR register. The PMT module continuously monitors each frame whose destination address corresponds to the specified magic packet pattern. It checks whether the destination address and source address fields of each received frame are followed by a synchronization stream in the form of 0xFFFFFFFFFFFF. Then, the PMT module checks whether the frame contains the MAC address repeated 16 times without any breaks or interrupts. If there is an interrupt in the 16 repeated addresses, the incoming frame is scanned again for the 0xFFFFFFFFFFFF pattern. These 16 repetitions can be located anywhere in the frame but must follow the synchronization stream (0xFFFFFFFFFFFF). As long as the MAC address repeated 16 times is detected, the device can receive multicast frames. For example, if the node's MAC address is 0x001122334455, the data sequence scanned by the MAC is:

```
Destination Address Source Address... FFFF FFFF FFFF
0011 2233 4455 0011 2233 4455 0011 2233 4455 0011 2233 4455
0011 2233 4455 0011 2233 4455 0011 2233 4455 0011 2233 4455
0011 2233 4455 0011 2233 4455 0011 2233 4455 0011 2233 4455
0011 2233 4455 0011 2233 4455 0011 2233 4455 0011 2233 4455
... CRC
```

Then, it is considered that a magic wake-up packet is received. The magic packet detection in the ETH\_MAC\_PMTCTLR register is updated for received magic packets. Additionally, a PMT interrupt is generated (if enabled) to indicate that a magic packet is received.

### System State During Power-down

1. The MAC receiver should remain enabled during Power-down mode because it needs to perform related reception detection for magic packets/remote wake-up frames. The MAC receiver is enabled by setting the RE bit in the ETH\_MAC\_CONFIGR register to 1.
2. The MAC transmitter should remain off during Power-down mode. The MAC transmitter is turned off by clearing the TE bit in the ETH\_MAC\_CONFIGR register.
3. The Ethernet DMA should be disabled during Power-down mode because there is no need to copy magic packets/network wake-up frames to system memory at this time. The TxDMA and RxDMA are disabled by clearing the STT bit and STR bit in the ETH\_DMA\_OPRMODR register.

The recommended Power-down and wake-up sequence is as follows:

- Step1.** Disable TxDMA and wait for all previous frame transmissions to complete;
- Step2.** Disable the MAC transmitter and MAC receiver by clearing the RE bit and TE bit in the ETH\_MAC\_CONFIGR register;
- Step3.** Wait for the RxDMA to empty all frames in the Rx FIFO;
- Step4.** RxDMA disabled;
- Step5.** Enable magic packet/remote wake-up frame detection by setting the MPEN/WKEN bit in the ETH\_MAC\_PMTCTLR register to 1;
- Step6.** Enable MAC Power-down mode by setting the PWDN bit in the ETH\_MAC\_PMTCTLR register to 1;
- Step7.** Enable the MAC receiver by setting the RE bit in the ETH\_MAC\_CONFIGR register to 1;
- Step8.** Waiting for receiving magic packet and remote wakeup frame;
- Step9.** When a active wake-up frame is received, the Ethernet controller exits Power-down mode;
- Step10.** Read ETH\_MAC\_PMTCTLR to clear the PMT event flag, and enable the MAC transmitter, TxDMA, and RxDMA.

## 38.3.4 ETH\_PTP Function

### 38.3.4.1 ETH\_PTP Function Overview

The IEEE1588 standard defines a protocol that supports precise clock synchronization in measurement and control systems implemented using technologies such as network communication, local computing, and distributed objects. This protocol is suitable for systems that communicate using local area networks (including but not limited to Ethernet) that support multicast messaging. This protocol is used to synchronize non-uniform systems that contain clocks with varying inherent accuracy, resolution, and stability. This protocol supports system-level synchronization accuracy in the sub-second range and requires minimal network and local clock computing resources. This message-based protocol is called the Precision Time Protocol (PTP), and it is transmitted over UDP/IP. Systems or networks are classified as master and slave nodes for distributing timing/clock information. This protocol synchronizes slave nodes with the master node by exchanging PTP messages, as shown in the figure below.

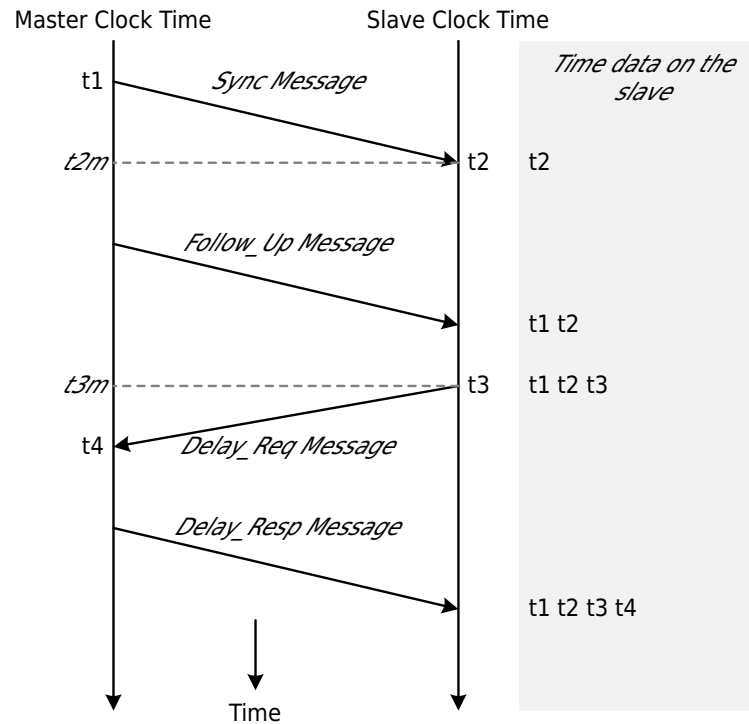


Figure 38-15 Clock Synchronization Diagram

1. The master node broadcasts PTP sync messages to all its nodes. The sync message contains the reference time information of the master node. The time when the message leaves the master node system is  $t_1$ . For Ethernet ports, this time must be captured via the MII interface;
2. The slave node receives the sync message and captures the precise time  $t_2$  using its reference timing;
3. The master node then sends a Follow\_Up message containing  $t_1$  information to the slave node for later use;
4. The slave node sends a Delay\_Req message to the master node, marking the precise time  $t_3$  when the frame leaves the MII;
5. The master node receives this message and captures the precise time  $t_4$  when the message enters its system;
6. The master node sends the  $t_4$  information in a Delay\_Resp message to the slave node;
7. The slave node uses these four values ( $t_1$ ,  $t_2$ ,  $t_3$ , and  $t_4$ ) to synchronize its local reference timing with the master node's reference timing.

Although most protocols are implemented in software above the UDP layer, hardware is required to capture the precise time when specific PTP packets enter or leave the Ethernet port via the MII interface.

#### 38.3.4.2 PTP Reference Timing Source

According to the IEEE1588 specification, to obtain a time snapshot, the core requires a 64-bit format reference time (divided into two 32-bit channels: the high 32 bits represent seconds, and the low 32 bits represent sub-seconds).

The PTP reference clock input is used to internally generate the reference time (also called system time) and capture timestamps. The frequency of this reference clock must be greater than or equal to the resolution of the timestamp counter. The target synchronization accuracy between the master node and slave nodes is approximately 100ns. Accuracy depends on the PTP reference clock input period, the oscillator's features (drift), and the frequency of the synchronization process.

Due to synchronization from the Tx and Rx clock input domains to the PTP reference clock domain, the uncertainty of the timestamp latch value is 1 reference clock period. If the uncertainty caused by resolution is added, it increases by half the timestamp period.

#### 38.3.4.3 PTP Message Types

After the timestamp function is enabled, the TSPMTSEL bit in the Timestamp Control Register (ETH\_PTP\_TSPCTLR) can control which message or type of message to take a timestamp snapshot of. The specific correspondence is shown in the table below.

**Table 38-7 Timestamp Snapshot Target Frames**

| TSPMTSEL[3:0] | Timestamp-Enabled Frame Types                                                          |
|---------------|----------------------------------------------------------------------------------------|
| 0b00X0        | SYNC, Follow_Up, Delay_Req, Delay_Resp                                                 |
| 0b0001        | SYNC                                                                                   |
| 0b0011        | Delay_Req                                                                              |
| 0b01X0        | SYNC, Follow_Up, Delay_Req, Delay_Resp, Pdelay_Req, Pdelay_Resp, Pdelay_Resp_Follow_Up |
| 0b0101        | SYNC, Pdelay_Req, Pdelay_Resp                                                          |
| 0b0111        | Delay_Req, Pdelay_Req, Pdelay_Resp                                                     |
| 0b10XX        | SYNC, Delay_Req                                                                        |
| 0b11XX        | Pdelay_Req, Pdelay_Resp                                                                |

#### 38.3.4.4 PTP Frame Transmission Function

The timestamp is captured when the SFD of the frame is output on the MII. For frames requiring timestamp capture, each transmitted frame can be marked to indicate whether it is necessary to capture its timestamp.

PTP frames can be identified without processing the transmitted frames; frame control can be performed via the control bit in the transmit descriptor.

The captured timestamp is returned to the application in the same way as the frame status is provided. The timestamp is sent back along with the frame's transmission status to the corresponding transmit descriptor, automatically associating the timestamp with the specific PTP frame. The 64-bit timestamp information is written back to the TDES6 and TDES7 fields, with TDES6 holding the 32 least significant bits of the timestamp.

#### 38.3.4.5 PTP Frame Reception Function

When the IEEE1588 timestamp function is enabled, the Ethernet MAC captures timestamps for all frames received on the MII. The MAC provides the timestamp when the frame reception process is completed.

The captured timestamp is returned to the application in the same way as the frame status is provided. The timestamp is sent back along with the frame's receive status to the corresponding receive descriptor. The 64-bit timestamp information is written back to the RDES6 and RDES7 fields, with RDES6 holding the 32 least significant bits of the timestamp.

#### 38.3.4.6 PTP System Time Adjustment

The 64-bit PTP time is updated using the PTP input reference clock. This PTP time can be used as a clock source to capture snapshots (timestamps) of Ethernet frames transmitted or received on the MII. The system time timer can be initialized or calibrated using coarse or fine calibration methods.



In coarse calibration mode, the initial value or offset value is written to the timestamp update register. For initialization, the value in the timestamp update register is written to the system time counter; for system time calibration, the offset value (timestamp update register) is added to or subtracted from the system time.

In fine calibration mode, the offset of the reference clock frequency relative to the master clock (as defined in IEEE1588) is calibrated over a period of time, unlike coarse calibration, which is done within a single clock period. The longer the calibration time, the better it helps maintain linear time and prevents large variations (or jitter) in the reference time between PTP synchronization message intervals. In this method, an accumulator is used to sum the contents of the basic addend register (ETH\_PTP\_TSPADDR), as shown in the figure below. The arithmetic carry generated by the accumulator is used as a pulse to increment the system time counter. Both the accumulator and the basic addend register are 32-bit registers. Here, the accumulator serves as a high-precision frequency multiplier or divider.

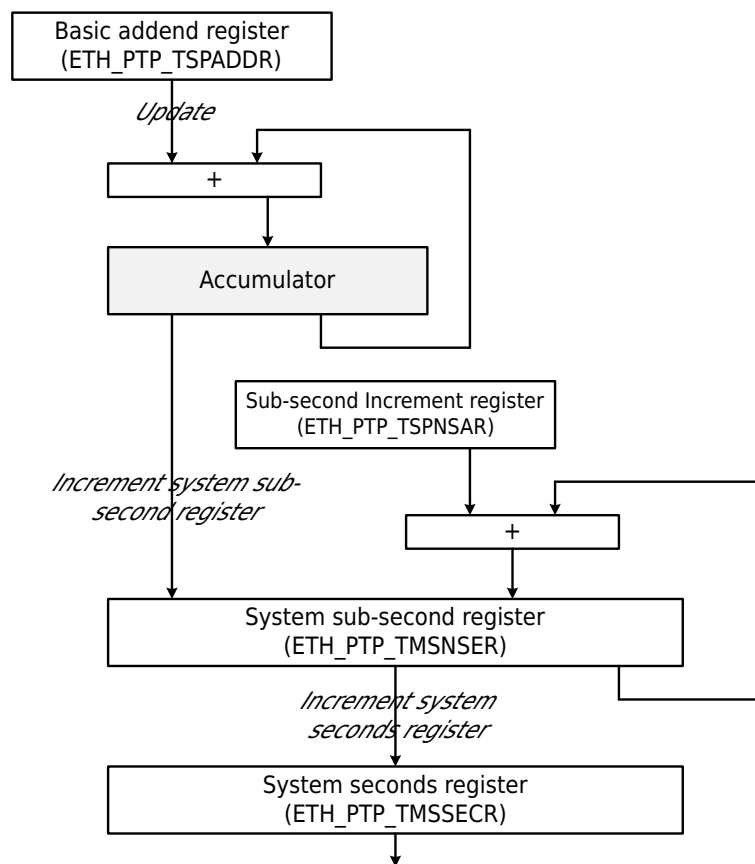


Figure 38-16 System Time Calibration

The system time update logic requires a 50MHz clock frequency to achieve 20ns accuracy. The frequency dividing ratio is the ratio of the reference clock frequency to the desired clock frequency. If the reference clock frequency is 66MHz, the frequency dividing ratio is calculated as  $66\text{MHz}/50\text{MHz}=1.32$ , and the default addend value set in the basic addend register is  $2^{32}/1.32$ , equivalent to 0xC1F07C1F. If the reference clock drifts down to 65MHz, the frequency dividing ratio becomes  $65\text{MHz}/50\text{MHz}=1.3$ , and the value to be set in the basic addend register is  $2^{32}/1.3$ , equivalent to 0xC4EC4EC4. If the clock drifts up to 67MHz, the basic addend register must be set to 0xBF0B7672. When the clock offset is zero, the default addend value to be programmed is 0xC1F07C1F ( $2^{32}/1.32$ ).

The value used to increment the sub-second addend register is set to the decimal number 43. This enables the system time accuracy to reach 20ns (in other words, the incremental step time is 20ns).



The software must calculate the frequency offset based on the sync message and update the basic addend register accordingly. First, the slave clock is set using FreqCompensationValue0 in the basic addend register. The formula for this value is as follows:

$$\text{FreqCompensationValue0} = 2^{32} / \text{FreqDivisionRatio}$$

If it is initially assumed that MasterToSlaveDelay is the same for consecutive sync messages, the following algorithm must be applied. After several synchronization periods, the frequency locks. The slave clock can then determine the precise value of MasterToSlaveDelay and resynchronize with the master clock using the new value.

The algorithm is as follows:

- At MasterSyncTime(n), the master clock sends a sync message to the slave clock. The slave clock receives this message at its local clock time SlaveClockTime(n) and calculates MasterClockTime(n) using the following formula:  

$$\text{MasterClockTime}(n) = \text{MasterSyncTime}(n) + \text{MasterToSlaveDelay}(n)$$
- The master clock counting MasterClockCount(n) for the current sync period is calculated as follows:  

$$\text{MasterClockCount}(n) = \text{MasterClockTime}(n) - \text{MasterClockTime}(n-1)$$
 (Assuming MasterToSlaveDelay is the same for sync periods n and n-1)
- The slave clock counting SlaveClockCount(n) for the current sync period is calculated as follows:  

$$\text{SlaveClockCount}(n) = \text{SlaveClockTime}(n) - \text{SlaveClockTime}(n-1)$$
- The difference in master-slave clock counting SlaveClockCount(n) for the current sync period is calculated as follows:  

$$\text{ClockDiffCount}(n) = \text{MasterClockCount}(n) - \text{SlaveClockCount}(n)$$
- The slave clock's frequency dividing coefficient FreqScaleFactor(n) is calculated as follows:  

$$\text{FreqScaleFactor}(n) = (\text{MasterClockCount}(n) + \text{ClockDiffCount}(n)) / \text{SlaveClockCount}(n)$$
- The frequency compensation value FreqCompensationValue(n) for the addend register is calculated as follows:  

$$\text{FreqCompensationValue}(n) = \text{FreqScaleFactor}(n) \times \text{FreqCompensationValue}(n-1)$$

Theoretically, this algorithm can achieve lock within one sync period, but due to varying network propagation delays and operating conditions, multiple periods may be required.

This algorithm can self-calibrate: if the slave clock initially set by the master clock is incorrect for some reason, the algorithm spends more sync periods calibrating it.

#### 38.3.4.7 PTP System Time Generation Initialization

The timestamp function is enabled by setting the TSPEN bit in the timestamp control register (ETH\_PTP\_TSPCTLR) to 1. However, the timestamp counter must then be initialized to start timestamp operations. The setup sequence is as follows:

- Step1.** Mask the timestamp-triggered interrupt by setting the TSPIM bit in the ETH\_MAC\_INTMSKR register to 1.
- Step2.** Program the TSPEN bit in the ETH\_PTP\_TSPCTLR register to enable the timestamp.
- Step3.** Program the sub-second addend register (ETH\_PTP\_TAPNSAR) based on the PTP clock frequency.
- Step4.** Program the basic addend register (ETH\_PTP\_TSPADDR) and set the TSPADUP bit in the timestamp control register to 1 (basic addend register update).
- Step5.** Poll the timestamp control register to confirm the TSPADUP bit is cleared.
- Step6.** To select fine calibration mode, set the TSPUPSEL bit in the timestamp control register to 1.

- Step7.** Program the timestamp update second register and timestamp update sub-second register with appropriate time values.
- Step8.** Set the TSPINI bit in the timestamp control register to 1 (timestamp initialization).
- Step9.** After initializing the timestamp counter with the value written in the timestamp update register, the timestamp counter begins running.
- Step10.** Enable the MAC receiver and transmitter for the timestamp function to operate normally.

#### 38.3.4.8 PTP PPS Output

The PTP module can output the generated internal clock via PPS (PPS0) to the ETH\_PTP\_PPS port (continuous output mode) for clock synchronization of all nodes in the network system. There are also two PPS outputs (PPS0 and PPS1) (single output mode), which can serve as event outputs to control other blocks.

In PPS continuous output mode, after the system time undergoes second carry (sub-second time Overflow), the PPS0 channel can generate a series of specific pulses or clocks for external network node clock synchronization. The specific output pulse or output clock frequency is set by the PPSFRE0 bit in the PPS output control register (ETH\_PTP\_PPSCTLR). Only the PPS0 channel supports continuous output mode.

In PPS single output mode, if the system time equals the time set in target time register 0 or target time register 1, a pulse can be generated on the PPS0 or PPS1 channel, respectively. This function is only enabled when the TT0SEL bit or TT1SEL bit in the PPS output control register (ETH\_PTP\_PPSCTLR) is set; it is activated when the PPSFRE0 bit or PPSFRE1 bit in the PPS output control register (ETH\_PTP\_PPSCTLR) is set. After the pulse is generated, to generate another pulse output, the target time register 0/1 and the PPSFRE0/1 bits in the PPS output control register must be reset.

For example, users can link the event output of PPS0 or PPS1 in single output mode with Timer2 as needed. For specific implementation, refer to the "General Timer (Timer2) " and "Interrupt Controller (INTC) " chapters. This linkage with Timer2 has two main purposes:

1. When the system time equals the target time, the trigger interrupt provided by the MAC introduces a known delay, causing uncertainty in command execution time. During the period from interrupt trigger to command execution, the timing function of Timer2 can be used to obtain an accurate interrupt processing delay time.
2. System time deviation calibration. PPS outputs a fixed-time pulse, which is detected using Timer2's pulse width measurement function. The PPS system time is adjusted based on Timer2's measurement feedback to calibrate the PTP internal system time.

#### 38.3.5 ETH\_DMA Function

##### 38.3.5.1 ETH\_DMA Function Overview

The ETHMAC controller DMA has independent transmit and receive engines. The transmit engine transfers data from system memory to the Tx FIFO, while the receive engine transfers data from the Rx FIFO to system memory. The DMA can efficiently transfer data between FIFOs and system memory via descriptors without any CPU intervention. The DMA controller can be programmed by the application to generate CPU interrupts upon completion of frame transmission and reception operations or under other normal/error conditions. Therefore, it can be seen that the ETHMAC controller DMA and MCU core communicate via two data structures:

- Control and Status Registers
- Descriptor Lists and Data Buffers

The DMA can transfer data frames received by the MAC to the system's data buffer or send data from the system's data buffer to the MAC. The descriptors contain pointers to these data buffers. The DMA has two descriptor lists: an Rx descriptor list for reception and a Tx descriptor list for transmission. The base addresses of the Rx and Tx descriptor lists are set via the descriptor list registers (ETH\_DMA\_RXDLADR and ETH\_DMA\_TXDLADR), respectively. The descriptor list is a forward-linked list (either implicit or explicit), with the last descriptor pointing back to the first to form a ring structure. The descriptor list resides in the system's physical memory space.

Each descriptor can point to up to two buffers, allowing two physically addressed buffers to replace two contiguous buffers in memory. Explicit linking of descriptors is achieved by configuring the second linked address in the Rx and Tx descriptors (RDES1[14] and TDES0[20]).

The data buffer also resides in the system's physical memory space, typically consisting of an entire frame or part of a frame but never exceeding a single frame. The data buffer contains only data, the status of data communication is stored in the descriptor. When the end of a frame is detected, the DMA jumps to the next frame buffer. Data linking allows frames to span multiple data buffers and can be enabled or disabled. Descriptors have two structure types: ring structure and link structure, as shown in the figure below.

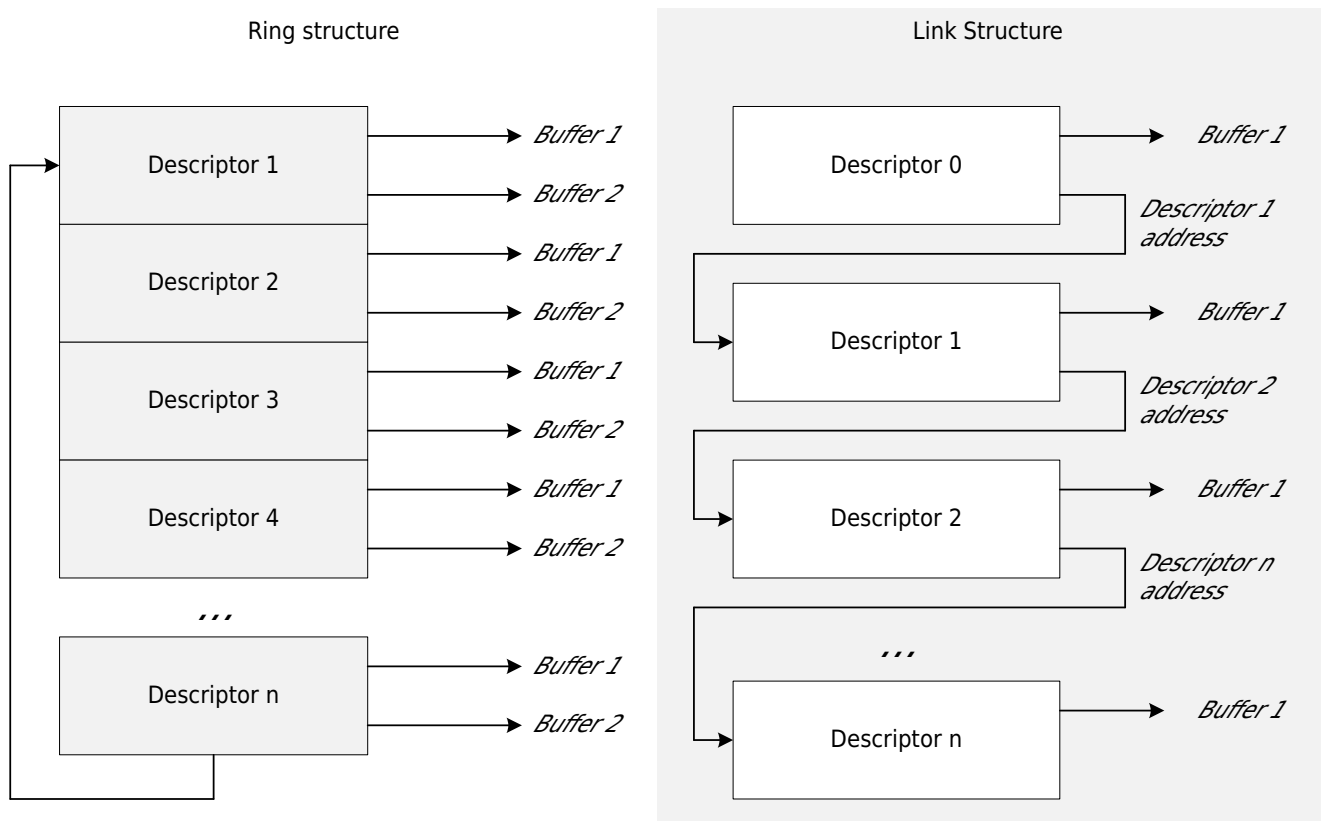


Figure 38-17 Descriptor Structure

### 38.3.5.2 DMA Initialization

The initialization steps for DMA are as follows:

- Step1.** Perform a write operation on ETH\_DMA\_BUSMODR to set bus access parameters;
- Step2.** Perform a write operation on the ETH\_DMA\_INTENAR register to mask unnecessary interrupt sources;
- Step3.** Create descriptor lists and perform write operations on the ETH\_DMA\_RXDLADR and ETH\_DMA\_TXDLADR registers to provide the DMA with the start address of each list;

- Step4.** Perform write operations on the MAC register to set the desired filtering options;
- Step5.** Perform a write operation on the ETH\_MAC\_CONFIGR register to configure and enable transmit and receive operating modes;
- Step6.** Perform a write operation on the ETH\_DMA\_OPRMODR register, setting the STT bit and STR bit to 1 to start transmission and reception;
- Step7.** The transmit and receive engines enter the running state and attempt to fetch descriptors from their respective descriptor lists. These engines then start processing receive and transmit operations. The transmit and receive processes are independent of each other and can be started or stopped separately.

### Bus Burst Access

The DMA attempts to perform fixed-length burst transfers on the AHB Master interface (when the FBST bit in ETH\_DMA\_BUSMODR is configured accordingly). The maximum burst length is indicated and limited by the TPBL or RPBL fields (ETH\_DMA\_BUSMODR [13:8] and [22:17]). For the 16 bytes to be read, the receive and transmit descriptors are always accessed using the maximum possible burst size.

The TxDMA initiates data transfer only when there is enough space in the Tx FIFO to accommodate the configured burst or the number of bytes before the end of the frame (when the frame is shorter than the configured burst length). The TxDMA indicates the start address and the required number of transfers to the AHB master interface. When the AHB interface is configured for fixed-length bursts, the optimal combination of INCR4, INCR8, INCR16, and SINGLE is used to transfer data. Otherwise (non-fixed-length bursts), INCR (undefined length) and SINGLE is used to transfer data.

RxDMA initiates data transfer only when there is sufficient data in the Rx FIFO for the configured burst, or when a frame end is detected in the Rx FIFO (when the frame is shorter than the configured burst length). RxDMA indicates the start address and the number of transfers required to the AHB Master interface. When the AHB interface is configured for fixed-length bursts, the optimal combination of INCR4, INCR8, INCR16, and SINGLE is used to transfer data. If the frame end is reached before the fixed burst on the AHB interface ends, a empty transfer is performed to complete the fixed-length burst transfer. Otherwise (for non-fixed length bursts), INCR (undefined length) and SINGLE is used to transfer data.

When the AHB interface is configured for address-aligned beats (ETH\_DMA\_BUSMODR.AAL=1), the two DMA engines ensures that the first burst transfer initiated by the AHB is less than or equal to the size of the configured PBL. In this way, all subsequent beats starts from an address aligned with the configured PBL. Due to AHB interface transfers exceeding INCR16, the DMA can only align addresses up to a maximum of 16 beats (PBL>16).

### Data Buffer Alignment

There are no restrictions on the start address alignment for transmit and receive data buffers. The start address of the buffer can be aligned with any of the four bytes. However, the DMA always initiates transfers when the address is aligned with the bus width and transmits empty data on unnecessary byte lanes. This typically occurs during the start or end of an Ethernet frame transfer.

- **Buffer Read Operation Example:**

If the transmit buffer address is 0x00000FF2 and 15 bytes need to be transferred, the TxDMA reads five full words from address 0x00000FF0, but discards or ignore the extra bytes (the first two bytes) when transferring the data to the Tx FIFO. Similarly, the last three bytes of the last transfer is also ignored. The TxDMA always ensures that full 32-bit data is transferred to the Tx FIFO, unless it is the end of the frame.

- Buffer Write Operation Example:

If the receive buffer address is 0x0000FF2 and 16 bytes of the received frame need to be transferred, the RxDMA starts writing five full 32-bit data from address 0x0000FF0. However, the first two bytes of the first transfer and the last two bytes of the third transfer contains empty data.

### Buffer Size Calculation

The Ethernet controller DMA does not update the buffer size field in the transmit and receive descriptors, but only updates the status field of the descriptors. The application must use the driver to calculate the buffer size.

The TxDMA transmits the exact number of bytes to the MAC core (indicated by the buffer size field in TDES1). If the descriptor is marked as the first descriptor (by setting the TFS bit in TDES0 to 1), the DMA marks the first transfer of the buffer as the start of the frame. If the descriptor is marked as the last descriptor (by setting the TLS bit in TDES0 to 1), the DMA marks the last transfer of the data buffer as the end of the frame.

The RxDMA continues to transfer data to the buffer until the buffer is full or the end of the frame is received. When the RFS bit of the descriptor is set to 1, if the descriptor is not marked as the last descriptor (the RLS bit in RDES0), the corresponding buffer of the descriptor is filled, and the amount of active data is indicated by the buffer size in RDES1. If the descriptor is marked as the last descriptor, the buffer is not filled, and the amount of active data is represented by the result obtained by subtracting the data buffer pointer offset (the FRAL bit in RDES0) from the buffer size field. When the data buffer pointer is aligned with the data bus width, the offset is zero.



#### Note

Even when the start address of the receive buffer is not aligned with the system data bus width, the system should allocate a receive buffer whose size is aligned with the system bus width. For example, if the system allocates a receive buffer with a start address of 0x1000 and a size of 1024 bytes (1 KB), the software can program the buffer start address in the receive descriptor to have an offset of 0x1002. The RxDMA writes the frame into the buffer, where the first two units (0x1000 and 0x1001) contain empty data. The actual frame starts writing from unit 0x1002. Therefore, although the buffer size is programmed to 1024 bytes, due to the presence of the start offset address, the actual usable space of the buffer is 1022 bytes.

### DMA Arbiter

The arbiter within the Ethernet controller DMA arbitrates between accesses to the AHB Master interface by the transmit and receive channels. Two types of arbitration can be used: round-robin priority and fixed priority. If round-robin priority arbitration (ETH\_DMA\_BUSMODR.DMAA=0) is selected, the arbiter allocates the data bus according to the priority method and ratio set by the ETH\_DMA\_BUSMODR.TXPR and PTAT bits when the transmit and receive DMA simultaneously request access. If fixed priority arbitration (ETH\_DMA\_BUSMODR.DMAA=1) is selected, the arbiter allocates the data bus according to the priority method set by the ETH\_DMA\_BUSMODR.TXPR bit when the transmit and receive DMA simultaneously request access.

#### 38.3.5.3 DMA Error Response

For any data transfer initiated by a DMA channel, if the slave responds with an error, the corresponding DMA stops all operations and update the error bit and fatal bus error bit in the status register (ETH\_DMA\_DMASTR). At this point, the DMA controller can only resume operation after a soft reset or hard reset of the peripheral and reinitialization of the DMA.

### 38.3.5.4 DMA Transmit Configuration

#### TxDMA Operation

During the continuous frame transfer process, the TxDMA processes the second frame differently based on the setting of the OSF bit in the operation mode register (ETH\_DMA\_OPRMODR). When the OSF bit is set to 1, after the transmission of the first frame is completed, the transmit process immediately polls the transmit descriptor list of the second frame. If the second frame is enabled, the transmit process transmits the second frame before writing the status information of the first frame. This type of transmit process, which can obtain two frames simultaneously without closing the status descriptor of the first frame, is called OSF mode. The opposite is non-OSF mode.

The operation sequence of the TxDMA engine when ETH\_DMA\_OPRMODR.OSF=0 is as follows:

- Step1.** After the data buffer is set with the corresponding Ethernet frame data, the user configures the transmit descriptors (TDES0-TDES3) and sets the OWN bit (TDES0[31]) to 1.
- Step2.** After setting the STT bit (ETH\_DMA\_OPRMODR register [13]) to 1, the DMA immediately enters the running state.
- Step3.** In the running state, the DMA polls the transmit descriptor list for the frame to be transferred. After polling starts, the DMA continues in a continuous circular order of descriptors or linked order. If the DMA detects a descriptor marked as owned by the CPU (TDES0.OWN=0) or an error condition occurs, it suspends the transfer and set the transmit buffer unavailable bit (ETH\_DMA\_DMASTR register TUS bit) and the normal interrupt summary enable bit (ETH\_DMA\_DMASTR register AIS bit) to 1. The transmit engine continues to execute step 9.
- Step4.** If the obtained descriptor is marked as owned by the DMA (TDES0.OWN=1), the DMA decodes the transmit data buffer address according to the obtained descriptor.
- Step5.** The DMA fetches the transmit data from the system memory and transfers these data.
- Step6.** If the Ethernet frame is stored in the data buffers of multiple descriptors, the DMA closes the intermediate descriptor and obtain the next descriptor. Repeat steps 3, 4, and 5 until the Ethernet frame data transfer is completed.
- Step7.** After the frame transfer is completed, if the IEEE1588 timestamp is enabled for the frame (as indicated in the transmit status), the timestamp value is written into the transmit descriptors (TDES6 and TDES7), and then the status information is written into the status bits (TDES0) of the transmit descriptor. In this step, the OWN bit is cleared, and the descriptor becomes owned by the CPU.
- Step8.** After the frame transmission is completed, when the last descriptor bit (TDES0.TLS) is set to 1, the TIS bit of the DMA action status register is set to 1. The DMA engine then returns to step 3.
- Step9.** In the suspended state, the DMA attempts to reacquire the descriptor (return to step 3) upon receiving a transmit poll request and clears the underflow interrupt status bit.

The action flowchart of TxDMA in non-OSF mode is shown in the figure below.

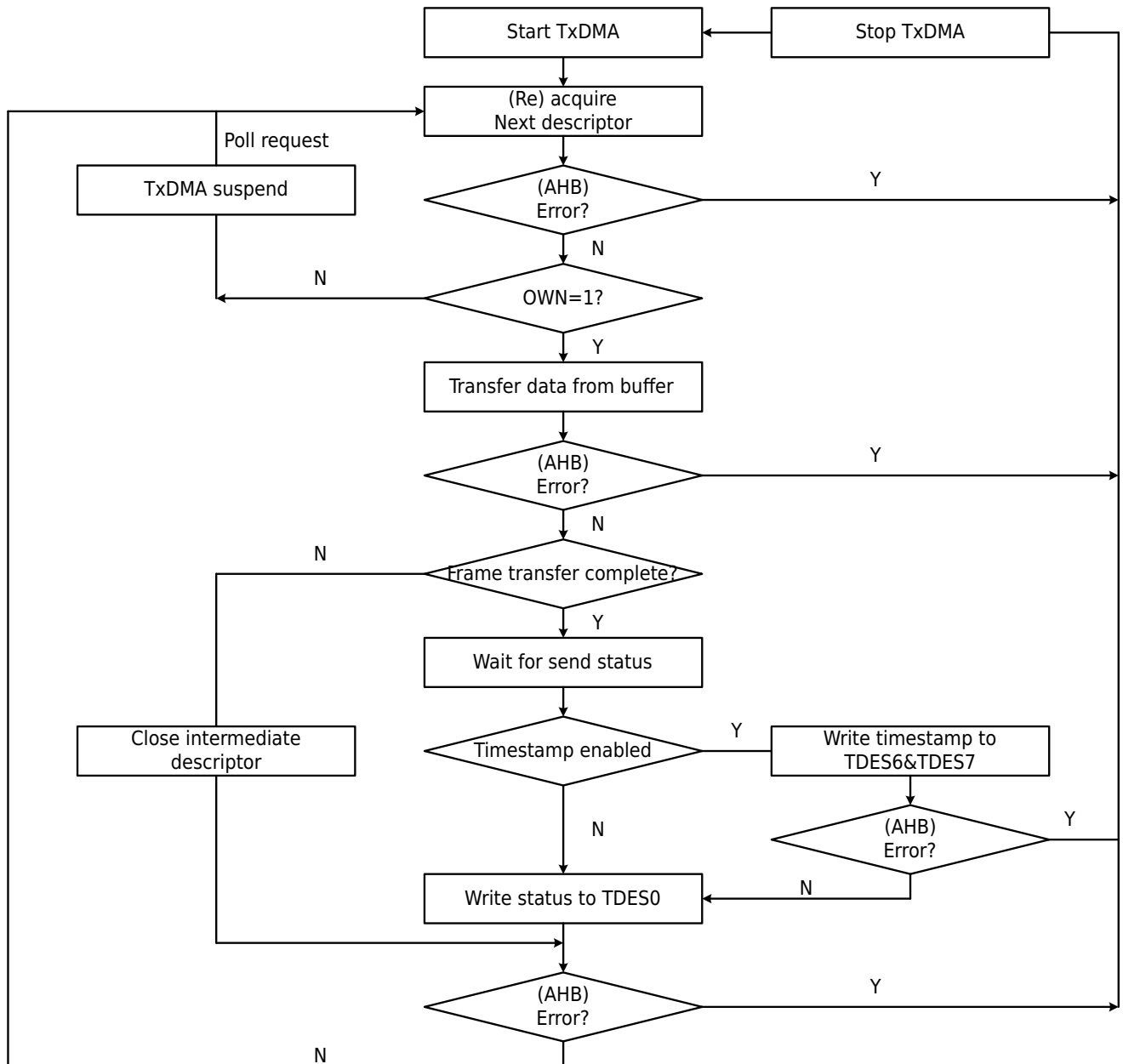


Figure 38-18 TxDMA Workflow (Non-OSF Mode)

The operation sequence of the TxDMA engine when ETH\_DMA\_OPRMODR.OSF=1 is as follows:

- Step1.** The operation process of the DMA is as described in steps 1 to 6 of the TxDMA in non-OSF mode.
- Step2.** Without closing the last descriptor of the previous frame, the DMA can obtain the next descriptor.
- Step3.** If the DMA owns the required descriptor, it decodes the transmit buffer address in the descriptor. If the DMA does not own the descriptor, it enters the suspended mode and jump to step 7.
- Step4.** The DMA fetches the transmit frame from the system memory and sends the frame until the end of the frame data transmission. If the frame is split into multiple descriptors, the intermediate descriptors is closed simultaneously.
- Step5.** The DMA waits for the transmission status and timestamp of the previous frame. When the status is available, if the timestamp (indicated by the status bit) is captured, the DMA writes these timestamps into TDES6 and TDES7. Then, the OWN bit is cleared, the DMA writes the



status into the corresponding TDES0, and then closes the descriptor. If the timestamp is not enabled for the previous frame, the DMA does not change the content of TDES6 and TDES7.

- Step6.** If the timestamp is enabled, the transmit interrupt is set to 1, the DMA obtains the next descriptor, and then continue to execute step 3 (when the status is normal). If the previous transmit status shows an underflow error, the DMA enters the suspended mode (step 7).
- Step7.** In the suspended mode, if the DMA receives the suspended status and timestamp, it writes the timestamp (if the timestamp is enabled for the current frame) into TDES6 and TDES7, and then write the status into the corresponding TDES0. Then, the relevant interrupt is set to 1 and the suspended mode is returned.
- Step8.** Only after receiving the transmit poll request (ETH\_DMA\_TXPOLLR register), the DMA exits the suspended mode and enters the running state (go to step 1 or step 2, depending on the suspended status).

The action flowchart of TxDMA in OSF mode is shown in the figure below.

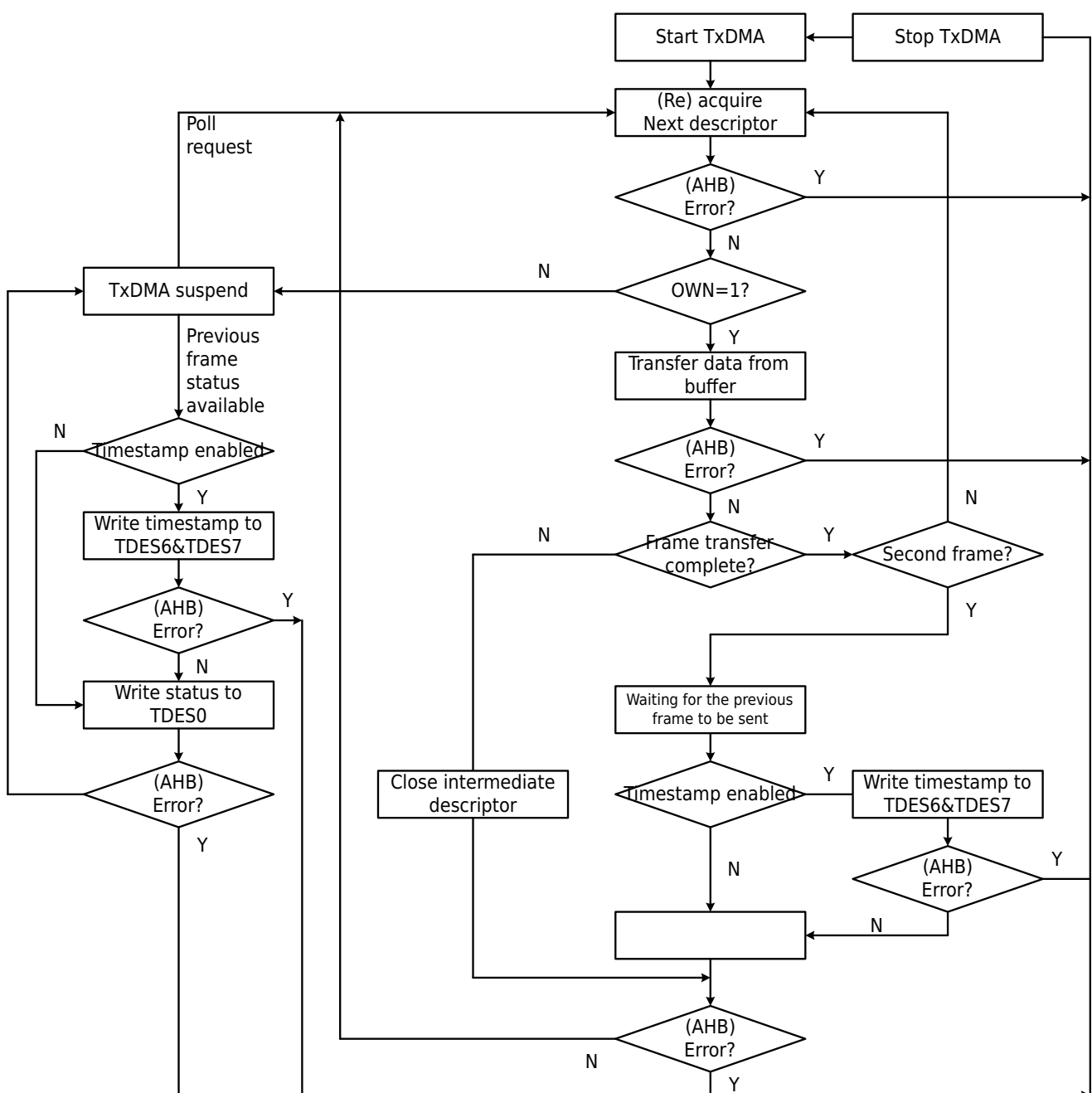


Figure 38-19 TxDMA Workflow (OSF Mode)



## Transmit Frame Handling

The system data buffer must contain the complete Ethernet frame, excluding the header, PAD, and FCS fields, and the DA, SA, and type/length fields must be included in the active data. If the transmit descriptor indicates that the MAC core must prohibit the insertion of CRC or padding bytes, the buffer must have a complete Ethernet frame including the CRC bytes (excluding the header). The frame can be in the form of data linking or span multiple buffers. The frame must be bounded by the first descriptor bit (TDES0.TFS) and the last descriptor bit (TDES0.TLS). When the transmission is initiated, the TDES0.TFS of the first descriptor must be set to 1, and then the frame data is transmitted from the memory buffer to the Tx FIFO. At the same time, if the TDES0.TLS of the current frame is 0, the transmit process attempts to obtain the next descriptor. If TDES0.TLS is 0, it indicates the intermediate buffer, and if TDES0.TLS is 1, it indicates the last buffer of the frame. After the last buffer of the frame is transmitted, the DMA writes the final status information back to the relevant status bit (TDES0) of the descriptor. At this time, if the TDES0.IOC bit is set to 1, the TIS bit of the transmit status register is set to 1, the next descriptor is obtained, and then the process is repeated.

Depending on the setting of the TSF bit in the ETH\_DMA\_OPRMODR register, the actual frame transmission process is initiated when the Tx FIFO reaches the programmable transmit threshold (TTC bit in the ETH\_DMA\_OPRMODR register) or when the Tx FIFO contains a complete frame. After the DMA completes the frame transmission, the descriptor is released (the TDES0.OWN bit is cleared).

## Transmit Poll Suspend

The transmit poll can be suspended by any of the following conditions:

- The DMA detects a descriptor owned by the CPU (TDES0[31]=0), and the transmit buffer unavailable flag TUS in the ETH\_DMA\_DMASTR register is set to 1. To resume, the driver must hand over the ownership of the descriptor to the DMA and then issue a poll query request command.
- When a transmission error caused by underflow is detected, the frame transmission is aborted. The corresponding transmit descriptor (TDES0) bit is set to 1.

If the DMA enters the suspended state due to the first condition, the normal interrupt summary bit and the transmit buffer unavailable bit (NIS and TUS bits in the ETH\_DMA\_DMASTR register) of the transmit status register are both set to 1. If the second condition occurs, the abnormal interrupt summary bit and the transmit underflow bit (AIS and UNS bits in the ETH\_DMA\_DMASTR register) are both be set to 1, and the information is written into the transmit descriptor, causing the suspension. In both cases, the positions in the transmit list is retained. The retained position is the descriptor position following the last descriptor closed by the DMA. After correcting the cause of the suspension, the driver must explicitly issue a transmit poll request command.

## Normal Tx Descriptor

The normal Tx descriptor consists of four 32-bit words, as shown in the table below. These words are defined as TDES0, TDES1, TDES2, and TDES3.

Table 38-8 Normal Tx Descriptor

| Descriptor Structure | Bit Position |     |     |     |      |      |      |      |     |    |     |      |       |    |      |     |       |     |     |      |      |     |      |     |     |   |     |   |   |   |     |     |     |
|----------------------|--------------|-----|-----|-----|------|------|------|------|-----|----|-----|------|-------|----|------|-----|-------|-----|-----|------|------|-----|------|-----|-----|---|-----|---|---|---|-----|-----|-----|
|                      | 31           | 30  | 29  | 28  | 27   | 26   | 25   | 24   | 23  | 22 | 21  | 20   | 19    | 18 | 17   | 16  | 15    | 14  | 13  | 12   | 11   | 10  | 9    | 8   | 7   | 6 | 5   | 4 | 3 | 2 | 1   | 0   |     |
| TDES0                | OWN          | IOC | TLS | TFS | DCRC | DPAD | TTSE | CRCR | CIC |    | TER | TSAC | VLANC |    | TTSS | IHE | ETSUM | JTE | FFF | TPCE | LOCE | NCE | TLCE | ECE | VLF |   | COC |   |   |   | EDE | UDE | DEE |

| Descrip-<br>tor Struc-<br>ture | Bit Position |    |    |      |    |    |    |    |    |    |    |    |    |    |    |    |    |          |    |      |    |    |   |   |   |   |   |   |   |   |   |   |
|--------------------------------|--------------|----|----|------|----|----|----|----|----|----|----|----|----|----|----|----|----|----------|----|------|----|----|---|---|---|---|---|---|---|---|---|---|
|                                | 31           | 30 | 29 | 28   | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14       | 13 | 12   | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| TDES1                          | SAIRC        |    |    | TBS2 |    |    |    |    |    |    |    |    |    |    |    |    |    | Reserved |    | TBS1 |    |    |   |   |   |   |   |   |   |   |   |   |
|                                |              |    |    |      |    |    |    |    |    |    |    |    |    |    |    |    |    |          |    |      |    |    |   |   |   |   |   |   |   |   |   |   |
|                                |              |    |    |      |    |    |    |    |    |    |    |    |    |    |    |    |    |          |    |      |    |    |   |   |   |   |   |   |   |   |   |   |
| TDES2                          | TBAP1        |    |    |      |    |    |    |    |    |    |    |    |    |    |    |    |    |          |    |      |    |    |   |   |   |   |   |   |   |   |   |   |
| TDES3                          | TBAP2        |    |    |      |    |    |    |    |    |    |    |    |    |    |    |    |    |          |    |      |    |    |   |   |   |   |   |   |   |   |   |   |

The specific bits of TDES0-TDES3 are explained below.

● The functions of each bit in TDES0 are as follows:

| Bit | Name | Function                      | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|-----|------|-------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 31  | OWN  | All Related Bits              | <p>► 0b0: The descriptor is owned by the CPU</p> <p>► 0b1: The descriptor is owned by the DMA</p> <p>The DMA clears this bit after completing the frame transmission or after reading all the buffers assigned to the descriptor.</p> <p>After setting all subsequent descriptors of the same frame to 1, the OWN bit of the first descriptor of the frame should also be set to 1.</p>                                                                      |
| 30  | IOC  | End of Transmission Interrupt | <p>When this bit is set to 1, the TIS bit in the action status register is set to 1 after the current frame is transmitted.</p> <div>  <b>Note</b><br/>           This bit is only valid when the last descriptor bit (TDES0.TLS) is set to 1.         </div>                                                                                                               |
| 29  | TLS  | Last Descriptor               | This bit indicates that the descriptor contains the last buffer of the frame.                                                                                                                                                                                                                                                                                                                                                                                |
| 28  | TFS  | First Descriptor              | This bit indicates that the descriptor contains the first buffer of the frame. If the size of the first buffer is 0, the second buffer contains the frame header; if the size of the second buffer is 0, the next descriptor contains the frame header                                                                                                                                                                                                       |
| 27  | DCRC | Invalid CRC                   | <p>When this bit is set to 1, the MAC does not append the cyclic redundancy check (CRC) to the end of the transmitted frame.</p> <div>  <b>Note</b><br/>           This bit is only valid when the first descriptor bit (TDES0.TFS) is set to 1.         </div>                                                                                                           |
| 26  | DPAD | Invalid PAD                   | <p>► 0b0: The DMA automatically adds padding and CRC to frames less than 64 bytes, regardless of the setting of the DCRC bit in the descriptor.</p> <p>► 0b1: The MAC does not automatically add padding to frames less than 64 bytes.</p> <div>  <b>Note</b><br/>           This bit is only valid when the first descriptor bit (TDES0.TFS) is set to 1.         </div> |
| 25  | TTSE | Timestamp enabled             | <p>When this bit is set to 1, the IEEE1588 hardware timestamp function is activated for the transmit frame described by the descriptor.</p> <div>  <b>Note</b><br/>           This bit is only valid when the first descriptor bit (TDES0.TFS) is set to 1.         </div>                                                                                                |

| Bit   | Name  | Function                   | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|-------|-------|----------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 24    | CRCR  | CRC Replacement Control    | <p>When this bit is set to 1, the MAC replaces the FCS field of the transmit frame to be sent with the calculated CRC value.</p> <div>  <b>Note</b><br/>           This bit is only valid when the TDES0.DCRC bit of the descriptor is set to 1.         </div>                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 23:22 | CIC   | Checksum Insertion Control | <p>These bits control the calculation and insertion of the checksum as follows:</p> <ul style="list-style-type: none"> <li>▶ 0b00: Disable checksum insertion</li> <li>▶ 0b01: Enable the calculation and insertion of the IP header checksum only</li> <li>▶ 0b10: Enable the calculation and insertion of the IP header checksum and the TCP/UDP/ICMP checksum, but do not calculate the pseudo-header checksum in hardware</li> <li>▶ 0b11: Enable the calculation and insertion of the IP header checksum and the TCP/UDP/ICMP checksum, and calculate the pseudo-header checksum in hardware</li> </ul> <div>  <b>Note</b><br/>           This bit is only valid when the last descriptor bit (TDES0.TLS) is set to 1.         </div> |
| 21    | TER   | End of Ring Transmission   | This bit indicates that the descriptor list has reached its last descriptor, and the DMA returns to the first address of the descriptor list, forming a descriptor ring.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 20    | TSAC  | Second Address Link        | <p>When this bit is set to 1, the second address in the descriptor is the address of the next descriptor, not the second buffer address, and the TBS2 bit of the descriptor is invalid.</p> <div>  <b>Note</b><br/>           The priority of the TER bit is higher than that of the TSAC bit.         </div>                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 19:18 | VLANC | VLAN Insertion Control     | <ul style="list-style-type: none"> <li>▶ 0b00: No processing for VLAN frames</li> <li>▶ 0b01: Remove the tag and type fields of the VLAN frame before sending</li> <li>▶ 0b10: Insert the VLAN tag value from the VLAN tag transmit control register (ETH_MAC_VTACTLR) into the transmit frame</li> <li>▶ 0b11: Replace the VLAN identifier in the original transmit frame with the VLAN tag value from the VLAN tag transmit control register (ETH_MAC_VTACTLR)</li> </ul>                                                                                                                                                                                                                                                                                                                                                  |

| Bit | Name  | Function                   | Description                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|-----|-------|----------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 17  | TTSS  | Transmit Time-stamp Status | <p>This bit indicates that a timestamp is captured for the current transmit frame, and when set to 1, TDES6 and TDES7 saves the captured timestamp value of the transmit frame.</p> <div>  <b>Note</b><br/>           This bit is only valid when the last descriptor bit (TDES0.TLS) is set to 1.         </div>                                           |
| 16  | IHE   | IP Header Error            | <p>When this bit is set to 1, it indicates that the MAC transmitter has detected an error in the IP datagram header.</p> <div>  <b>Note</b><br/>           When the COE engine detects an IP Header error, it still insert the calculated Checksum value into the IPv4 Header Checksum field.         </div>                                                |
| 15  | ETSUM | Transmit Error Summary     | <p>If one of the bits Bit[16], Bit[14], Bit[13], Bit[12], Bit[11], Bit[10], Bit[9], Bit[8], Bit[2], or Bit[1] of the descriptor is set, this bit is also set.</p>                                                                                                                                                                                                                                                                            |
| 14  | JTE   | Jabber Timeout Error       | <p>When this bit is set to 1, it indicates that the MAC transmitter has experienced a jabber timeout.</p> <div>  <b>Note</b><br/>           This bit is only set to 1 if the MJD bit in the MAC configuration register (ETH_MAC_CONFIGR) is not set to 1.         </div>                                                                                  |
| 13  | FFF   | Frame Flushed              | <p>This bit indicates that the DMA has refreshed the frame according to the software refresh command issued by the CPU.</p>                                                                                                                                                                                                                                                                                                                  |
| 12  | TPCE  | Payload Error              | <p>This bit indicates that the COE engine has detected an error in the payload of the TCP, UDP, or ICMP message, and does not update the Checksum field in the original frame.</p>                                                                                                                                                                                                                                                           |
| 11  | LOCE  | Carrier Loss Error         | <p>This bit indicates that the carrier was lost during frame transmission, that is, the MII_CRS signal was inactive for one or more transmit clock periods during frame transmission.</p> <div>  <b>Note</b><br/>           This bit is only valid for frames that achieve conflict-free transmission when the MAC is in half-duplex mode.         </div> |
| 10  | NCE   | Carrierless Error          | <p>This bit indicates that the carrier sense signal was not triggered by the PHY during transmission.</p>                                                                                                                                                                                                                                                                                                                                    |

| Bit | Name | Function                  | Description                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|-----|------|---------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 9   | TLCE | Late Collision Error      | <p>This bit indicates that the frame transmission process was terminated due to a conflict after the contention window (64 byte times in MII mode, including the header).</p> <div>  <b>Note</b><br/>           If the UDE error bit of the descriptor is set to 1, this bit is invalid.         </div>                                                                  |
| 8   | ECE  | Excessive Collision Error | <p>This bit indicates that the transmission of the current frame was terminated due to 16 consecutive collisions.</p> <div>  <b>Note</b><br/>           If the DRTY (disable retry) bit in the MAC configuration register (ETH_MAC_CONFIGR) is set to 1, this bit is set to 1 after the first collision and the frame transmission process is terminated.         </div> |
| 7   | VLF  | VLAN Frame                | This bit indicates that the transmitted frame is a VLAN frame.                                                                                                                                                                                                                                                                                                                                                                                            |
| 6:3 | COC  | Collision Count           | <p>This bit indicates the number of collisions that occurred before the frame was sent.</p> <div>  <b>Note</b><br/>           When the excessive collision bit (TDES0.ECE) is set to 1, this bit is invalid.         </div>                                                                                                                                             |
| 2   | EDE  | Excessive Delay Error     | <p>This bit indicates that the transmission was terminated due to excessive delay exceeding 24288 bit times.</p> <div>  <b>Note</b><br/>           This bit is valid when the delay check (DC) bit in the MAC configuration register (ETH_MAC_CONFIGR) is set to 1.         </div>                                                                                     |
| 1   | UDE  | Underflow Error           | This bit indicates that the MAC has terminated the frame transmission due to the data in the transmit buffer not arriving in time, and the underflow error indicates that the DMA encountered an empty transmit buffer during frame transmission.                                                                                                                                                                                                         |
| 0   | DEE  | Late Error                | <p>This bit indicates that the MAC has delayed transmission due to the presence of a carrier.</p> <div>  <b>Note</b><br/>           This bit is only valid in half-duplex mode.         </div>                                                                                                                                                                         |

- The functions of each bit in TDES1 are as follows:

| Bit   | Name            | Function                             | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
|-------|-----------------|--------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 31:29 | SAIRC           | SA Insertion and Replacement Control | The SA field of the transmit frame is processed as follows: <ul style="list-style-type: none"> <li>▶ 0b001: Insert the address value from MAC address register 0 as the SA address into the transmit frame</li> <li>▶ 0b101: Insert the address value from MAC address register 1 as the SA address into the transmit frame</li> <li>▶ 0b010: Replace the SA field in the transmit frame with the address value from MAC address register 0 as the SA address</li> <li>▶ 0b110: Replace the SA field in the transmit frame with the address value from MAC address register 1 as the SA address</li> <li>▶ Others: Setting prohibited</li> </ul> |
| 28:16 | TBS2            | Buffer 2 Size field                  | This bit indicates the size of the second data buffer in bytes. <div>  <b>Note</b><br/> This bit is invalid when the TDES0.TSAC bit is set to 1. </div>                                                                                                                                                                                                                                                                                                                                                                                                         |
| 15:13 | <i>Reserved</i> | <i>Reserved</i>                      | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 12:0  | TBS1            | Buffer 1 Size field                  | This bit indicates the size of the first data buffer in bytes. If the field is set to 0, the DMA engine will ignore this buffer and automatically proceed to buffer 2 or the next descriptor, depending on the TSAC configuration.                                                                                                                                                                                                                                                                                                                                                                                                               |

- The functions of each bit in TDES2 are as follows:

| Bit  | Name  | Function                  | Description                                                                                                                                                                                                               |
|------|-------|---------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 31:0 | TBAP1 | Transmit Buffer 1 Address | This bit indicates to the DMA the location of the data in the memory. When the software provides this descriptor (the OWN bit is set to 1 in TDES0) to the DMA, these bits will indicate the physical address of buffer 1 |

- The functions of each bit in TDES3 are as follows:

| Bit  | Name  | Function                                          | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|------|-------|---------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 31:0 | TBAP2 | Transmit Buffer 2 Address/Next Descriptor Address | This bit indicates to the DMA the location of the data in the memory. When the software provides this descriptor (the OWN bit is set to 1 in TDES0) to the DMA and uses the descriptor ring structure, these bits will indicate the physical address of buffer 2. If the TDES0.TSAC bit is set to 1, this address contains a pointer to the physical register of the next descriptor. <div>  <b>Note</b><br/> The buffer address pointer must match the bus width only when the TDES0.TSAC bit is set to 1. </div> |

### Enhanced Tx Descriptor

When the timestamp function (ETH\_PTP\_TSPCTLR.TSPEN=1) or the checksum offload function (ETH\_MAC\_CONFIGR.IPCO=1) is enabled, an enhanced Tx descriptor should be used.

The enhanced Tx descriptor consists of eight 32-bit words, defined as: TDES0, TDES1, TDES2, TDES3, TDES4, TDES5, TDES6, and TDES7. The functions of TDES0 to TDES3 are the same as those of the conventional type.

Table 38-13 Enhanced Tx Descriptor

| Descriptor Structure | Bit Position |     |      |     |      |      |      |      |     |     |      |       |      |     |          |          |      |      |      |     |      |     |     |     |   |   |   |     |     |     |   |   |
|----------------------|--------------|-----|------|-----|------|------|------|------|-----|-----|------|-------|------|-----|----------|----------|------|------|------|-----|------|-----|-----|-----|---|---|---|-----|-----|-----|---|---|
|                      | 31           | 30  | 29   | 28  | 27   | 26   | 25   | 24   | 23  | 22  | 21   | 20    | 19   | 18  | 17       | 16       | 15   | 14   | 13   | 12  | 11   | 10  | 9   | 8   | 7 | 6 | 5 | 4   | 3   | 2   | 1 | 0 |
| TDES0                | OWN          | IOC | TLS  | TFS | DCRC | DPAD | TTSE | CRCR | CIC | TER | TSAC | VLANC | TTSS | IHE | ETSUM    | JTE      | FFF  | TPCE | LOCE | NCE | TLCE | ECE | VLF | COC |   |   |   | EDE | UDE | DEE |   |   |
| TDES1                | SAIRC        |     | TBS2 |     |      |      |      |      |     |     |      |       |      |     | Reserved |          | TBS1 |      |      |     |      |     |     |     |   |   |   |     |     |     |   |   |
| TDES2                |              |     |      |     |      |      |      |      |     |     |      |       |      |     |          | TBAP1    |      |      |      |     |      |     |     |     |   |   |   |     |     |     |   |   |
| TDES3                |              |     |      |     |      |      |      |      |     |     |      |       |      |     |          | TBAP2    |      |      |      |     |      |     |     |     |   |   |   |     |     |     |   |   |
| TDES4                |              |     |      |     |      |      |      |      |     |     |      |       |      |     |          | Reserved |      |      |      |     |      |     |     |     |   |   |   |     |     |     |   |   |
| TDES5                |              |     |      |     |      |      |      |      |     |     |      |       |      |     |          | Reserved |      |      |      |     |      |     |     |     |   |   |   |     |     |     |   |   |
| TDES6                |              |     |      |     |      |      |      |      |     |     |      |       |      |     |          | TTSL     |      |      |      |     |      |     |     |     |   |   |   |     |     |     |   |   |
| TDES7                |              |     |      |     |      |      |      |      |     |     |      |       |      |     |          | TTSH     |      |      |      |     |      |     |     |     |   |   |   |     |     |     |   |   |

The specific bits of TDES0-TDES7 are explained below.

- **Function of each bit of TDES0:** Same as the function of the normal descriptor TDES0.
- **Function of each bit of TDES1:** Same as the function of the normal descriptor TDES1.
- **Function of each bit of TDES2:** Same as the function of the normal descriptor TDES2.
- **Function of each bit of TDES3:** Same as the function of the normal descriptor TDES3.
- **TDES4, TDES5:** Reserved
- **The functions of each bit in TDES6 are as follows:**

| Bit  | Name | Function          | Description                                                                                                                                                                                                                                                                                                                                                                                                                         |
|------|------|-------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 31:0 | TTSL | Timestamp Low Bit | <p>The DMA updates this field with the lower 32 bits of the timestamp captured by the corresponding transmit frame.</p> <div>  <b>Note</b><br/>           This field contains the timestamp only when the last descriptor bit (TDES0.TLS) and the transmit timestamp status bit (TDES0.TTSS) in the descriptor are both set to 1.         </div> |



- The functions of each bit in TDES7 are as follows:

| Bit  | Name | Function           | Description                                                                                                                                                                                                                                                                                                                                                                                                                       |
|------|------|--------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 31:0 | TTSH | Timestamp High Bit | <p>The DMA updates this field with the upper 32 bits of the timestamp captured by the corresponding transmit frame.</p> <div>  <b>Note</b><br/>           This field contains the timestamp only when the last descriptor bit (TDES0.TLS) and the transmit timestamp status bit (TDES0.TTSS) in the descriptor are both set to 1.         </div> |

### 38.3.5.5 DMA Receive Configuration

#### RxDMA Operation

The operation sequence of the RxDMA engine is as follows:

- Step1.** The user sets the receive descriptors (RDES0-RDES3) and sets the OWN bit (RDES0[31]) to 1.
- Step2.** After setting the ETH\_DMA\_OPRMODR.STR bit in the operation mode register to 1, the DMA enters the running state. In the running state, the DMA polls the receive descriptor list to attempt to obtain an idle descriptor. If the obtained descriptor is not idle (owned by the CPU), the DMA enters the suspended state and jumps to step 9.
- Step3.** The DMA decodes the address of the receive data buffer in the obtained descriptor.
- Step4.** Process the incoming frame and place it into the data buffer of the obtained descriptor.
- Step5.** When the buffer is full or the frame transmission is complete, the receive engine obtains the next descriptor.
- Step6.** If the current frame transmission is complete, the DMA continues to execute step 7. If the DMA does not own the next receive descriptor and the frame transmission is not complete (the EOF field has not been transmitted), the DMA sets the descriptor error bit DPE in RDES0 to 1 (unless refresh is disabled). The DMA closes the current descriptor (clearing the OWN bit) and marks it as an intermediate descriptor by clearing the last descriptor bit (RDES0.RLS) (if refresh is disabled, it marks it as the last descriptor), then continues to execute step 8. If the DMA owns the next descriptor but the current frame transmission is not complete, the DMA closes the current descriptor as an intermediate value and returns to step 4.
- Step7.** If the IEEE1588 timestamp function is enabled, the DMA writes the timestamp (if available) into RDES6 and RDES7 of the current descriptor. The DMA then obtains the status of the received frame and writes this status word into RDES0 of the current descriptor, while clearing the OWN bit and setting the last descriptor bit to 1.
- Step8.** The receive engine checks the OWN bit of the latest descriptor. If the CPU owns the descriptor (OWN bit is 0), the receive buffer unavailable bit (ETH\_DMA\_DMASTR.RUS) is set to 1 and the DMA receive engine enters the suspended state (step 9). If the DMA owns the descriptor, the engine returns to step 4 and waits for the next frame.
- Step9.** Before the receive engine enters the suspended state, some frames is refreshed from the Rx FIFO (refresh can be controlled by the DFRF bit in the ETH\_DMA\_OPRMODR register).
- Step10.** When a receive poll request command is received or the start of the next frame can be obtained from the Rx FIFO, the RxDMA exits the suspended state. The engine continues to execute step 2 and reacquires the next descriptor.

The action flowchart of RxDMA is shown in the figure below.

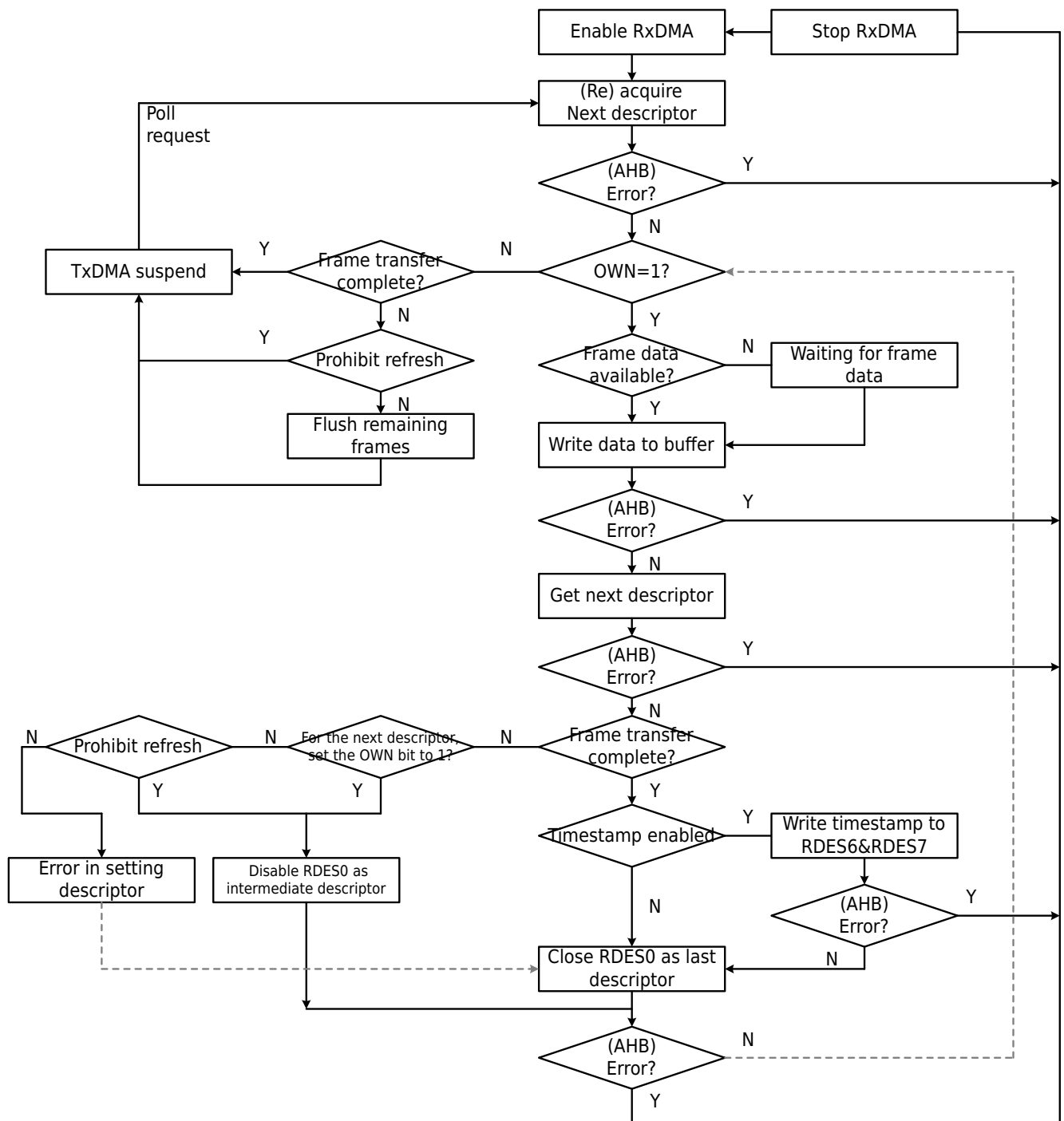


Figure 38-20 RxDMA Workflow

The receive status of RxDMA is only confirmed after the timestamp is written back and the status is ready to be written back to the descriptor. If the timestamp function is enabled, when the active timestamp value of the frame cannot be obtained (for example, the Rx FIFO is full before writing the timestamp), the DMA writes 1 to all bits of RDES6 and RDES7. If the timestamp function is not enabled, RDES6 and RDES7 remain unchanged.

### Receive Frame Acquisition

The receive engine always tries to obtain an additional descriptor for the incoming frame. As long as any of the following conditions are met, the descriptor is attempted to be obtained:

- After the DMA enters the running state, the receive start/stop bit (ETH\_DMA\_OPRMODR.STR) is immediately set to 1.
- Before the current frame transmission ends, the data buffer of the current descriptor is full.

- The MAC controller has completed the reception of the data frame, but the current receive descriptor has not been closed.
- The receive process is suspended because the descriptor is owned by the CPU (RDES0.OWM=0), and a new frame is received.
- A receive poll request command is issued.

### Receive Frame Processing

When the frame passes address filtering and its size is greater than or equal to the configurable threshold number of bytes set for the Rx FIFO, or when the entire frame is written to the Rx FIFO in store-and-forward mode, the MAC transfers the received frame to the system memory. If the frame does not pass address filtering, it is discarded (unless the receive all bit ETH\_MAC\_FLTCTLR.RA is set to 1). Frames less than 64 bytes may be cleared from the Rx FIFO due to conflicts or premature termination. After receiving 64 (configurable threshold) bytes, the DMA begins to transfer the frame data to the receive buffer specified by the current descriptor. After the DMA AHB interface is ready to receive data transfer, if the DMA has not currently obtained transmit data from memory, the first descriptor bit (RDES0.RFS) is set to 1 to separate the frame. When the data buffer is full or the end of the frame is transmitted to the receive buffer, the OWN (RDES0) bit is reset to 0, and the descriptor is released at this time. If the frame is contained in a single descriptor, both the last descriptor bit (RDES0.RLS) and the first descriptor bit (RDES0.RFS) is set to 1.

When the DMA obtains the descriptor of the next frame, it sets the last descriptor bit (FDES0.RLS) to 1, releases the status bit in the previous frame descriptor, and then sets the receive interrupt bit (ETH\_DMA\_DMASTR register RIS) to 1. This process is repeated until the DMA encounters a descriptor marked as owned by the CPU. In this case, the receive process sets the receive buffer unavailable bit (RUS) in the ETH\_DMA\_DMASTR register to 1, then enters the suspended state, but the position in the receive list is still retained.

### Receive Process Suspend

If a new receive frame arrives while the receive process is in the suspended state, the DMA reacquires the current descriptor in the system memory. If the descriptor is now owned by the DMA, the receive process re-enters the running state and begin to receive the frame. If the descriptor is still owned by the master, by default, the DMA discards the current frame at the top of the Rx FIFO and increment the lost frame counter. If the Rx FIFO stores multiple frames, this process is repeated.

Setting the DFRF bit in the DMA operating mode register (ETH\_DMA\_OPRMODR) to 1 can prevent discarding or refreshing the frame at the top of the Rx FIFO. In this case, the receive process sets the receive buffer unavailable bit (RUS) in the action status register (ETH\_DMA\_DMASTR) to 1 and returns to the suspended state.

## Normal Rx Descriptor

The structure of the normal Rx descriptor consists of four 32-bit words, as shown in the table below, defined as RDES0, RDES1, RDES2, and RDES3 respectively.

Table 38-16 Normal Rx Descriptor

| Descriptor Structure | Bit Position |          |      |    |    |    |    |    |    |    |    |    |    |    |    |    |       |      |          |      |     |      |     |     |             |      |      |     |     |     |     |         |
|----------------------|--------------|----------|------|----|----|----|----|----|----|----|----|----|----|----|----|----|-------|------|----------|------|-----|------|-----|-----|-------------|------|------|-----|-----|-----|-----|---------|
|                      | 31           | 30       | 29   | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15    | 14   | 13       | 12   | 11  | 10   | 9   | 8   | 7           | 6    | 5    | 4   | 3   | 2   | 1   | 0       |
| RDES0                | OWN          | DAF      | FRAL |    |    |    |    |    |    |    |    |    |    |    |    |    | ERSUM | DPE  | SAF      | LEE  | OVE | VLAT | RFS | RLS | IPE/TSPA/GF | RLCE | FRAT | WTE | REE | DBE | CRE | DAS/ESA |
| RDES1                | DIC          | Reserved | RBS2 |    |    |    |    |    |    |    |    |    |    |    |    |    | RER   | RSAC | Reserved | RBS1 |     |      |     |     |             |      |      |     |     |     |     |         |
| RDES2                | RBAP1        |          |      |    |    |    |    |    |    |    |    |    |    |    |    |    |       |      |          |      |     |      |     |     |             |      |      |     |     |     |     |         |
| RDES3                | RBAP2        |          |      |    |    |    |    |    |    |    |    |    |    |    |    |    |       |      |          |      |     |      |     |     |             |      |      |     |     |     |     |         |

The specific bits of RDES0 to RDES3 are explained below respectively.

● The functions of each bit in RDES0 are as follows:

| Bit   | Name  | Function                              | Description                                                                                                                                                                                                                                                                                                                                                                                                                                             |
|-------|-------|---------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 31    | OWN   | All related bits                      | <ul style="list-style-type: none"> <li>0b0: The descriptor is owned by the CPU</li> <li>0b1: The descriptor is owned by the DMA</li> </ul> The DMA clears this bit when the frame reception is complete or the associated buffer of this descriptor is full.                                                                                                                                                                                            |
| 30    | DAF   | Destination Address Filtering Failure | This bit indicates that the frame failed the DA filtering in the MAC core.                                                                                                                                                                                                                                                                                                                                                                              |
| 29:16 | FRAL  | Frame Length Statistic                | This bit indicates the byte length of the received frame transmitted to the host memory (including CRC).<br>This field is only valid when the last descriptor bit (RDES0.RLS) is set to 1, and the descriptor error bit (RDES0.DPE) and overflow error bit (RDES0.OVE) are reset.<br>When neither the last descriptor bit nor the error summary bit is set to 1, this field indicates the cumulative number of bytes transmitted for the current frame. |
| 15    | ERSUM | Receive Error Summary                 | If one of the bits Bit[14], Bit[11], Bit[7], Bit[6], Bit[4], Bit[3], Bit[1] of this descriptor and the bits Bit[4] and Bit[3] of the REDS4 descriptor is set, this bit is also set.                                                                                                                                                                                                                                                                     |

| Bit | Name        | Function                         | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|-----|-------------|----------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 14  | DPE         | Descriptor Error                 | <p>This bit indicates that a frame is truncated due to exceeding the size of the current descriptor buffer and that the DMA does not own the next descriptor.</p> <div>  <b>Note</b><br/>           This bit is only valid when the last descriptor (RDES0.RLS) is set to 1.         </div>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| 13  | SAF         | Source Address Filtering Failure | This bit indicates that the SA field of the frame failed the SA filtering in the MAC core.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| 12  | LEE         | Length Error                     | <p>This bit indicates that the actual length of the received frame does not match the value of the length/type field.</p> <div>  <b>Note</b> <ul style="list-style-type: none"> <li>▶ This bit is only valid after the frame type bit (RDES0.FRAT) is reset.</li> <li>▶ This bit is invalid when the CRC error bit (RDES0.CRE) is enable.</li> </ul> </div>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| 11  | OVE         | Overflow Error                   | This bit indicates that the received frame is corrupted due to buffer overflow.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| 10  | VLAT        | VLAN Identifier                  | This bit indicates that the frame pointed to by the descriptor is a VLAN frame marked by the MAC core.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| 9   | RFS         | First Descriptor                 | This bit indicates that the descriptor contains the first buffer of the frame. If the size of the first buffer is 0, the second buffer contains the frame header; if the size of the second buffer is 0, the next descriptor contains the frame header                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| 8   | RLS         | Last Descriptor                  | This bit indicates that the buffer pointed to by this descriptor is the last buffer of the frame.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 7   | IPE/TSPA/GF | COE Error/Time-stamp/Jumbo Frame | <p><b>When the Checksum Offload function is enabled:</b></p> <ul style="list-style-type: none"> <li>▶ This bit indicates that there is an error in the IPv4 or IPv6 header.</li> </ul> <p>The possible causes of this error are:</p> <ol style="list-style-type: none"> <li>1. The Ethernet type field is inconsistent with the IP header version field value.</li> <li>2. The checksum of the header in IPv4 does not match.</li> <li>3. The Ethernet frame lacks the required number of IP header bytes.</li> </ol> <p><b>When the timestamp function is enabled:</b></p> <ul style="list-style-type: none"> <li>▶ This bit indicates that a timestamp snapshot is recorded in RDES6 and RDES7.</li> </ul> <div>  <b>Note</b><br/>           This bit is only valid when the last descriptor (RDES0.RLS) is set to 1.         </div> <p><b>When neither of the above is enable:</b></p> |

| Bit | Name        | Function                                       | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|-----|-------------|------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 7   | IPE/TSPA/GF | COE Error/Time-stamp/Jumbo Frame               | ► This bit indicates that the MAC has received a jumbo frame.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 6   | RLCE        | Late Collision Error                           | This bit indicates that a delayed collision occurred while receiving a frame in half-duplex mode.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| 5   | FRAT        | Frame Type                                     | ► 0b0: The MAC receives an Ethernet frame.<br>► 0b1: The MAC receives a PTP frame.<br><div>  <b>Note</b><br/>             This bit is invalid when a runt frame is received.           </div>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| 4   | WTE         | Watchdog Error                                 | This bit indicates that the receive watchdog timer has timed out while receiving the current frame, and the current frame is truncated after the watchdog timeout.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 3   | REE         | Receive Error                                  | This bit indicates that during frame reception, the PHY issued an RX_ER signal when it issued an RX_DV signal.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| 2   | DBE         | Dribble Bit Error                              | This bit indicates that the received frame has a non-integer multiple of bytes (odd half-byte).<br><div>  <b>Note</b><br/>             This bit is only valid in MII mode.           </div>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 1   | CRE         | CRC Error                                      | This bit indicates that the received frame has a cyclic redundancy check (CRC) error.<br><div>  <b>Note</b><br/>             This bit is only valid when the last descriptor (RDES0.RLS) is set to 1.           </div>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| 0   | DAS/ESA     | Address Filtering Success/Status bit Expansion | <b>When the COE function and the PTP function are invalid:</b> <ul style="list-style-type: none"> <li>► 0b0: The received frame passed the DA address filtering of MAC address register 0.</li> <li>► 0b1: The received frame passed the DA address filtering of MAC address registers 1 to 5.</li> </ul> <div>  <b>Note</b><br/>             When the DAF bit of the descriptor is set, this bit is invalid.           </div> <b>When COE or PTP is enabled:</b> <ul style="list-style-type: none"> <li>► This bit indicates that the RDES4 descriptor is enable.</li> </ul> <div>  <b>Note</b><br/>             This bit is only valid when the last descriptor (RDES0.RLS) is set to 1.           </div> |

- The functions of each bit in RDES1 are as follows:

| Bit   | Name     | Function                        | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
|-------|----------|---------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 31    | DIC      | Disable Interrupt on Completion | <p>When this bit is set to 1, it prohibits the host from generating an interrupt when the received frame ends in the buffer indicated by the descriptor, and it also prevents the RIS bit in the action status register (ETH_DMA_DMASTR) from being set to 1.</p> <div>  <b>Note</b><br/>           This bit is only valid when the last descriptor bit (RDES0.RLS) is set to 1.         </div> |
| 30:29 | Reserved | Reserved                        | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 28:16 | RBS2     | Buffer 2 Size Field             | <p>This bit indicates the size of the second data buffer in bytes.</p> <div>  <b>Note</b><br/>           This bit is invalid when the RSAC bit of the descriptor is set to 1.         </div>                                                                                                                                                                                                    |
| 15    | RER      | End of Ring Reception           | This bit indicates that the descriptor list has reached its last descriptor, and the DMA returns to the first address of the descriptor list, forming a descriptor ring.                                                                                                                                                                                                                                                                                                         |
| 14    | RSAC     | Second Address Link             | <p>When this bit is set to 1, the second address in the descriptor is the address of the next descriptor, not the second buffer address, and the RBS2 bit of the descriptor is invalid.</p> <div>  <b>Note</b><br/>           The priority of the RER bit is higher than that of the RSAC bit.         </div>                                                                                 |
| 13    | Reserved | Reserved                        | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 12:0  | RBS1     | Buffer 1 Size Field             | This bit indicates the size of the first data buffer in bytes. If this field is 0, the DMA ignores this buffer and use buffer 2 or the next descriptor, depending on the setting of RSAC in the descriptor.                                                                                                                                                                                                                                                                      |

● The functions of each bit in RDES2 are as follows:

| Bit  | Name  | Function                 | Description                                                                                                                                                                                                               |
|------|-------|--------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 31:0 | RBAP1 | Receive Buffer 1 Address | This bit indicates to the DMA the location of the data in the memory. When the software provides this descriptor (the OWN bit is set to 1 in RDES0) to the DMA, these bits will indicate the physical address of buffer 1 |

● The functions of each bit in RDES3 are as follows:

| Bit  | Name  | Function                                         | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|------|-------|--------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 31:0 | RBAP2 | Receive Buffer 2 Address/Next Descriptor Address | <p>This bit indicates to the DMA the location of the data in the memory. When the software provides this descriptor (the OWN bit is set to 1 in RDES0) to the DMA and uses the descriptor ring structure, these bits will indicate the physical address of buffer 2. If the RDES1.RSAC bit is set to 1, this address contains a pointer to the physical register of the next descriptor.</p> <div>  <b>Note</b><br/>           The buffer address pointer must match the bus width only when the RDES1.RSAC bit is set to 1.         </div> |

### Enhanced Rx Descriptor

When the timestamp function (ETH\_PTP\_TSPCTLR.TSPEN=1) or the checksum offload function (ETH\_MAC\_CONFIGR.IPCO=1) is disabled, an enhanced Rx descriptor should be used.

The structure of the enhanced Tx descriptor consists of eight 32-bit words, as shown in the table below, defined as RDES0, RDES1, RDES2, RDES3, RDES4, RDES5, RDES6, and RDES7 respectively. The functions of RDES0 to RDES3 are the same as those of the normal type.

Table 38-21 Enhanced Rx Descriptor

| Descriptor Structure | Bit Position |          |      |    |       |    |       |    |          |    |    |    |    |    |    |    |       |      |          |      |      |      |       |     |             |      |      |     |        |     |        |         |      |  |      |  |      |  |      |  |
|----------------------|--------------|----------|------|----|-------|----|-------|----|----------|----|----|----|----|----|----|----|-------|------|----------|------|------|------|-------|-----|-------------|------|------|-----|--------|-----|--------|---------|------|--|------|--|------|--|------|--|
|                      | 31           | 30       | 29   | 28 | 27    | 26 | 25    | 24 | 23       | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15    | 14   | 13       | 12   | 11   | 10   | 9     | 8   | 7           | 6    | 5    | 4   | 3      | 2   | 1      | 0       |      |  |      |  |      |  |      |  |
| RDES0                | OWN          | DAF      | FRAL |    |       |    |       |    |          |    |    |    |    |    |    |    | ERSUM | DPE  | SAF      | LEE  | OVE  | VLAT | RFS   | RLS | IPE/TSPA/GF | RLCE | FRAT | WTE | REE    | DBE | CRE    | DAS/ESA |      |  |      |  |      |  |      |  |
| RDES1                | DIC          | Reserved | RBS2 |    |       |    |       |    |          |    |    |    |    |    |    |    | RER   | RSAC | Reserved | RBS1 |      |      |       |     |             |      |      |     |        |     |        |         |      |  |      |  |      |  |      |  |
| RDES2                | RBAP1        |          |      |    |       |    |       |    |          |    |    |    |    |    |    |    |       |      |          |      |      |      |       |     |             |      |      |     |        |     |        |         |      |  |      |  |      |  |      |  |
| RDES3                | RBAP2        |          |      |    |       |    |       |    |          |    |    |    |    |    |    |    |       |      |          |      |      |      |       |     |             |      |      |     |        |     |        |         |      |  |      |  |      |  |      |  |
| RDES4                | Reserved     |          |      |    | L4FMS |    | L3FMS |    | Reserved |    |    |    |    |    |    |    |       |      | TSPD     |      | PTPV |      | PTPFT |     | MTP         |      |      |     | IPv6DR |     | IPv4DR |         | IPCB |  | IPPE |  | IPHE |  | IPPT |  |
| RDES5                | Reserved     |          |      |    |       |    |       |    |          |    |    |    |    |    |    |    |       |      |          |      |      |      |       |     |             |      |      |     |        |     |        |         |      |  |      |  |      |  |      |  |
| RDES6                | RTSL         |          |      |    |       |    |       |    |          |    |    |    |    |    |    |    |       |      |          |      |      |      |       |     |             |      |      |     |        |     |        |         |      |  |      |  |      |  |      |  |
| RDES7                | RTSH         |          |      |    |       |    |       |    |          |    |    |    |    |    |    |    |       |      |          |      |      |      |       |     |             |      |      |     |        |     |        |         |      |  |      |  |      |  |      |  |



The specific bits of RDES0 to RDES7 are explained below respectively.

- **Function of each bit of RDES0:** Same as the function of the normal descriptor RDES0.
- **Function of each bit of RDES1:** Same as the function of the normal descriptor RDES1.
- **Function of each bit of RDES2:** Same as the function of the normal descriptor RDES2.
- **Function of each bit of RDES3:** Same as the function of the normal descriptor RDES3.
- **The functions of each bit in RDES4 are as follows:**

| Bit   | Name     | Function             | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|-------|----------|----------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 31:26 | Reserved | Reserved             | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| 25    | L4FMS    | L4 Filtering Success | This bit indicates that the received frame has passed the L4 port filtering.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 24    | L3FMS    | L3 Filtering Success | This bit indicates that the received frame has passed the L3 address filtering.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 23:15 | Reserved | Reserved             | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| 14    | TSPD     | Timestamp Discard    | This bit indicates that due to Rx FIFO overflow, although a timestamp snapshot was taken, the timestamp was discarded.                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| 13    | PTPV     | PTP Version          | This bit indicates the version type of the received PTP frame.<br>▶ 0b0: IEEE1588v1 version PTP frame<br>▶ 0b1: IEEE1588v2 version PTP frame                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 12    | PTPFT    | PTP Frame Type       | This bit indicates the type of the received PTP frame:<br>▶ 0b0: Over Ethernet type PTP frame<br>▶ 0b1: Over UDP-IPv4 or Over UDP-IPv6 type PTP frame<br><div>  <b>Note</b><br/> Whether it is Over UDP-IPv4 or Over UDP-IPv6 can be obtained through the IPv6FR and IPv4FR bits of the descriptor. </div>                                                                                                                                                                                               |
| 11:8  | MTP      | Message Type         | This bit indicates the type of the received message:<br>▶ 0b0000: No PTP message received<br>▶ 0b0001: SYNC (all clock types)<br>▶ 0b0010: Follow_Up (all clock types)<br>▶ 0b0011: Delay_Req (all clock types)<br>▶ 0b0100: Delay_Resp (all clock types)<br>▶ 0b0101: Pdelay_Req (in point-to-point transparent clock)<br>▶ 0b0110: Pdelay_Resp (in point-to-point transparent clock)<br>▶ 0b0111: Pdelay_Resp_Follow_Up (in point-to-point transparent clock)<br>▶ 0b1000: Announce<br>▶ 0b1001: Management<br>▶ 0b1010: Signaling<br>▶ 0b1111: Default message type<br>▶ Other: Reserved |

| Bit | Name   | Function                       | Description                                                                                                                                                                                                                                                                                                                                   |
|-----|--------|--------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 7   | IPv6DR | Received IPv6 Report           | This bit indicates that an IPv6 report is received.                                                                                                                                                                                                                                                                                           |
| 6   | IPv4DR | Received IPv4 Report           | This bit indicates that an IPv4 report is received.                                                                                                                                                                                                                                                                                           |
| 5   | IPCB   | Checksum Off-load Pass-Through | This bit indicates bypassing the COE engine.                                                                                                                                                                                                                                                                                                  |
| 4   | IPPE   | IP Payload Error               | The following situations set this bit to 1:<br>1. When the 16-bit IP payload checksum (i.e., TCP, UDP, or ICMP checksum) calculated by the core does not match the corresponding checksum field in the received frame.<br>2. When the length of the TCP, UDP, or ICMP segment does not match the payload length value in the IP header field. |
| 3   | IPHE   | IP Header Error                | This bit indicates that the 16-bit IPv4 header checksum calculated by the core does not match the received checksum byte, or the IP header version field is inconsistent with the Ethernet type value.                                                                                                                                        |
| 2:0 | IPPT   | Payload Type                   | This bit indicates the type of payload encapsulated in the IP datagram, and these bits are "000" when the IP header is in error or there is fragmented IP.<br>► 0b000: Unknown, or unprocessed IP payload<br>► 0b001: UDP<br>► 0b010: TCP<br>► 0b011: ICMP<br>► 0b1xx: Reserved                                                               |

- **RDES5:** Reserved
- The functions of each bit in RDES6 are as follows:

| Bit  | Name | Function          | Description                                                                                                                                                                                                                                                                                                                             |
|------|------|-------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 31:0 | RTSL | Timestamp Low Bit | The DMA updates this field with the lower 32 bits captured by the corresponding received frame.<br><br><div>  <b>Note</b><br/> This field contains the timestamp only when the last descriptor bit (RDES0.RLS) in the descriptor is set to 1. </div> |

- The functions of each bit in RDES7 are as follows:

| Bit  | Name | Function           | Description                                                                                                                                                                                                                                                                                                                             |
|------|------|--------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 31:0 | RTSH | Timestamp High Bit | The DMA updates this field with the upper 32 bits captured by the corresponding received frame.<br><br><div>  <b>Note</b><br/> This field contains the timestamp only when the last descriptor bit (RDES0.RLS) in the descriptor is set to 1. </div> |

### 38.3.6 Interrupt

#### 38.3.6.1 ETHMAC Interrupt Overview

The figure below shows the interrupt implementation scheme in the Ethernet MAC controller. From the figure, it can be seen that ETHMAC summarizes DMA normal events (NIS), DMA abnormal events (AIS), PMT events (PMTIS), MMC events (MMC\*\*IS), PTP events (TSPIS), etc. into a single interrupt event output to the interrupt module through interrupt enable or mask control. Therefore, when the system responds to an interrupt, it needs to read the corresponding status information to determine which type of event caused the interrupt.

At the same time, the PMT module of ETHMAC also has a non-maskable interrupt output. When a PMT event occurs, a PMT non-maskable interrupt is generated regardless of whether the PMTIM bit in the interrupt mask register (ETH\_MAC\_INTMSKR) is set to 1.

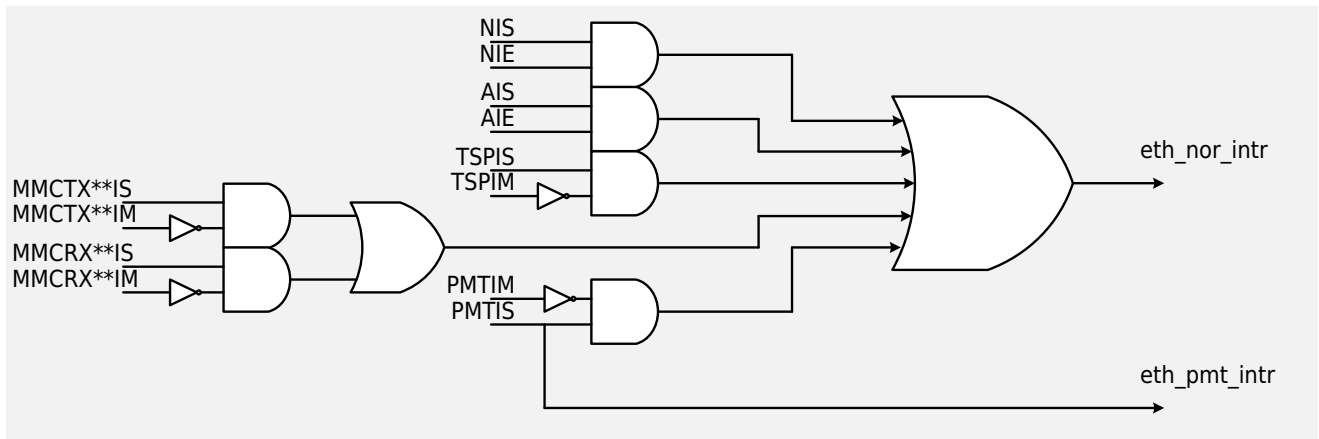


Figure 38-21 ETHMAC Interrupt Scheme

#### 38.3.6.2 DMA Interrupt

Various normal events generated by the DMA during transmit and receive data trigger normal interrupts, and various abnormal events trigger abnormal interrupts. As shown in the figure below.

The interrupt control of each specific event can be controlled by the respective bits in the DMA interrupt enable register ([ETH\\_DMA\\_INTENAR](#)).

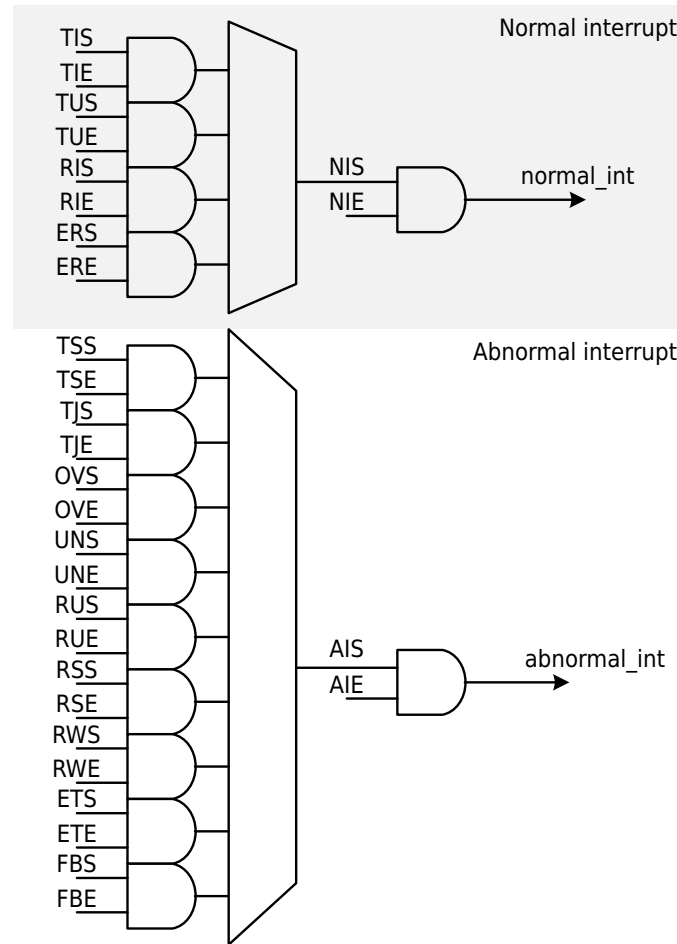


Figure 38-22 DMA Interrupt Components

### 38.3.6.3 PMT Interrupt

The MAC can trigger the generation of PMT interrupts (maskable and non-maskable interrupts) when it receives a remote wake-up frame and a magic packet. The generation of maskable interrupts is controlled by the PMTIM bit in the MAC interrupt mask register (ETH\_MAC\_INTMSKR).

### 38.3.6.4 PTP Interrupt

When the PTP function is enabled, a PTP interrupt can be triggered when the system time reaches the set value in the timestamp target time register 0-1. The interrupt is controlled by the TT0SEL and TT1SEL bits of the PPS output control register (ETH\_PTP\_PPSCTLR), the TSPINT bit of the ETH\_PTP timestamp control register (ETH\_PTP\_TSPCTLR), and the TSPIM bit of the MAC interrupt mask register (ETH\_MAC\_INTMSKR).

### 38.3.6.5 MMC Interrupt

MMC interrupts can be triggered when the transmit and receive frame statistics in the MMC overflow. The interrupt is controlled by the respective bits of the MMC Rx interrupt control register (ETH\_MMC\_RITCTLR) and the MMC Tx interrupt control register (ETH\_MMC\_TITCTLR).

## 38.4 Register Description

### 38.4.1 Register List

ETHMAC Common Register Base Address: 0x40055400

Table 38-25 ETHMAC Common Register Offset

| Offset | Register        | Description                          | Width |
|--------|-----------------|--------------------------------------|-------|
| 0x0010 | ETH_MAC_IFCONFR | ETH_MAC Interface Selection Register | 32    |

ETHMAC base address: 0x40060000

Table 38-26 Ethernet MAC Controller (ETHMAC) Register Offset

| Offset | Register         | Description                                  | Width |
|--------|------------------|----------------------------------------------|-------|
| 0x0000 | ETH_MAC_CONFIGR  | ETH_MAC MAC Configuration Register           | 32    |
| 0x0018 | ETH_MAC_FLOCTLR  | ETH_MAC MAC Flow Control Register            | 32    |
| 0x0024 | ETH_MAC_MACSTSR  | ETH_MAC MAC Status Register                  | 32    |
| 0x0038 | ETH_MAC_INTSTSR  | ETH_MAC Interrupt Status Register            | 32    |
| 0x003C | ETH_MAC_INTMSKR  | ETH_MAC Interrupt Mask Register              | 32    |
| 0x0010 | ETH_MAC_SMIADDR  | ETH_MAC SMI Address Register                 | 32    |
| 0x0014 | ETH_MAC_SMIDATR  | ETH_MAC SMI Data Register                    | 32    |
| 0x002C | ETH_MAC_PMTCTLR  | ETH_MAC PMT Control Status Register          | 32    |
| 0x0028 | ETH_MAC_RTWKFFR  | ETH_MAC Remote Wake-Up Frame Filter Register | 32    |
| 0x0004 | ETH_MAC_FLTCTLR  | ETH_MAC MAC Frame Filter Control Register    | 32    |
| 0x0040 | ETH_MAC_MACADHR0 | ETH_MAC MAC Address High Bit Register 0      | 32    |
| 0x0044 | ETH_MAC_MACADLR0 | ETH_MAC MAC Address Low Bit Register 0       | 32    |
| 0x0048 | ETH_MAC_MACADHR1 | ETH_MAC MAC Address High Bit Register 1      | 32    |
| 0x004C | ETH_MAC_MACADLR1 | ETH_MAC MAC Address Low Bit Register 1       | 32    |
| 0x0050 | ETH_MAC_MACADHR2 | ETH_MAC MAC Address High Bit Register 2      | 32    |
| 0x0054 | ETH_MAC_MACADLR2 | ETH_MAC MAC Address Low Bit Register 2       | 32    |
| 0x0058 | ETH_MAC_MACADHR3 | ETH_MAC MAC Address High Bit Register 3      | 32    |
| 0x005C | ETH_MAC_MACADLR3 | ETH_MAC MAC Address Low Bit Register 3       | 32    |
| 0x0060 | ETH_MAC_MACADHR4 | ETH_MAC MAC Address High Bit Register 4      | 32    |
| 0x0064 | ETH_MAC_MACADLR4 | ETH_MAC MAC Address Low Bit Register 4       | 32    |
| 0x0008 | ETH_MAC_HASHTHR  | ETH_MAC MAC Hash Table High Bit Register     | 32    |

| Offset | Register                         | Description                                   | Width |
|--------|----------------------------------|-----------------------------------------------|-------|
| 0x000C | <a href="#">ETH_MAC_HASHTLR</a>  | ETH_MAC MAC Hash Table Low Bit Register       | 32    |
| 0x0584 | <a href="#">ETH_MAC_VTACTLR</a>  | ETH_MAC VLAN Tag Transmit Control Register    | 32    |
| 0x001C | <a href="#">ETH_MAC_VTAFLTR</a>  | ETH_MAC VLAN Tag Receive Filter Register      | 32    |
| 0x0588 | <a href="#">ETH_MAC_VLAHTBR</a>  | ETH_MAC VLAN Hash Table Register              | 32    |
| 0x0400 | <a href="#">ETH_MAC_L34CTLR</a>  | ETH_MAC LAY3LAY4 Control Register             | 32    |
| 0x0404 | <a href="#">ETH_MAC_L4PORTR</a>  | ETH_MAC LAY4 Port Register                    | 32    |
| 0x0410 | <a href="#">ETH_MAC_L3ADDR0</a>  | ETH_MAC LAY3 Address Register 0               | 32    |
| 0x0414 | <a href="#">ETH_MAC_L3ADDR1</a>  | ETH_MAC LAY3 Address Register 1               | 32    |
| 0x0418 | <a href="#">ETH_MAC_L3ADDR2</a>  | ETH_MAC LAY3 Address Register 2               | 32    |
| 0x041C | <a href="#">ETH_MAC_L3ADDR3</a>  | ETH_MAC LAY3 Address Register 3               | 32    |
| 0x0700 | <a href="#">ETH_PTP_TSPCTLR</a>  | ETH_PTP Timestamp Control Register            | 32    |
| 0x0728 | <a href="#">ETH_PTP_TSPSTSR</a>  | ETH_PTP Timestamp Status Register             | 32    |
| 0x0718 | <a href="#">ETH_PTP_TSPADDR</a>  | ETH_PTP Timestamp Basic Addend Register       | 32    |
| 0x0704 | <a href="#">ETH_PTP_TSPNSAR</a>  | ETH_PTP Timestamp Subsecond Addend Register   | 32    |
| 0x0708 | <a href="#">ETH_PTP_TMSSECR</a>  | ETH_PTP Timestamp System Second Register      | 32    |
| 0x070C | <a href="#">ETH_PTP_TMSNSER</a>  | ETH_PTP Timestamp System Subsecond Register   | 32    |
| 0x0710 | <a href="#">ETH_PTP_TMUSECR</a>  | ETH_PTP Timestamp Update Second Register      | 32    |
| 0x0714 | <a href="#">ETH_PTP_TMUNSER</a>  | ETH_PTP Timestamp Update Subsecond Register   | 32    |
| 0x071C | <a href="#">ETH_PTP_TMTSECR0</a> | ETH_PTP Timestamp Target Second Register 0    | 32    |
| 0x0720 | <a href="#">ETH_PTP_TMTNSER0</a> | ETH_PTP Timestamp Target Subsecond Register 0 | 32    |
| 0x0780 | <a href="#">ETH_PTP_TMTSECR1</a> | ETH_PTP Timestamp Target Second Register 1    | 32    |
| 0x0784 | <a href="#">ETH_PTP_TMTNSER1</a> | ETH_PTP Timestamp Target Subsecond Register 1 | 32    |
| 0x072C | <a href="#">ETH_PTP_PPSCTLR</a>  | ETH_PTP PPS Output Control Register           | 32    |
| 0x1000 | <a href="#">ETH_DMA_BUSMODR</a>  | ETH_DMA Bus Mode Register                     | 32    |
| 0x1018 | <a href="#">ETH_DMA_OPRMODR</a>  | ETH_DMA Operation Mode Register               | 32    |
| 0x1014 | <a href="#">ETH_DMA_DMASTR</a>   | ETH_DMA Action Status Register                | 32    |

| Offset | Register                         | Description                                          | Width |
|--------|----------------------------------|------------------------------------------------------|-------|
| 0x101C | <a href="#">ETH_DMA_INTENAR</a>  | ETH_DMA Interrupt Enable Register                    | 32    |
| 0x1020 | <a href="#">ETH_DMA_RFR CNTR</a> | ETH_DMA Frame Loss Statistics Register               | 32    |
| 0x1024 | <a href="#">ETH_DMA_REVWDTR</a>  | ETH_DMA Watchdog Timer Register                      | 32    |
| 0x1004 | <a href="#">ETH_DMA_TXPOLLR</a>  | ETH_DMA Transmit Poll Demand Register                | 32    |
| 0x1008 | <a href="#">ETH_DMA_RXPOLLR</a>  | ETH_DMA Receive Poll Demand Register                 | 32    |
| 0x1010 | <a href="#">ETH_DMA_TXDLADR</a>  | ETH_DMA Transmit Descriptor List Address Register    | 32    |
| 0x100C | <a href="#">ETH_DMA_RXDLADR</a>  | ETH_DMA Receive Descriptor List Address Register     | 32    |
| 0x1048 | <a href="#">ETH_DMA_CHTXDER</a>  | ETH_DMA Current Host Transmit Descriptor Register    | 32    |
| 0x104C | <a href="#">ETH_DMA_CHRXDER</a>  | ETH_DMA Current Host Receive Descriptor Register     | 32    |
| 0x1050 | <a href="#">ETH_DMA_CHTXBFR</a>  | ETH_DMA Current Host Transmit Buffer Register        | 32    |
| 0x1054 | <a href="#">ETH_DMA_CHRXBFR</a>  | ETH_DMA Current Host Receive Buffer Register         | 32    |
| 0x0100 | <a href="#">ETH_MMC_MMCCTLR</a>  | ETH_MMC MMC Control Register                         | 32    |
| 0x0104 | <a href="#">ETH_MMC_REVSTSR</a>  | ETH_MMC Rx Statistics Status Register                | 32    |
| 0x0108 | <a href="#">ETH_MMC_TRSSTSR</a>  | ETH_MMC Tx Statistics Status Register                | 32    |
| 0x010C | <a href="#">ETH_MMC_RITCTLR</a>  | ETH_MMC Rx Interrupt Control Register                | 32    |
| 0x0110 | <a href="#">ETH_MMC_TITCTLR</a>  | ETH_MMC Tx Interrupt Control Register                | 32    |
| 0x01C4 | <a href="#">ETH_MMC_RXUNGFR</a>  | ETH_MMC Rx Unicast Good Frame Statistics Register    | 32    |
| 0x0190 | <a href="#">ETH_MMC_RXMUGFR</a>  | ETH_MMC Rx Multicast Good Frame Statistics Register  | 32    |
| 0x018C | <a href="#">ETH_MMC_RXBRGFR</a>  | ETH_MMC Rx Broadcast Good Frame Statistics Register  | 32    |
| 0x0194 | <a href="#">ETH_MMC_RXCREFR</a>  | ETH_MMC Rx CRC Error Frame Statistics Register       | 32    |
| 0x0198 | <a href="#">ETH_MMC_RXALEFR</a>  | ETH_MMC Rx Alignment Error Frame Statistics Register | 32    |
| 0x019C | <a href="#">ETH_MMC_RXRUEFR</a>  | ETH_MMC Rx Dwarf Error Frame Statistics Register     | 32    |
| 0x01C8 | <a href="#">ETH_MMC_RXLEEFR</a>  | ETH_MMC Rx Length Error Frame Statistics Register    | 32    |
| 0x01CC | <a href="#">ETH_MMC_RXOREFR</a>  | ETH_MMC Rx Oversize Error Frame Statistics Register  | 32    |

| Offset | Register                        | Description                                                    | Width |
|--------|---------------------------------|----------------------------------------------------------------|-------|
| 0x0168 | <a href="#">ETH_MMC_TXUNGFR</a> | ETH_MMC Tx Unicast Good Frame Statistics Register              | 32    |
| 0x0120 | <a href="#">ETH_MMC_TXMUGFR</a> | ETH_MMC Tx Multicast Good Frame Statistics Register            | 32    |
| 0x011C | <a href="#">ETH_MMC_TXBRGFR</a> | ETH_MMC Tx Broadcast Good Frame Statistics Register            | 32    |
| 0x0154 | <a href="#">ETH_MMC_TXDEEFR</a> | ETH_MMC Tx Deferred Error Frame Statistics Register            | 32    |
| 0x0160 | <a href="#">ETH_MMC_TXCAEFR</a> | ETH_MMC Tx Carrier Error Frame Statistics Register             | 32    |
| 0x0158 | <a href="#">ETH_MMC_TXLCEFR</a> | ETH_MMC Tx Delay Collision Error Frame Statistics Register     | 32    |
| 0x015C | <a href="#">ETH_MMC_TXECEFR</a> | ETH_MMC Tx Excessive Collision Error Frame Statistics Register | 32    |
| 0x016C | <a href="#">ETH_MMC_TXEDEFR</a> | ETH_MMC Tx Excessive Delay Error Frame Statistics Register     | 32    |



## 38.4.2 ETH\_MAC Register

### 38.4.2.1 ETH\_MAC Interface Selection Register (ETH\_MAC\_IFCONFR)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |        |   |        |   |          |   |       |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|--------|---|--------|---|----------|---|-------|---|
| 0x0010 | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7      | 6 | 5      | 4 | 3        | 2 | 1     | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |        |   |        |   |          |   |       |   |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   | TCKINV |   | RCKINV |   | Reserved |   | IFSEL |   |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   | RW     |   | RW     |   |          |   | RW    |   |

| Bit  | Name     | Function                | Description                                                                                                                                                                                                                        | Access |
|------|----------|-------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:6 | Reserved | Reserved                | -                                                                                                                                                                                                                                  | -      |
| 5    | TCKINV   | TX Clock Inversion      | <ul style="list-style-type: none"> <li>0b0: The clock input from the ETH_MII_TXCLK port maintains its original polarity</li> <li>0b1: The polarity of the clock input from the ETH_MII_TXCLK port is inverted</li> </ul>           | RW     |
| 4    | RCKINV   | RX Clock Inversion      | <ul style="list-style-type: none"> <li>0b0: The clock input from the ETH_MII_RMII_RXCLK port maintains its original polarity</li> <li>0b1: The polarity of the clock input from the ETH_MII_RMII_RXCLK port is inverted</li> </ul> | RW     |
| 3:1  | Reserved | Reserved                | -                                                                                                                                                                                                                                  | -      |
| 0    | IFSEL    | Interface Selection Bit | <ul style="list-style-type: none"> <li>0b0: MII interface</li> <li>0b1: RMII interface</li> </ul>                                                                                                                                  | RW     |

## 38.4.2.2 ETH\_MAC MAC Configuration Register (ETH\_MAC\_CONFIGR)

| Offset | Bit Position |       |    |          |    |     |          |     |     |          |    |     |    |      |          |     |    |    |    |      |      |          |     |    |   |    |    |    |          |   |   |   |
|--------|--------------|-------|----|----------|----|-----|----------|-----|-----|----------|----|-----|----|------|----------|-----|----|----|----|------|------|----------|-----|----|---|----|----|----|----------|---|---|---|
| 0x0000 | 31           | 30    | 29 | 28       | 27 | 26  | 25       | 24  | 23  | 22       | 21 | 20  | 19 | 18   | 17       | 16  | 15 | 14 | 13 | 12   | 11   | 10       | 9   | 8  | 7 | 6  | 5  | 4  | 3        | 2 | 1 | 0 |
| Reset  | 0x00008000   |       |    |          |    |     |          |     |     |          |    |     |    |      |          |     |    |    |    |      |      |          |     |    |   |    |    |    |          |   |   |   |
| Name   | Reserved     | SAIRC |    | Reserved |    | CST | Reserved | MWD | MJB | Reserved |    | IFG |    | DCRS | Reserved | FES | DO | LM | DM | IPCO | DRTY | Reserved | ACS | BL |   | DC | TE | RE | Reserved |   |   |   |
| Access |              | RW    |    |          |    | RW  |          | RW  | RW  |          |    | RW  |    | RW   |          | RW  | RW | RW | RW | RW   | RW   |          | RW  | RW |   | RW | RW | RW | RW       |   |   |   |

| Bit   | Name     | Function                                 | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | Access |
|-------|----------|------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31    | Reserved | Reserved                                 | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | -      |
| 30:28 | SAIRC    | Source Address Insertion and Replacement | <ul style="list-style-type: none"> <li>0b010: Insert the address value from MAC address register 0 as the SA address into the transmit frame</li> <li>0b011: Replace the SA field in the transmit frame with the address value from MAC address register 0 as the SA address</li> <li>0b110: Insert the address value from MAC address register 1 as the SA address into the transmit frame</li> <li>0b111: Replace the SA field in the transmit frame with the address value from MAC address register 1 as the SA address</li> <li>0bx0x: The processing of the SA field address in the transmitted frame is controlled by the SAIRC bit of the Tx descriptor TDES1</li> </ul> | RW     |
| 27:26 | Reserved | Reserved                                 | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | -      |
| 25    | CST      | TYPE Frame FCS Removal                   | <ul style="list-style-type: none"> <li>0b0: The function is disabled</li> <li>0b1: Before forwarding the frame to the application, remove and discard the last 4 bytes (FCS field) of all TYPE frames (type field greater than 0x0600)</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                | RW     |
| 24    | Reserved | Reserved                                 | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | -      |
| 23    | MWD      | Watchdog Disabled                        | <ul style="list-style-type: none"> <li>0b0: The MAC allows received frames not to exceed 2048 bytes and truncates any bytes received beyond this limit</li> <li>0b1: The MAC disables the watchdog timer on the receiver and can receive frames up to 16384 bytes</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                     | RW     |
| 22    | MJB      | Jabber Disabled                          | <ul style="list-style-type: none"> <li>0b0: If the application sends data exceeding 2048 bytes during transmission, the MAC truncates the transmitter</li> <li>0b1: The MAC disables the Jabber timer on the transmitter and can send frames up to 16384 bytes</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                        | RW     |
| 21:20 | Reserved | Reserved                                 | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | -      |

| Bit   | Name            | Function                    | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | Access |
|-------|-----------------|-----------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 19:17 | IFG             | Inter-frame Gap             | Minimum interframe gap during transmission: <ul style="list-style-type: none"> <li>● 0b000: 96-bit time</li> <li>● 0b001: 88-bit time</li> <li>● 0b010: 80-bit time</li> <li>● .....</li> <li>● 0b111: 40-bit time</li> </ul> <div>  <b>Note</b><br/>           In half-duplex mode, the minimum IFG can only be configured to 64-bit time (IFG=100), and lower values are not considered.         </div>                                                                                                                                                   | RW     |
| 16    | DCRS            | Carrier Sense Disabled      | <ul style="list-style-type: none"> <li>● 0b0: In half-duplex mode, when the MII_CRS signal is enable during frame transmission, the MAC transmitter generates a carrier sense error and may even abort the transmission</li> <li>● 0b1: In half-duplex mode, the MII_CRS signal during frame transmission is ignored, and no error is generated due to carrier loss or no carrier during this transmission period</li> </ul> <div>  <b>Note</b><br/>           This bit is only active in half-duplex mode and inactive in full-duplex mode.         </div> | RW     |
| 15    | <i>Reserved</i> | <i>Reserved</i>             | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | -      |
| 14    | FES             | Speed Selection             | <ul style="list-style-type: none"> <li>● 0b0: 10Mbps</li> <li>● 0b1: 100Mbps</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | RW     |
| 13    | DO              | Disable Receive Own Packets | <ul style="list-style-type: none"> <li>● 0b0: In half-duplex mode, the MAC receives all packets from the PHY</li> <li>● 0b1: In half-duplex mode, the MAC disables receiving frames</li> </ul> <div>  <b>Note</b><br/>           This bit is only active in half-duplex mode and inactive in full-duplex mode.         </div>                                                                                                                                                                                                                             | RW     |
| 12    | LM              | Loopback Mode               | <ul style="list-style-type: none"> <li>● 0b0: Loopback mode disabled</li> <li>● 0b1: Loopback mode enabled</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | RW     |
| 11    | DM              | Full-Duplex Mode            | <ul style="list-style-type: none"> <li>● 0b0: Half-Duplex Mode</li> <li>● 0b1: Full-Duplex Mode</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | RW     |
| 10    | IPCO            | Checksum Offload Function   | <ul style="list-style-type: none"> <li>● 0b0: The IP Checksum Offload function is disabled</li> <li>● 0b1: The IP Checksum Offload function is enable</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | RW     |

| Bit | Name     | Function                  | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | Access |
|-----|----------|---------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 9   | DRTY     | Retry Disabled            | <ul style="list-style-type: none"> <li>0b0: In MII mode, when a collision occurs, the MAC attempts to retry based on the setting of the BL bit in this register</li> <li>0b1: In MII mode, when a collision occurs, the MAC ignores the current frame transmission and terminate with an excessive collision error report frame in the transmit frame state, that is, the MAC only attempts to send once</li> </ul> <div>  <b>Note</b><br/>           This bit is only active in half-duplex mode and inactive in full-duplex mode.         </div>                                                                                                                               | RW     |
| 8   | Reserved | Reserved                  | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | -      |
| 7   | ACS      | Automatic PAD/FCS Removal | <ul style="list-style-type: none"> <li>0b0: The MAC transmits all received frames to the application regardless of the length of the received frame</li> <li>0b1: The MAC automatically removes the PAD/FCS field from the received frame when the length field value is less than 1536 bytes, and does not remove the PAD/FCS field when the length field is greater than or equal to 1536 bytes</li> </ul>                                                                                                                                                                                                                                                                                                                                                      | RW     |
| 6:5 | BL       | Backoff Limit             | <p>The backoff limit determines the random integer (r) slot time delays (512-bit time) that the MAC waits before rescheduling a transmission after a collision during retry:</p> <ul style="list-style-type: none"> <li>0b00: <math>k = \min(n, 10)</math></li> <li>0b01: <math>k = \min(n, 8)</math></li> <li>0b10: <math>k = \min(n, 4)</math></li> <li>0b11: <math>k = \min(n, 1)</math></li> </ul> <p>where n is the number of retransmission attempts. The value range of the random integer r is:<br/> <math>0 \leq r &lt; 2^k</math></p> <div>  <b>Note</b><br/>           This bit is only active in half-duplex mode and inactive in full-duplex mode.         </div> | RW     |

| Bit | Name     | Function         | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | Access |
|-----|----------|------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 4   | DC       | Delay Check      | <ul style="list-style-type: none"> <li>● 0b0: The delay check function is disabled. The MAC is delayed until the CRS signal becomes inactive</li> <li>● 0b1: The delay check function is enabled. When the transmit state machine delays for more than 24288 bits of time, the MAC indicates a frame abort state and set the excessive delay error bit to 1 in the transmit frame state</li> </ul> <div>  <b>Note</b> <ol style="list-style-type: none"> <li>1. Delay generation: The delay count starts when the transmitter is ready to send but is blocked by the presence of a active CRS (carrier sense) signal on the MII. The delay time is non-cumulative. If the transmitter delays for 10000 bits (bit) of time, it re-executes the transmission, collision, and backoff, and then reset the delay counter to 0 after completing the backoff and restart counting</li> <li>2. This bit is only active in half-duplex mode and inactive in full-duplex mode.</li> </ol> </div> | RW     |
| 3   | TE       | Transmit Enabled | <ul style="list-style-type: none"> <li>● 0b0: Transmit disabled</li> <li>● 0b1: Transmit enabled</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | RW     |
| 2   | RE       | Receive Enabled  | <ul style="list-style-type: none"> <li>● 0b0: Receive disabled</li> <li>● 0b1: Receive enabled</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | RW     |
| 1:0 | Reserved | Reserved         | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | -      |

## 38.4.2.3 ETH\_MAC MAC Flow Control Register (ETH\_MAC\_FLOCTLR)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |          |    |    |    |    |    |   |   |    |      |          |   |   |   |   |   |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |    |  |  |    |     |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----------|----|----|----|----|----|---|---|----|------|----------|---|---|---|---|---|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|----|--|--|----|-----|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
| 0x0018 | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15       | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7  | 6    | 5        | 4 | 3 | 2 | 1 | 0 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |    |  |  |    |     |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |          |    |    |    |    |    |   |   |    |      |          |   |   |   |   |   |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |    |  |  |    |     |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Name   | PAUSET       |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | Reserved |    |    |    |    |    |   |   |    | DZPQ | Reserved |   |   |   |   |   |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |    |  |  |    |     |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |          |    |    |    |    |    |   |   | RW |      |          |   |   |   |   |   |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | RW |  |  | RW | PLT |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|        |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |          |    |    |    |    |    |   |   |    |      |          |   |   |   |   |   |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |    |  |  |    |     |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

| Bit   | Name            | Function                              | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | Access |
|-------|-----------------|---------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:16 | PAUSET          | Pause Time                            | This bit indicates the value to be used in the pause time field of the transmit control frame                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | RW     |
| 15:8  | <i>Reserved</i> | <i>Reserved</i>                       | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | -      |
| 7     | DZPQ            | Zero-length Time Slice Frame Disabled | <ul style="list-style-type: none"> <li>0b0: When the flow control signal is enabled, a zero-length time slice pause control frame is normally generated</li> <li>0b1: When the flow control signal is enabled, the generation of zero-length time slice pause control frames is disabled</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | RW     |
| 6     | <i>Reserved</i> | <i>Reserved</i>                       | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | -      |
| 5:4   | PLT             | Pause Lower Threshold                 | <p>Configure the threshold of the pause timer, and the PAUSE frame is retransmitted automatically when this value is reached:</p> <ul style="list-style-type: none"> <li>0b00: Subtract 4 time slots from the pause time</li> <li>0b01: Subtract 28 time slots from the pause time</li> <li>0b10: Subtract 144 time slots from the pause time</li> <li>0b11: Subtract 256 time slots from the pause time</li> </ul> <div>  <b>Note</b> <ol style="list-style-type: none"> <li>A time slot is defined as the time required for the MII interface to send 512 bits (64 bytes).</li> <li>This threshold should always be less than the pause time. For example, if the pause time PAUSET=0x100 (256 time slots), and PLT=0b01, then after the first PAUSE frame is sent, the second PAUSE frame is automatically sent when the 228th (256-28) time slot starts.</li> </ol> </div> | RW     |

| Bit | Name    | Function                      | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | Access |
|-----|---------|-------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 3   | UNP     | Unicast Pause Frame Detection | <ul style="list-style-type: none"> <li>0b0: The MAC only detects pause frames with the unique multicast address specified in the 802.3x standard</li> <li>0b1: In addition to detecting pause frames with the unique multicast address, the MAC also detects pause frames with the station unicast address specified by the ETH_MAC_MACADHR0 and ETH_MAC_MACADLR0 registers</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                         | RW     |
| 2   | RFE     | Receive Flow Control Enabled  | <ul style="list-style-type: none"> <li>0b0: The decoding function of pause frames is disabled</li> <li>0b1: The MAC decodes the received pause frames and disables their transmission during the specified pause time</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | RW     |
| 1   | TFE     | Transmit Flow Control Enabled | <p><b>In Full-duplex Mode:</b></p> <ul style="list-style-type: none"> <li>0b0: The MAC is disabled from transmitting any pause frames</li> <li>0b1: Flow control operation is enabled, and the MAC sends pause frames</li> </ul> <p><b>In Half-duplex Mode:</b></p> <ul style="list-style-type: none"> <li>0b0: Backpressure function is disabled</li> <li>0b1: Backpressure operation is enabled</li> </ul>                                                                                                                                                                                                                                                                                                                                                                   | RW     |
| 0   | FCA_BPA | Flow Control Status           | <p>This bit should be read as 0 before writing data to the flow control register</p> <p><b>In Full-duplex Mode:</b><br/>When this bit is set to 1, the MAC core initiates a pause control frame. During the control frame transmission process, this bit remains set to 1 to indicate that the frame transmission is in progress. After the pause control frame transmission is completed, the MAC resets this bit to 0. After this bit is cleared, write operations can be performed on the flow control register</p> <p><b>In Half-duplex Mode:</b><br/>When this bit is set to 1, the MAC core enables the backpressure function. During backpressure operation, when the MAC receives a new frame, the transmitter starts sending a Jam signal that causes a collision</p> | RW     |

## 38.4.2.4 ETH\_MAC MAC Status Register (ETH\_MAC\_MACSTSR)

| Offset | Bit Position |    |    |    |    |    |     |      |          |      |      |     |     |      |          |    |    |    |    |    |      |          |      |      |          |     |      |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|-----|------|----------|------|------|-----|-----|------|----------|----|----|----|----|----|------|----------|------|------|----------|-----|------|---|---|---|---|---|
| 0x0024 | 31           | 30 | 29 | 28 | 27 | 26 | 25  | 24   | 23       | 22   | 21   | 20  | 19  | 18   | 17       | 16 | 15 | 14 | 13 | 12 | 11   | 10       | 9    | 8    | 7        | 6   | 5    | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |     |      |          |      |      |     |     |      |          |    |    |    |    |    |      |          |      |      |          |     |      |   |   |   |   |   |
| Name   | Reserved     |    |    |    |    |    | TFF | TFNE | Reserved | TFWA | TFRS | MTP | MTS | MTEA | Reserved |    |    |    |    |    | RFFL | Reserved | RFRS | RFWA | Reserved | MRS | MREA |   |   |   |   |   |
| Access |              |    |    |    |    |    | RO  | RO   |          | RO   | RO   | RO  | RO  | RO   |          |    |    |    |    |    | RO   |          | RO   | RO   |          | RO  | RO   |   |   |   |   |   |

| Bit   | Name     | Function                       | Description                                                                                                                                                                                                                                                                                        | Access |
|-------|----------|--------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:26 | Reserved | Reserved                       | -                                                                                                                                                                                                                                                                                                  | -      |
| 25    | TFF      | Tx FIFO full                   | <ul style="list-style-type: none"> <li>0b0: The Tx FIFO is not full</li> <li>0b1: The Tx FIFO is full and cannot receive any more transmitted frames</li> </ul>                                                                                                                                    | RO     |
| 24    | TFNE     | Tx FIFO not empty              | <ul style="list-style-type: none"> <li>0b0: The Tx FIFO is empty</li> <li>0b1: The Tx FIFO is not empty and there are still frames to be sent</li> </ul>                                                                                                                                           | RO     |
| 23    | Reserved | Reserved                       | -                                                                                                                                                                                                                                                                                                  | -      |
| 22    | TFWA     | Tx FIFO write enabled          | <ul style="list-style-type: none"> <li>0b0: Tx FIFO write is disabled</li> <li>0b1: The Tx FIFO write controller is enable and is transferring data to the Tx FIFO</li> </ul>                                                                                                                      | RO     |
| 21:20 | TFRS     | Tx FIFO read status            | <ul style="list-style-type: none"> <li>0b00: Idle state</li> <li>0b01: Read state (data is transferred from the Tx FIFO to the MAC transmitter)</li> <li>0b10: Waiting for Tx Status from the MAC transmitter</li> <li>0b11: Write the received Tx Status or refresh the Tx FIFO</li> </ul>        | RO     |
| 19    | MTP      | MAC transmitter pause          | <ul style="list-style-type: none"> <li>0b0: The MAC transmitter is in the transmit data state</li> <li>0b1: The MAC transmitter is in the pause state (in full-duplex mode) and does not schedule any frame transmission</li> </ul>                                                                | RO     |
| 18:17 | MTS      | MAC transmitter status         | <ul style="list-style-type: none"> <li>0b00: Idle state</li> <li>0b01: Waiting for the status of the previous frame or the end of the IFG/backoff phase</li> <li>0b10: Generate and send a pause control frame (in full-duplex mode)</li> <li>0b11: Transmit the input frame to be sent</li> </ul> | RO     |
| 16    | MTEA     | MAC MII transmit engine status | <ul style="list-style-type: none"> <li>0b0: The MAC MII transmit engine is in the idle state</li> <li>0b1: The MAC MII transmit engine is actively transmitting data and is not in the idle state</li> </ul>                                                                                       | RO     |



| Bit   | Name            | Function                      | Description                                                                                                                                                                                                                                                                                            | Access |
|-------|-----------------|-------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 15:10 | <i>Reserved</i> | <i>Reserved</i>               | -                                                                                                                                                                                                                                                                                                      | -      |
| 9:8   | RFLL            | Rx FIFO fill level            | <ul style="list-style-type: none"> <li>● 0b00: The Rx FIFO is empty</li> <li>● 0b01: The Rx FIFO fill level is below flow control, deactivate the threshold</li> <li>● 0b10: The Rx FIFO fill level is above flow control, activate the threshold</li> <li>● 0b11: The Rx FIFO is full</li> </ul>      | RO     |
| 7     | <i>Reserved</i> | <i>Reserved</i>               | -                                                                                                                                                                                                                                                                                                      | -      |
| 6:5   | RFRS            | Rx FIFO read status           | <ul style="list-style-type: none"> <li>● 0b00: Idle state</li> <li>● 0b01: Read frame data</li> <li>● 0b10: Read frame status or timestamp</li> <li>● 0b11: Refresh frame data and status</li> </ul>                                                                                                   | RO     |
| 4     | RFWA            | Rx FIFO write enabled         | <ul style="list-style-type: none"> <li>● 0b0: Rx FIFO write is disabled</li> <li>● 0b1: Rx FIFO write is enable and is transferring the received frame to the Rx FIFO</li> </ul>                                                                                                                       | RO     |
| 3     | <i>Reserved</i> | <i>Reserved</i>               | -                                                                                                                                                                                                                                                                                                      | -      |
| 2:1   | MRS             | MAC receiver state            | <ul style="list-style-type: none"> <li>● 0b00: Idle</li> <li>● 0b01: The FIFO write controller in the MAC receiver is working</li> <li>● 0b10: The FIFO read controller in the MAC receiver is working</li> <li>● 0b11: The FIFO read and write controllers in the MAC receiver are working</li> </ul> | RO     |
| 0     | MREA            | MAC MII receive engine status | <ul style="list-style-type: none"> <li>● 0b0: The MAC MII receive engine is in the idle state</li> <li>● 0b1: The MAC MII receive engine is actively receiving data and is not in the idle state</li> </ul>                                                                                            | RO     |

## 38.4.2.5 ETH\_MAC Interrupt Status Register (ETH\_MAC\_INTSTSR)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |       |          |         |         |       |       |          |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|-------|----------|---------|---------|-------|-------|----------|---|---|
| 0x0038 | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8     | 7        | 6       | 5       | 4     | 3     | 2        | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |       |          |         |         |       |       |          |   |   |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   | TSPIS | Reserved | MMCTXIS | MMCRXIS | MMCIS | PMTIS | Reserved |   |   |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   | RO    |          | RO      | RO      | RO    | RO    |          |   |   |

| Bit   | Name     | Function                          | Description                                                                                                                                                                                                                                                                                                                                               | Access |
|-------|----------|-----------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:10 | Reserved | Reserved                          | -                                                                                                                                                                                                                                                                                                                                                         | -      |
| 9     | TSPIS    | PTP timestamp interrupt           | <p>When the system time value is equal to or greater than the value specified in the target time register, or when the system time overflows, this bit is set to 1</p> <p> <b>Note</b><br/>This bit is cleared when reading the Bit0 of the ETH_PTP_TSPSTSR register</p> | RO     |
| 8:7   | Reserved | Reserved                          | -                                                                                                                                                                                                                                                                                                                                                         | -      |
| 6     | MMCTXIS  | MMC transmit statistics interrupt | <p>Interrupt generated by ETH_MMC_TITCTLR register, this bit is set to high</p> <p> <b>Note</b><br/>This bit is cleared when all interrupts described in the ETH_MMC_TITCTLR register are cleared</p>                                                                  | RO     |
| 5     | MMCRXIS  | MMC receive statistics interrupt  | <p>When an interrupt is generated in the ETH_MMC_RITCTLR register, this bit becomes high level</p> <p> <b>Note</b><br/>This bit is cleared when all interrupts described in the ETH_MMC_RITCTLR register are cleared</p>                                               | RO     |
| 4     | MMCIS    | MMC statistic interrupt           | <p>When the MMCTXIS or MMCRXIS bit of this register has a high level, this bit becomes high level</p> <p> <b>Note</b><br/>This bit is cleared when Bit6 to Bit5 of this register are all set to 0</p>                                                                  | RO     |

| Bit | Name     | Function      | Description                                                                                                                                                                                                                                                                                                                  | Access |
|-----|----------|---------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 3   | PMTIS    | PMT interrupt | <p>When a magic packet or remote wakeup frame is received in Power-down mode, this bit becomes high level</p> <div> <b>Note</b><br/>This bit is cleared when both the WKFR and MPFR bits of the ETH_MAC_PMTCTLR register are set to 0</div> | RO     |
| 2:0 | Reserved | Reserved      | -                                                                                                                                                                                                                                                                                                                            | -      |

## 38.4.2.6 ETH\_MAC Interrupt Mask Register (ETH\_MAC\_INTMSKR)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |       |          |   |   |   |   |       |          |   |  |  |  |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|-------|----------|---|---|---|---|-------|----------|---|--|--|--|
| 0x003C | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8     | 7        | 6 | 5 | 4 | 3 | 2     | 1        | 0 |  |  |  |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |       |          |   |   |   |   |       |          |   |  |  |  |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   | TSPIM | Reserved |   |   |   |   | PMTIM | Reserved |   |  |  |  |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   | RW    |          |   |   |   |   | RW    |          |   |  |  |  |

| Bit   | Name     | Function                     | Description                                                                                                                                          | Access |
|-------|----------|------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:10 | Reserved | Reserved                     | -                                                                                                                                                    | -      |
| 9     | TSPIM    | PTP Timestamp Interrupt Mask | <ul style="list-style-type: none"> <li>0b0: The PTP timestamp interrupt is not masked</li> <li>0b1: The PTP timestamp interrupt is masked</li> </ul> | RW     |
| 8:4   | Reserved | Reserved                     | -                                                                                                                                                    | -      |
| 3     | PMTIM    | PMT Interrupt Mask           | <ul style="list-style-type: none"> <li>0b0: The PMT interrupt is not masked</li> <li>0b1: The PMT interrupt is masked</li> </ul>                     | RW     |
| 2:0   | Reserved | Reserved                     | -                                                                                                                                                    | -      |

## 38.4.2.7 ETH\_MAC SMI Address Register (ETH\_MAC\_SMIADDR)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |      |    |    |    |      |    |   |   |      |   |   |   |      |   |      |    |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|------|----|----|----|------|----|---|---|------|---|---|---|------|---|------|----|
| 0x0010 | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15   | 14 | 13 | 12 | 11   | 10 | 9 | 8 | 7    | 6 | 5 | 4 | 3    | 2 | 1    | 0  |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |      |    |    |    |      |    |   |   |      |   |   |   |      |   |      |    |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | SMIA |    |    |    | SMIR |    |   |   | SMIC |   |   |   | SMIW |   | SMIB |    |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |      |    |    |    |      |    |   |   |      |   |   |   | RW   |   |      | RW |

| Bit   | Name     | Function               | Description                                                                                                                                                                                                                                                                                                                                                                                                                 | Access |                  |                     |           |        |           |                     |        |            |                     |        |          |                     |        |          |                     |       |   |          |
|-------|----------|------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|------------------|---------------------|-----------|--------|-----------|---------------------|--------|------------|---------------------|--------|----------|---------------------|--------|----------|---------------------|-------|---|----------|
| 31:16 | Reserved | Reserved               | -                                                                                                                                                                                                                                                                                                                                                                                                                           | -      |                  |                     |           |        |           |                     |        |            |                     |        |          |                     |        |          |                     |       |   |          |
| 15:11 | SMIA     | Physical Layer Address | Indicates which of the 32 possible PHY devices to access                                                                                                                                                                                                                                                                                                                                                                    | RW     |                  |                     |           |        |           |                     |        |            |                     |        |          |                     |        |          |                     |       |   |          |
| 10:6  | SMIR     | PHY Register           | Indicates which register in the selected PHY device to select                                                                                                                                                                                                                                                                                                                                                               | RW     |                  |                     |           |        |           |                     |        |            |                     |        |          |                     |        |          |                     |       |   |          |
| 5:2   | SMIC     | MDC Clock              | Based on the system frequency (PCLK1) to determine the MDC clock frequency:                                                                                                                                                                                                                                                                                                                                                 | RW     |                  |                     |           |        |           |                     |        |            |                     |        |          |                     |        |          |                     |       |   |          |
|       |          |                        | <table><tr><th>Option</th><th>System Frequency</th><th>MDC Clock</th></tr><tr><td>0b0000</td><td>60-100MHz</td><td>System Frequency/42</td></tr><tr><td>0b0001</td><td>100-120MHz</td><td>System Frequency/62</td></tr><tr><td>0b0010</td><td>20-35MHz</td><td>System Frequency/16</td></tr><tr><td>0b0011</td><td>35-60MHz</td><td>System Frequency/26</td></tr><tr><td>Other</td><td>-</td><td>Reserved</td></tr></table> |        | Option           | System Frequency    | MDC Clock | 0b0000 | 60-100MHz | System Frequency/42 | 0b0001 | 100-120MHz | System Frequency/62 | 0b0010 | 20-35MHz | System Frequency/16 | 0b0011 | 35-60MHz | System Frequency/26 | Other | - | Reserved |
|       |          |                        | Option                                                                                                                                                                                                                                                                                                                                                                                                                      |        | System Frequency | MDC Clock           |           |        |           |                     |        |            |                     |        |          |                     |        |          |                     |       |   |          |
|       |          |                        | 0b0000                                                                                                                                                                                                                                                                                                                                                                                                                      |        | 60-100MHz        | System Frequency/42 |           |        |           |                     |        |            |                     |        |          |                     |        |          |                     |       |   |          |
|       |          |                        | 0b0001                                                                                                                                                                                                                                                                                                                                                                                                                      |        | 100-120MHz       | System Frequency/62 |           |        |           |                     |        |            |                     |        |          |                     |        |          |                     |       |   |          |
|       |          |                        | 0b0010                                                                                                                                                                                                                                                                                                                                                                                                                      |        | 20-35MHz         | System Frequency/16 |           |        |           |                     |        |            |                     |        |          |                     |        |          |                     |       |   |          |
|       |          |                        | 0b0011                                                                                                                                                                                                                                                                                                                                                                                                                      |        | 35-60MHz         | System Frequency/26 |           |        |           |                     |        |            |                     |        |          |                     |        |          |                     |       |   |          |
| Other | -        | Reserved               |                                                                                                                                                                                                                                                                                                                                                                                                                             |        |                  |                     |           |        |           |                     |        |            |                     |        |          |                     |        |          |                     |       |   |          |
|       |          |                        |                                                                                                                                                                                                                                                                                                                                                                                                                             |        |                  |                     |           |        |           |                     |        |            |                     |        |          |                     |        |          |                     |       |   |          |
|       |          |                        |                                                                                                                                                                                                                                                                                                                                                                                                                             |        |                  |                     |           |        |           |                     |        |            |                     |        |          |                     |        |          |                     |       |   |          |
|       |          |                        |                                                                                                                                                                                                                                                                                                                                                                                                                             |        |                  |                     |           |        |           |                     |        |            |                     |        |          |                     |        |          |                     |       |   |          |
|       |          |                        |                                                                                                                                                                                                                                                                                                                                                                                                                             |        |                  |                     |           |        |           |                     |        |            |                     |        |          |                     |        |          |                     |       |   |          |
|       |          |                        |                                                                                                                                                                                                                                                                                                                                                                                                                             |        |                  |                     |           |        |           |                     |        |            |                     |        |          |                     |        |          |                     |       |   |          |
| 1     | SMIW     | SMI Read/Write         | <ul style="list-style-type: none"><li>● 0b0: SMI read operation</li><li>● 0b1: SMI write operation</li></ul>                                                                                                                                                                                                                                                                                                                | RW     |                  |                     |           |        |           |                     |        |            |                     |        |          |                     |        |          |                     |       |   |          |

| Bit | Name | Function        | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | Access |
|-----|------|-----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 0   | SMIB | SMI Access Busy | <ul style="list-style-type: none"> <li>● This bit should be read as logic 0 before writing to ETH_MAC_SMIADDR and ETH_MAC_SMIDATR</li> <li>● This bit must also be reset to 0 during the write process to ETH_MAC_SMIADDR</li> <li>● During the PHY register access process, this bit is set to 1 by the application to indicate that a read or write access is in progress</li> </ul> <p>During the write operation to the PHY, ETH_MAC_SMIDATR should always be active until the MAC clears this bit; during the read operation to the PHY, ETH_MAC_SMIDATR is always inactive until the MAC clears this bit. After this bit is cleared, a value can be written to ETH_MAC_SMIADDR</p> | RW     |

## 38.4.2.8 ETH\_MAC SMI Data Register (ETH\_MAC\_SMIDATR)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |      |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0x0014 | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15   | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |      |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | SMID |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RW   |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit   | Name     | Function | Description                                                                                                                                                                          | Access |
|-------|----------|----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:16 | Reserved | Reserved | -                                                                                                                                                                                    | -      |
| 15:0  | SMID     | SMI Data | The 16-bit data value read from the PHY after the station management read operation, or the 16-bit data value to be written to the PHY before the station management write operation | RW     |

## 38.4.2.9 ETH\_MAC PMT Control Status Register (ETH\_MAC\_PMTCTLR)

| Offset | Bit Position |          |    |    |    |        |    |    |    |          |    |    |    |    |    |    |    |    |    |    |    |        |      |   |          |    |      |      |   |          |    |      |      |      |
|--------|--------------|----------|----|----|----|--------|----|----|----|----------|----|----|----|----|----|----|----|----|----|----|----|--------|------|---|----------|----|------|------|---|----------|----|------|------|------|
| 0x002C | 31           | 30       | 29 | 28 | 27 | 26     | 25 | 24 | 23 | 22       | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10     | 9    | 8 | 7        | 6  | 5    | 4    | 3 | 2        | 1  | 0    |      |      |
| Reset  | 0x00000000   |          |    |    |    |        |    |    |    |          |    |    |    |    |    |    |    |    |    |    |    |        |      |   |          |    |      |      |   |          |    |      |      |      |
| Name   | RTWKFR       | Reserved |    |    |    | RTWKPT |    |    |    | Reserved |    |    |    |    |    |    |    |    |    |    |    | RTWKTR | GLBU |   | Reserved |    | WKFR | MPFR |   | Reserved |    | WKEN | MPEN | PWDN |
| Access | RW           |          |    |    |    | RO     |    |    |    |          |    |    |    |    |    |    |    |    |    |    |    | RW     | RW   |   |          | RO | RO   |      |   | RW       | RW | RW   |      |      |

| Bit   | Name     | Function                                              | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | Access |
|-------|----------|-------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31    | RTWKFR   | Remote wake-up frame filter register pointer reset    | <ul style="list-style-type: none"> <li>0b0: The pointer of the remote wakeup frame filter register is not reset</li> <li>0b1: The pointer of the remote wakeup frame filter register is reset</li> </ul> <div>  <b>Note</b><br/>           After the reset is completed, this bit is automatically cleared.         </div>                                                                                                                                                                                                                                                                                                                                         | RW     |
| 30:27 | Reserved | Reserved                                              | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | -      |
| 26:24 | RTWKPT   | Remote wake-up frame filter register pointer position | This bit indicates which register (value 0-7) the pointer of the remote wakeup frame filter register is on                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | RO     |
| 23:11 | Reserved | Reserved                                              | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | -      |
| 10    | RTWKTR   | Remote wake-up frame transmission                     | After this bit is set to 1, the MAC no longer receives packets other than magic packets and remote wakeup frames, and forwards the received remote wakeup frames to the application<br><br><div>  <b>Note</b><br/>           1. This bit automatically becomes 0 after receiving a magic packet or remote wakeup frame; it can also be software reset before receiving a magic packet or remote wakeup frame.<br/>           2. When the GLBU bit, WKEN bit, or MPEN bit of this register is 1, and the PWDN bit is 0 at the same time, this bit must be set to 1.<br/>           3. When the PWDN bit is 1, the setting of this bit is disabled.         </div> | RW     |
| 9     | GLBU     | Global unicast                                        | <ul style="list-style-type: none"> <li>0b0: Treat any unicast frame filtered by the MAC (DAF) address as inactive</li> <li>0b1: Treat any unicast frame filtered by the MAC (DAF) address as a wakeup frame</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | RW     |
| 8:7   | Reserved | Reserved                                              | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | -      |



| Bit | Name            | Function                     | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | Access |
|-----|-----------------|------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 6   | WKFR            | Receive remote wakeup frame  | <ul style="list-style-type: none"> <li>0b0: No remote wakeup frame received</li> <li>0b1: Remote wakeup frame received</li> </ul> <div>  <b>Note</b><br/>           This bit is automatically cleared after being read.         </div>                                                                                                                                                                                                                                                    | RO     |
| 5   | MPFR            | Receive magic packet         | <ul style="list-style-type: none"> <li>0b0: No magic packet received</li> <li>0b1: Magic packet received</li> </ul> <div>  <b>Note</b><br/>           This bit is automatically cleared after being read.         </div>                                                                                                                                                                                                                                                                  | RO     |
| 4:3 | <i>Reserved</i> | <i>Reserved</i>              | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | -      |
| 2   | WKEN            | Remote wake-up frame enabled | <ul style="list-style-type: none"> <li>0b0: Do not receive remote wakeup frames</li> <li>0b1: Receive remote wakeup frames and trigger PMT events</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                               | RW     |
| 1   | MPEN            | Magic packet enabled         | <ul style="list-style-type: none"> <li>0b0: Do not receive magic packets</li> <li>0b1: Receive magic packets and trigger PMT events</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                             | RW     |
| 0   | PWDN            | Power-down mode enabled      | <ul style="list-style-type: none"> <li>0b0: Active mode</li> <li>0b1: Power-down mode</li> </ul> <p>All received frames are discarded. Only when a magic packet or remote wakeup frame is received, this bit is automatically cleared, and the Power-down mode is exited. After this bit is cleared, the received frames is forwarded to the application.</p> <div>  <b>Note</b><br/>           Only when the WKEN bit or MPEN bit is set to 1 can this bit be set to 1.         </div> | RW     |

## 38.4.2.10 ETH\_MAC Remote Wake-Up Frame Filter Register (ETH\_MAC\_RTWKFFR)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0x0028 | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | WKUPFRMFT    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access | RW           |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit  | Name      | Function                          | Description                                                                                                                                                                                  | Access |
|------|-----------|-----------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:0 | WKUPFRMFT | Remote Wake-Up Frame Filter Value | The application performs write/read operations on the remote wakeup frame filter register through this address. For more information, see " <a href="#">Remote Wake-up Frame Filtering</a> " | RW     |

## 38.4.2.11 ETH\_MAC MAC Frame Filter Control Register (ETH\_MAC\_FLTCTLR)

| Offset | Bit Position |          |    |    |    |    |    |    |    |      |      |          |    |    |    |      |          |    |    |    |    |    |     |     |      |     |    |     |     |      |     |     |    |    |    |    |
|--------|--------------|----------|----|----|----|----|----|----|----|------|------|----------|----|----|----|------|----------|----|----|----|----|----|-----|-----|------|-----|----|-----|-----|------|-----|-----|----|----|----|----|
| 0x0004 | 31           | 30       | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22   | 21   | 20       | 19 | 18 | 17 | 16   | 15       | 14 | 13 | 12 | 11 | 10 | 9   | 8   | 7    | 6   | 5  | 4   | 3   | 2    | 1   | 0   |    |    |    |    |
| Reset  | 0x00000000   |          |    |    |    |    |    |    |    |      |      |          |    |    |    |      |          |    |    |    |    |    |     |     |      |     |    |     |     |      |     |     |    |    |    |    |
| Name   | RA           | Reserved |    |    |    |    |    |    |    | DNTU | IPFE | Reserved |    |    |    | VTFE | Reserved |    |    |    |    |    | HPF | SAF | SAIF | PCF |    | DBF | PMF | DAIF | HMC | HUC | PR |    |    |    |
| Access | RW           |          |    |    |    |    |    |    |    | RW   | RW   |          |    |    |    | RW   |          |    |    |    |    |    | RW  | RW  | RW   | RW  | RW | RW  | RW  | RW   | RW  | RW  | RW | RW | RW | RW |

| Bit   | Name     | Function                                 | Description                                                                                                                                                                                                                                                                                                                                               | Access |
|-------|----------|------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31    | RA       | Receive all                              | <ul style="list-style-type: none"> <li>0b0: The MAC receiver transmits the received frames through address filtering to the application</li> <li>0b1: The MAC receiver transmits all received frames to the application, regardless of whether they have passed address filtering</li> </ul>                                                              | RW     |
| 30:22 | Reserved | Reserved                                 | -                                                                                                                                                                                                                                                                                                                                                         | -      |
| 21    | DNTU     | Discard messages without TCP/UDP headers | <ul style="list-style-type: none"> <li>0b0: The MAC receiver forwards all received IP messages to the application, regardless of whether they have TCP/UDP fields.</li> <li>0b1: The MAC receiver discards all IP messages without TCP or UDP fields.</li> </ul>                                                                                          | RW     |
| 20    | IPFE     | L3&L4 filter enabled                     | <ul style="list-style-type: none"> <li>0b0: The MAC receiver forwards all received IP messages to the application, regardless of whether they have passed L3&amp;L4 filtering.</li> <li>0b1: The MAC receiver discards all IP messages that do not match the L3&amp;L4 filter settings and only processes those that pass L3&amp;L4 filtering.</li> </ul> | RW     |
| 19:17 | Reserved | Reserved                                 | -                                                                                                                                                                                                                                                                                                                                                         | -      |
| 16    | VTFE     | VLAN tag filter enabled                  | <ul style="list-style-type: none"> <li>0b0: The MAC receiver forwards all received VLAN frames to the application, regardless of whether they have passed VLAN tag filtering.</li> <li>0b1: The MAC receiver discards VLAN frames that do not match the VLAN tag filter settings.</li> </ul>                                                              | RW     |
| 15:11 | Reserved | Reserved                                 | -                                                                                                                                                                                                                                                                                                                                                         | -      |
| 10    | HPF      | Hash filter or perfect filter selection  | <ul style="list-style-type: none"> <li>0b0: If the HUC bit or HMC bit of this register is set to 1, the address filter only forwards frames that match the Hash filter.</li> <li>0b1: If the HUC bit or HMC bit of this register is set to 1, the address filter forwards frames that match either the perfect filter or the Hash filter.</li> </ul>      | RW     |

| Bit | Name | Function                          | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | Access |
|-----|------|-----------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 9   | SAF  | Source address filtering          | <ul style="list-style-type: none"> <li>0b0: The MAC receiver forwards all received frames to the application, regardless of whether they have passed source address filtering.</li> <li>0b1: The MAC receiver discards frames that do not match the source address filter settings.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | RW     |
| 8   | SAIF | Source address inverted filtering | <ul style="list-style-type: none"> <li>0b0: When the SA field of a frame received by the MAC receiver does not match the filter settings, it is considered a filter failure.</li> <li>0b1: When the SA field of a frame received by the MAC receiver matches the filter settings, it is considered a filter failure.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | RW     |
| 7:6 | PCF  | Transmit control frame            | <p>For control frames (including unicast pause frames and multicast pause frames), the forwarding operations are as follows:</p> <ul style="list-style-type: none"> <li>0b00: The MAC prevents all control frames from reaching the application.</li> <li>0b01: The MAC forwards all control frames except pause control frames to the application, even if their address filtering fails.</li> <li>0b10: The MAC forwards all control frames to the application, even if their address filtering fails.</li> <li>0b11: The MAC forwards control frames that pass address filtering.</li> </ul> <div>  <b>Note</b><br/> Notes on pause control frames: <ol style="list-style-type: none"> <li>In full-duplex mode, the handling of pause control frames is determined by the RFE bit of the flow control register (ETH_MAC_FLOCTL).</li> <li>When the UNP bit of the flow control register (ETH_MAC_FLOCTL) is set to 1, a received frame is considered a unicast pause frame if its destination address matches the value set in MAC address register 0.</li> <li>A received frame is considered a pause control frame if its TYPE field is 0x8808 or its OPCODE field is 0x0001.</li> </ol> </div> | RW     |
| 5   | DBF  | Broadcast frame mask              | <ul style="list-style-type: none"> <li>0b0: The address filter does not filter received broadcast frames.</li> <li>0b1: The address filter filters all incoming broadcast frames.</li> </ul> <div>  <b>Note</b><br/> When this bit is set to 1, its priority is higher than other filter settings. </div>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | RW     |

| Bit | Name | Function                               | Description                                                                                                                                                                                                                                                                                                                                                                                                                                 | Access |
|-----|------|----------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 4   | PMF  | Forward all multicast frames           | <ul style="list-style-type: none"> <li>0b0: Address filtering is performed on received frames with a multicast destination address (the first bit of the destination address field is 1), and the filtering method is determined by the HMC bit of this register.</li> <li>0b1: No address filtering is performed on received frames with a multicast destination address (the first bit of the destination address field is 1).</li> </ul> | RW     |
| 3   | DAIF | Destination address inverted filtering | <ul style="list-style-type: none"> <li>0b0: When the destination address of a unicast or multicast frame received by the MAC receiver does not match the filter settings, it is considered a filter failure.</li> <li>0b1: When the destination address of a unicast or multicast frame received by the MAC receiver matches the filter settings, it is considered a filter failure.</li> </ul>                                             | RW     |
| 2   | HMC  | Hash multicast control                 | <ul style="list-style-type: none"> <li>0b0: The MAC performs perfect destination address filtering on multicast frames, comparing the DA field with the set filter values.</li> <li>0b1: The MAC performs Hash destination address filtering on multicast frames.</li> </ul>                                                                                                                                                                | RW     |
| 1   | HUC  | Hash unicast control                   | <ul style="list-style-type: none"> <li>0b0: The MAC performs perfect destination address filtering on unicast frames, comparing the DA field with the set filter values.</li> <li>0b1: The MAC performs Hash destination address filtering on unicast frames.</li> </ul>                                                                                                                                                                    | RW     |
| 0   | PR   | Mixed mode                             | <ul style="list-style-type: none"> <li>0b0: The address filter performs normal address filtering.</li> <li>0b1: The address filter forwards all incoming frames regardless of destination or source address.</li> </ul>                                                                                                                                                                                                                     | RW     |

## 38.4.2.12 ETH\_MAC MAC Address High Bit Register 0 (ETH\_MAC\_MACADHR0)

| Offset | Bit Position |          |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |          |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |
|--------|--------------|----------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----------|----|----|----|----|---|---|---|---|---|---|---|---|---|---|--|
| 0x0040 | 31           | 30       | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14       | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |  |
| Reset  | 0x8000FFFF   |          |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |          |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |
| Name   |              | Reserved |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | ASDDRH 0 |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |
|        | AE0          |          |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |          |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |
| Access | RO           |          |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RW       |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |

| Bit   | Name     | Function                 | Description                                                                                                                                                                                                                                                                 | Access |
|-------|----------|--------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31    | AE0      | Address enabled 0        | <ul style="list-style-type: none"> <li>0b1: Use MAC address 0 for perfect destination address filtering.</li> </ul> <div>  <b>Note</b><br/>           This bit is always 1.         </div> | RO     |
| 30:16 | Reserved | Reserved                 | -                                                                                                                                                                                                                                                                           | -      |
| 15:0  | ASDDRH 0 | High bit address value 0 | The MAC uses this field to filter received frames and insert MAC addresses into transmitted flow control (pause) frames.                                                                                                                                                    | RW     |

## 38.4.2.13 ETH\_MAC MAC Address Low Bit Register 0 (ETH\_MAC\_MACADLR0)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0x0044 | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0xFFFFFFFF   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | ADDRLO       |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access | RW           |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit  | Name   | Function                | Description                                                                                                              | Access |
|------|--------|-------------------------|--------------------------------------------------------------------------------------------------------------------------|--------|
| 31:0 | ADDRLO | Low bit address value 0 | The MAC uses this field to filter received frames and insert MAC addresses into transmitted flow control (pause) frames. | RW     |

## 38.4.2.14 ETH\_MAC MAC Address High Bit Register 1 (ETH\_MAC\_MACADHR1)

| Offset | Bit Position |     |      |    |    |    |    |    |          |    |    |    |    |    |    |    |         |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |  |
|--------|--------------|-----|------|----|----|----|----|----|----------|----|----|----|----|----|----|----|---------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|--|--|
| 0x0048 | 31           | 30  | 29   | 28 | 27 | 26 | 25 | 24 | 23       | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15      | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |  |  |
| Reset  | 0x0000FFFF   |     |      |    |    |    |    |    |          |    |    |    |    |    |    |    |         |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |  |
| Name   |              |     | MBC1 |    |    |    |    |    | Reserved |    |    |    |    |    |    |    | ASDDRH1 |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |  |
|        | AE1          | SA1 |      |    |    |    |    |    |          |    |    |    |    |    |    |    |         |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |  |
| Access | RW           | RW  | RW   |    |    |    |    |    |          |    |    |    |    |    |    |    | RW      |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |  |

| Bit   | Name     | Function                 | Description                                                                                                                                                                                                                                                                                                                         | Access |
|-------|----------|--------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31    | AE1      | Address enabled 1        | <ul style="list-style-type: none"> <li>0b0: Ignore address filtering for MAC address 1.</li> <li>0b1: Use MAC address 1 for perfect filtering.</li> </ul>                                                                                                                                                                           | RW     |
| 30    | SA1      | Address selection 1      | <ul style="list-style-type: none"> <li>0b0: Use MAC address 1 to filter the DA of received frames.</li> <li>0b1: Use MAC address 1 to filter the SA of received frames.</li> </ul>                                                                                                                                                  | RW     |
| 29:24 | MBC1     | Byte masking 1           | Mask the corresponding bytes during address filtering. <ul style="list-style-type: none"> <li>Bit29: Mask ADDR1[15:8] bits</li> <li>Bit28: Mask ADDR1[7:0] bits</li> <li>Bit27: Mask ADDR1[31:24] bits</li> <li>Bit26: Mask ADDR1[23:16] bits</li> <li>Bit25: Mask ADDR1[15:8] bits</li> <li>Bit24: Mask ADDR1[7:0] bits</li> </ul> | RW     |
| 23:16 | Reserved | Reserved                 | -                                                                                                                                                                                                                                                                                                                                   | -      |
| 15:0  | ASDDRH1  | High bit address value 1 | The MAC uses this field to filter received frames.                                                                                                                                                                                                                                                                                  | RW     |

## 38.4.2.15 ETH\_MAC MAC Address Low Bit Register 1 (ETH\_MAC\_MACADLR1)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0x004C | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0xFFFFFFFF   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | ADDR1        |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access | RW           |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit  | Name  | Function                | Description                                        | Access |
|------|-------|-------------------------|----------------------------------------------------|--------|
| 31:0 | ADDR1 | Low bit address value 1 | The MAC uses this field to filter received frames. | RW     |

## 38.4.2.16 ETH\_MAC MAC Address High Bit Register 2 (ETH\_MAC\_MACADHR2)

| Offset | Bit Position |     |      |    |    |    |    |    |          |    |    |    |    |    |    |    |        |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |  |
|--------|--------------|-----|------|----|----|----|----|----|----------|----|----|----|----|----|----|----|--------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|--|--|
| 0x0050 | 31           | 30  | 29   | 28 | 27 | 26 | 25 | 24 | 23       | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15     | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |  |  |
| Reset  | 0x0000FFFF   |     |      |    |    |    |    |    |          |    |    |    |    |    |    |    |        |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |  |
| Name   |              |     | MBC2 |    |    |    |    |    | Reserved |    |    |    |    |    |    |    | ASDDR2 |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |  |
|        | AE2          | SA2 |      |    |    |    |    |    |          |    |    |    |    |    |    |    |        |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |  |
| Access | RW           | RW  | RW   |    |    |    |    |    |          |    |    |    |    |    |    |    | RW     |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |  |

| Bit   | Name     | Function             | Description                                                                                                                                                                                                                                                                                                                             | Access |
|-------|----------|----------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31    | AE2      | Address enabled 2    | <ul style="list-style-type: none"> <li>0b0: Ignore address filtering for MAC address 2.</li> <li>0b1: Use MAC address 2 for perfect filtering.</li> </ul>                                                                                                                                                                               | RW     |
| 30    | SA2      | Address selection 2  | <ul style="list-style-type: none"> <li>0b0: Use MAC address 2 to filter the DA of received frames.</li> <li>0b1: Use MAC address 2 to filter the SA of received frames.</li> </ul>                                                                                                                                                      | RW     |
| 29:24 | MBC2     | Byte masking 2       | Mask the corresponding bytes during address filtering. <ul style="list-style-type: none"> <li>Bit29: Mask ADDR2[15:8] bits</li> <li>Bit28: Mask ADDR2[7:0] bits</li> <li>Bit27: Mask ADDRL2[31:24] bits</li> <li>Bit26: Mask ADDRL2[23:16] bits</li> <li>Bit25: Mask ADDRL2[15:8] bits</li> <li>Bit24: Mask ADDRL2[7:0] bits</li> </ul> | RW     |
| 23:16 | Reserved | Reserved             | -                                                                                                                                                                                                                                                                                                                                       | -      |
| 15:0  | ASDDR2   | High address value 2 | The MAC uses this field to filter received frames.                                                                                                                                                                                                                                                                                      | RW     |

## 38.4.2.17 ETH\_MAC MAC Address Low Bit Register 2 (ETH\_MAC\_MACADLR2)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0x0054 | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0xFFFFFFFF   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | ADDR2        |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access | RW           |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit  | Name  | Function                | Description                                        | Access |
|------|-------|-------------------------|----------------------------------------------------|--------|
| 31:0 | ADDR2 | Low bit address value 2 | The MAC uses this field to filter received frames. | RW     |



## 38.4.2.18 ETH\_MAC MAC Address High Bit Register 3 (ETH\_MAC\_MACADHR3)

| Offset | Bit Position |     |    |    |      |    |    |    |          |    |    |    |    |    |    |    |         |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |  |
|--------|--------------|-----|----|----|------|----|----|----|----------|----|----|----|----|----|----|----|---------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|--|--|
| 0x0058 | 31           | 30  | 29 | 28 | 27   | 26 | 25 | 24 | 23       | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15      | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |  |  |
| Reset  | 0x0000FFFF   |     |    |    |      |    |    |    |          |    |    |    |    |    |    |    |         |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |  |
| Name   |              |     |    |    | MBC3 |    |    |    | Reserved |    |    |    |    |    |    |    | ASDDRH3 |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |  |
|        | AE3          | SA3 |    |    |      |    |    |    |          |    |    |    |    |    |    |    |         |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |  |
| Access | RW           | RW  | RW |    |      |    |    |    |          |    |    |    |    |    | RW |    |         |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |  |

| Bit   | Name     | Function                 | Description                                                                                                                                                                                                                                                                                                                         | Access |
|-------|----------|--------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31    | AE3      | Address enabled 3        | <ul style="list-style-type: none"> <li>0b0: Ignore address filtering for MAC address3.</li> <li>0b1: Use MAC address 3 for perfect filtering.</li> </ul>                                                                                                                                                                            | RW     |
| 30    | SA3      | Address selection3       | <ul style="list-style-type: none"> <li>0b0: Use MAC address 3 to filter the DA of received frames.</li> <li>0b1: Use MAC address 3 to filter the SA of received frames.</li> </ul>                                                                                                                                                  | RW     |
| 29:24 | MBC3     | Byte masking 3           | Mask the corresponding bytes during address filtering. <ul style="list-style-type: none"> <li>Bit29: Mask ADDR3[15:8] bits</li> <li>Bit28: Mask ADDR3[7:0] bits</li> <li>Bit27: Mask ADDR3[31:24] bits</li> <li>Bit26: Mask ADDR3[23:16] bits</li> <li>Bit25: Mask ADDR3[15:8] bits</li> <li>Bit24: Mask ADDR3[7:0] bits</li> </ul> | RW     |
| 23:16 | Reserved | Reserved                 | -                                                                                                                                                                                                                                                                                                                                   | -      |
| 15:0  | ASDDRH3  | High bit address value 3 | The MAC uses this field to filter received frames.                                                                                                                                                                                                                                                                                  | RW     |

## 38.4.2.19 ETH\_MAC MAC Address Low Bit Register 3 (ETH\_MAC\_MACADLR3)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0x005C | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0xFFFFFFFF   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | ADDR3        |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access | RW           |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit  | Name  | Function                | Description                                        | Access |
|------|-------|-------------------------|----------------------------------------------------|--------|
| 31:0 | ADDR3 | Low bit address value 3 | The MAC uses this field to filter received frames. | RW     |

## 38.4.2.20 ETH\_MAC MAC Address High Bit Register 4 (ETH\_MAC\_MACADHR4)

| Offset | Bit Position |     |      |    |    |    |    |    |          |    |    |    |    |    |    |    |         |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |  |
|--------|--------------|-----|------|----|----|----|----|----|----------|----|----|----|----|----|----|----|---------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|--|--|
| 0x0060 | 31           | 30  | 29   | 28 | 27 | 26 | 25 | 24 | 23       | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15      | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |  |  |
| Reset  | 0x0000FFFF   |     |      |    |    |    |    |    |          |    |    |    |    |    |    |    |         |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |  |
| Name   |              |     | MBC4 |    |    |    |    |    | Reserved |    |    |    |    |    |    |    | ASDDRH4 |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |  |
|        | AE4          | SA4 |      |    |    |    |    |    |          |    |    |    |    |    |    |    |         |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |  |
| Access | RW           | RW  | RW   |    |    |    |    |    |          |    |    |    |    |    |    |    | RW      |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |  |

| Bit   | Name     | Function                 | Description                                                                                                                                                                                                                                                                                                                         | Access |
|-------|----------|--------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31    | AE4      | Address enabled 4        | <ul style="list-style-type: none"> <li>0b0: Ignore address filtering for MAC address4.</li> <li>0b1: Use MAC address 4 for perfect filtering.</li> </ul>                                                                                                                                                                            | RW     |
| 30    | SA4      | Address selection 4      | <ul style="list-style-type: none"> <li>0b0: Use MAC address 4 to filter the DA of received frames.</li> <li>0b1: Use MAC address 4 to filter the SA of received frames.</li> </ul>                                                                                                                                                  | RW     |
| 29:24 | MBC4     | Byte masking 4           | Mask the corresponding bytes during address filtering. <ul style="list-style-type: none"> <li>Bit29: Mask ADDR4[15:8] bits</li> <li>Bit28: Mask ADDR4[7:0] bits</li> <li>Bit27: Mask ADDR4[31:24] bits</li> <li>Bit26: Mask ADDR4[23:16] bits</li> <li>Bit25: Mask ADDR4[15:8] bits</li> <li>Bit24: Mask ADDR4[7:0] bits</li> </ul> | RW     |
| 23:16 | Reserved | Reserved                 | -                                                                                                                                                                                                                                                                                                                                   | -      |
| 15:0  | ASDDRH4  | High bit address value 4 | The MAC uses this field to filter received frames.                                                                                                                                                                                                                                                                                  | RW     |

## 38.4.2.21 ETH\_MAC MAC Address Low Bit Register 4 (ETH\_MAC\_MACADLR4)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0x0064 | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0xFFFFFFFF   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | ADDR4        |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access | RW           |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit  | Name  | Function               | Description                                        | Access |
|------|-------|------------------------|----------------------------------------------------|--------|
| 31:0 | ADDR4 | Low bit address value4 | The MAC uses this field to filter received frames. | RW     |

## 38.4.2.22 ETH\_MAC MAC Hash Table High Bit Register (ETH\_MAC\_HASHTHR)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0x0008 | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | HTH          |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access | RW           |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit  | Name | Function                  | Description                     | Access |
|------|------|---------------------------|---------------------------------|--------|
| 31:0 | HTH  | Hash Table high bit value | High 32-bit value of Hash Table | RW     |

## 38.4.2.23 ETH\_MAC MAC Hash Table Low Bit Register (ETH\_MAC\_HASHTLR)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0x000C | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | HTL          |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access | RW           |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit  | Name | Function                 | Description                    | Access |
|------|------|--------------------------|--------------------------------|--------|
| 31:0 | HTL  | Hash Table low bit value | Low 32-bit value of Hash Table | RW     |

## 38.4.2.24 ETH\_MAC VLAN Tag Transmit Control Register (ETH\_MAC\_VTACTLR)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |       |       |       |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|-------|-------|-------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0x0584 | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17    | 16    | 15    | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |       |       |       |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    | VLANS | VLANC | VLANV |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    | RW    | RW    | RW    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit   | Name     | Function           | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | Access |
|-------|----------|--------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:19 | Reserved | Reserved           | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | -      |
| 18    | VLANS    | VLAN Tag Selection | <ul style="list-style-type: none"> <li>0b0: When transmitting VLAN frames, the handling of the tag field is determined by the VLANC control bit setting in TDES0 of the Tx descriptor.</li> <li>0b1: When transmitting VLAN frames, the handling of the tag field is determined by the VLANC control bit setting in this register.</li> </ul>                                                                                                                                                                           | RW     |
| 17:16 | VLANC    | VLAN Tag Control   | Handling of the tag field when transmitting VLAN frames: <ul style="list-style-type: none"> <li>0b00: No operation, transmit directly.</li> <li>0b01: Remove the TYPE field and tag field of the VLAN frame.</li> <li>0b10: After inserting the TYPE field, insert the value of the VLANV bit of this register as the tag field, regardless of whether the original frame contains a tag field.</li> <li>0b11: Replace the tag field in the original frame with the value of the VLANV bit of this register.</li> </ul> | RW     |
| 15:0  | VLANV    | VLAN Tag Value     | User-defined VLAN tag field value, used to insert or replace the tag field when transmitting frames.                                                                                                                                                                                                                                                                                                                                                                                                                    | RW     |

## 38.4.2.25 ETH\_MAC VLAN Tag Receive Filter Register (ETH\_MAC\_VTAFLTR)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |      |          |      |      |       |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|------|----------|------|------|-------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0x001C | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19   | 18       | 17   | 16   | 15    | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |      |          |      |      |       |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    | VTHM | Reserved | VTIM | VTAL | VLFLT |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access |              |    |    |    |    |    |    |    |    |    |    |    | RW   |          | RW   | RW   | RW    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit   | Name     | Function                     | Description                                                                                                                                                                                                                                                                                                                                                                              | Access |
|-------|----------|------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:20 | Reserved | Reserved                     | -                                                                                                                                                                                                                                                                                                                                                                                        | -      |
| 19    | VTHM     | VLAN Tag Hash Table Enabled  | <ul style="list-style-type: none"> <li>0b0: VLAN tag Hash filtering function is disabled.</li> <li>0b1: VLAN tag Hash filtering function is enabled.</li> </ul>                                                                                                                                                                                                                          | RW     |
| 18    | Reserved | Reserved                     | -                                                                                                                                                                                                                                                                                                                                                                                        | -      |
| 17    | VTIM     | VLAN Tag Inversion Filtering | <ul style="list-style-type: none"> <li>0b0: When the tag field of a VLAN frame does not match the filter value set by the VLFLT bit of this register, it is considered a filter failure.</li> <li>0b1: When the tag field of a VLAN frame matches the filter value set by the VLFLT bit of this register, it is considered a filter failure.</li> </ul>                                  | RW     |
| 16    | VTAL     | 12-bit VLAN Compare          | <ul style="list-style-type: none"> <li>0b0: Compare all 16 bits of the VLAN tag field with the VLFLT setting.</li> <li>0b1: Compare the lower 12 bits of the VLAN tag field with the VLFLT setting.</li> </ul> <div>  <b>Note</b><br/> The setting of this bit also applies to Hash filtering. </div> | RW     |
| 15:0  | VLFLT    | VLAN Tag Filter Value        | This field contains the comparison value for filtering VLAN tag fields, and this bit is compared with the 15th and 16th bytes of received VLAN frames.                                                                                                                                                                                                                                   | RW     |

38.4.2.26 ETH\_MAC VLAN Hash Table Register (ETH\_MAC\_VLAHTBR)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |      |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0x0588 | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15   | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |      |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | VLHT |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RW   |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit   | Name     | Function        | Description           | Access |
|-------|----------|-----------------|-----------------------|--------|
| 31:16 | Reserved | Reserved        | -                     | -      |
| 15:0  | VLHT     | VLAN Hash Table | VLAN Hash Table value | RW     |

## 38.4.2.27 ETH\_MAC LAY3LAY4 Control Register (ETH\_MAC\_L34CTLR)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |        |       |        |       |          |       |        |    |    |    |    |        |   |   |   |   |        |       |        |       |          |       |    |
|--------|--------------|----|----|----|----|----|----|----|----|----|--------|-------|--------|-------|----------|-------|--------|----|----|----|----|--------|---|---|---|---|--------|-------|--------|-------|----------|-------|----|
| 0x0400 | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21     | 20    | 19     | 18    | 17       | 16    | 15     | 14 | 13 | 12 | 11 | 10     | 9 | 8 | 7 | 6 | 5      | 4     | 3      | 2     | 1        | 0     |    |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |        |       |        |       |          |       |        |    |    |    |    |        |   |   |   |   |        |       |        |       |          |       |    |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    | L4DPIM | L4DPM | L4SPIM | L4SPM | Reserved | L4PEN | L3HDBM |    |    |    |    | L3HSBM |   |   |   |   | L3DAIM | L3DAM | L3SAIM | L3SAM | Reserved | L3PEN |    |
| Access |              |    |    |    |    |    |    |    |    |    | RW     | RW    | RW     | RW    |          | RW    | RW     |    |    |    |    | RW     |   |   |   |   | L3DAIM | RW    | RW     | RW    | RW       |       | RW |

| Bit   | Name     | Function                               | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | Access |
|-------|----------|----------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:22 | Reserved | Reserved                               | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | -      |
| 21    | L4DPIM   | L4 Destination Port Inverted Filtering | <ul style="list-style-type: none"> <li>0b0: When the L4 destination port field does not match the filter settings, it is considered a filter failure.</li> <li>0b1: When the L4 destination port field matches the filter settings, it is considered a filter failure.</li> </ul> <div>  <b>Note</b><br/>           This bit setting is only active when the L4DPM bit of this register is enable.         </div> | RW     |
| 20    | L4DPM    | L4 Destination Port Filtering          | <ul style="list-style-type: none"> <li>0b0: Ignore filtering for the L4 destination port field.</li> <li>0b1: Filter the L4 destination port field.</li> </ul>                                                                                                                                                                                                                                                                                                                                      | RW     |
| 19    | L4SPIM   | L4 Source Port Inverted Filtering      | <ul style="list-style-type: none"> <li>0b0: When the L4 source port field does not match the filter settings, it is considered a filter failure.</li> <li>0b1: When the L4 source port field matches the filter settings, it is considered a filter failure.</li> </ul> <div>  <b>Note</b><br/>           This bit setting is only active when the L4SPM bit of this register is enabled.         </div>         | RW     |
| 18    | L4SPM    | L4 Source Port Filtering               | <ul style="list-style-type: none"> <li>0b0: Ignore filtering for the L4 source port field.</li> <li>0b1: Filter the L4 source port field.</li> </ul>                                                                                                                                                                                                                                                                                                                                                | RW     |
| 17    | Reserved | Reserved                               | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | -      |
| 16    | L4PEN    | L4 Protocol Enabled                    | <ul style="list-style-type: none"> <li>0b0: Perform port filtering on L4 TCP protocol frames.</li> <li>0b1: Perform port filtering on L4 UDP protocol frames.</li> </ul>                                                                                                                                                                                                                                                                                                                            | RW     |
| 15:11 | L3HDBM   | L3IP Destination Address High Bit Mask | Bit mask settings for L3 destination address filtering:<br><b>For IPv4:</b> <ul style="list-style-type: none"> <li>0b00000: Filter the entire destination address.</li> </ul>                                                                                                                                                                                                                                                                                                                       | RW     |

| Bit   | Name   | Function                               | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | Access |
|-------|--------|----------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 15:11 | L3HDBM | L3IP Destination Address High Bit Mask | <ul style="list-style-type: none"> <li>● 0b00001: Ignore filtering for the LSB[0] bit of the destination address.</li> <li>● 0b00010: Ignore filtering for the LSB[1:0] bits of the destination address.</li> <li>● 0b00011: Ignore filtering for the LSB[2:0] bits of the destination address.</li> <li>● .....</li> <li>● 0b11111: Ignore filtering for the LSB[30:0] bits of the destination address.</li> </ul> <p><b>For IPv6:</b></p> <ul style="list-style-type: none"> <li>● Bit2-1 of L3HDBM is used as Bit7-6 of L3HSBM</li> </ul> <div>  <b>Note</b><br/>           This bit is only active when the L3DAM of this register is enabled.         </div>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | RW     |
| 10:6  | L3HSBM | L3IP Source Address High Bit Mask      | <p>Bit mask settings for source address filtering:</p> <p><b>For IPv4:</b></p> <ul style="list-style-type: none"> <li>● 0b00000: Filter the entire source address.</li> <li>● 0b00001: Ignore filtering for the LSB[0] bit of the source address.</li> <li>● 0b00010: Ignore filtering for the LSB[1:0] bits of the source address.</li> <li>● 0b00011: Ignore filtering for the LSB[2:0] bits of the source address.</li> <li>● .....</li> <li>● 0b11111: Ignore filtering for the LSB[30:0] bits of the source address.</li> </ul> <p><b>For IPv6:</b></p> <ul style="list-style-type: none"> <li>● 0b0000000: Filter the entire source or destination address.</li> <li>● 0b0000001: Ignore filtering for the LSB[0] bit of the source or destination address.</li> <li>● 0b0000010: Ignore the filtering of the least significant bit LSB[1:0] of the source or destination address</li> <li>● 0b0000011: Ignore the filtering of the least significant bit LSB[2:0] of the source or destination address</li> <li>● .....</li> <li>● 0b1111111: Ignore the filtering of the least significant bit LSB[126:0] of the source or destination address</li> </ul> <div>  <b>Note</b><br/>           1. This bit is only active when the L3SAM or L3DAM of this register is enabled.<br/>           2. When IPv6 functionality is selected and both L3DAM and L3SAM are enabled, the source and destination address filtering controls share a set of filter addresses, and the high bit mask control bit is also shared.         </div> | RW     |



| Bit | Name            | Function                                  | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | Access |
|-----|-----------------|-------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 5   | L3DAIM          | L3 Destination Address Inverted Filtering | <ul style="list-style-type: none"> <li>0b0: When the destination address field of an L3 IP packet does not match the filter settings, it is considered a filter failure.</li> <li>0b1: When the destination address field of an L3 IP packet matches the filter settings, it is considered a filter failure.</li> </ul> <div>  <b>Note</b><br/>           This bit setting is only active when the L3DAM bit of this register is enabled.         </div> | RW     |
| 4   | L3DAM           | L3 Destination Address Filtering          | <ul style="list-style-type: none"> <li>0b0: Ignore filtering for the destination address field of L3 IP packets.</li> <li>0b1: Filter the destination address field of L3 IP packets.</li> </ul>                                                                                                                                                                                                                                                                                                                                          | RW     |
| 3   | L3SAIM          | L3 Source Address Inverted Filtering      | <ul style="list-style-type: none"> <li>0b0: When the source address field of an L3 IP packet does not match the filter settings, it is considered a filter failure.</li> <li>0b1: When the source address field of an L3 IP packet matches the filter settings, it is considered a filter failure.</li> </ul> <div>  <b>Note</b><br/>           This bit setting is only active when the L3SAM bit of this register is enabled.         </div>          | RW     |
| 2   | L3SAM           | L3 Source Address Filtering               | <ul style="list-style-type: none"> <li>0b0: Ignore filtering for the source address field of L3 IP packets.</li> <li>0b1: Filter the source address field of L3 IP packets.</li> </ul>                                                                                                                                                                                                                                                                                                                                                    | RW     |
| 1   | <i>Reserved</i> | <i>Reserved</i>                           | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | -      |
| 0   | L3PEN           | L3 Protocol Enabled                       | <ul style="list-style-type: none"> <li>0b0: Perform address filtering on L3 IPv4 packets.</li> <li>0b1: Perform address filtering on L3 IPv6 packets.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                          | RW     |

## 38.4.2.28 ETH\_MAC LAY4 Port Register (ETH\_MAC\_L4PORTR)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |         |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0x0404 | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15      | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |         |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | L4DPVAL      |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | L4SPVAL |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access | RW           |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RW      |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit   | Name    | Function                      | Description                      | Access |
|-------|---------|-------------------------------|----------------------------------|--------|
| 31:16 | L4DPVAL | Destination Port Filter Value | L4 destination port filter value | RW     |
| 15:0  | L4SPVAL | Source Port Filter Value      | L4 source port filter value      | RW     |

## 38.4.2.29 ETH\_MAC LAY3 Address Register 0 (ETH\_MAC\_L3ADDR0)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0x0410 | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | L3ADDR 0     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access | RW           |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit  | Name    | Function               | Description                                                                                                                                                                                                                                                                                                                                                                                                                              | Access |
|------|---------|------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:0 | L3ADDR0 | Address Filter Value 0 | L3 Address Filter Value: <ul style="list-style-type: none"> <li>When L3PEN=0 and L3SAM=1, this register serves as the L3 source address filter value for IPv4 packets.</li> <li>When L3PEN=1 and L3SAM=1, this register serves as the L3 source address [31:0] filter value for IPv6 packets.</li> <li>When L3PEN=1 and L3DAM=1, this register serves as the L3 destination address [31:0] bit filter value for IPv6 packets.</li> </ul> | RW     |

## 38.4.2.30 ETH\_MAC LAY3 Address Register 1 (ETH\_MAC\_L3ADDRR1)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0x0414 | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | L3ADDR1      |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access | RW           |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit  | Name    | Function               | Description                                                                                                                                                                                                                                                                                                                                                                                                                                     | Access |
|------|---------|------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:0 | L3ADDR1 | Address Filter Value 1 | L3 Address Filter Value: <ul style="list-style-type: none"> <li>When L3PEN=0 and L3DAM=1, this register serves as the L3 destination address filter value for IPv4 packets.</li> <li>When L3PEN=1 and L3SAM=1, this register serves as the L3 source address [63:32] filter value for IPv6 packets.</li> <li>When L3PEN=1 and L3DAM=1, this register serves as the L3 destination address [63:32] bit filter value for IPv6 packets.</li> </ul> | RW     |

## 38.4.2.31 ETH\_MAC LAY3 Address Register 2 (ETH\_MAC\_L3ADDRR2)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0x0418 | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | L3ADDR 2     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access | RW           |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit  | Name    | Function               | Description                                                                                                                                                                                                                                                                                                                | Access |
|------|---------|------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:0 | L3ADDR2 | Address Filter Value 2 | L3 Address Filter Value: <ul style="list-style-type: none"> <li>When L3PEN=1 and L3SAM=1, this register serves as the L3 source address [95:64] filter value for IPv6 packets.</li> <li>When L3PEN=1 and L3DAM=1, this register serves as the L3 destination address [95:64] bit filter value for IPv6 packets.</li> </ul> | RW     |

## 38.4.2.32 ETH\_MAC LAY3 Address Register 3 (ETH\_MAC\_L3ADDRR3)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0x041C | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | L3ADDR 3     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access | RW           |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit  | Name    | Function               | Description                                                                                                                                                                                                                                                                                                                  | Access |
|------|---------|------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:0 | L3ADDR3 | Address Filter Value 3 | L3 Address Filter Value: <ul style="list-style-type: none"> <li>When L3PEN=1 and L3SAM=1, this register serves as the L3 source address [127:96] filter value for IPv6 packets.</li> <li>When L3PEN=1 and L3DAM=1, this register serves as the L3 destination address [127:96] bit filter value for IPv6 packets.</li> </ul> | RW     |

### 38.4.3 ETH\_PTP Register

#### 38.4.3.1 ETH\_PTP Timestamp Control Register (ETH\_PTP\_TSPCTLR)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |        |          |    |    |    |           |           |          |        |        |         |          |         |        |       |        |          |       |    |    |    |    |    |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|--------|----------|----|----|----|-----------|-----------|----------|--------|--------|---------|----------|---------|--------|-------|--------|----------|-------|----|----|----|----|----|
| 0x0700 | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17     | 16       | 15 | 14 | 13 | 12        | 11        | 10       | 9      | 8      | 7       | 6        | 5       | 4      | 3     | 2      | 1        | 0     |    |    |    |    |    |
| Reset  | 0x00002000   |    |    |    |    |    |    |    |    |    |    |    |    |    |        |          |    |    |    |           |           |          |        |        |         |          |         |        |       |        |          |       |    |    |    |    |    |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    | TSPADF | TSPMTSEL |    |    |    | TSPOVIPV4 | TSPOVIPV6 | TSPOVETH | TSPVER | TSPSSR | TSPEALL | Reserved | TSPADUP | TSPINT | TSPUP | TSPINI | TSPUPSEL | TSPEN |    |    |    |    |    |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    | RW     | RW       |    |    |    | RW        | RW        | RW       | RW     | RW     | RW      |          | RW      | RW     | RW    | RW     | RW       | RW    | RW | RW | RW | RW | RW |

| Bit   | Name      | Function                                     | Description                                                                                                                                                                    | Access |
|-------|-----------|----------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:19 | Reserved  | Reserved                                     | -                                                                                                                                                                              | -      |
| 18    | TSPADF    | PTP frame filtering                          | When a PTP frame is transmitted directly over Ethernet, if this bit is set to 1, destination address filtering is performed on the PTP frame.                                  | RW     |
| 17:14 | TSPMTSEL  | PTP message type selection                   | Select the message type for timestamp snapshot. For details, refer to " <a href="#">PTP Message Type</a> ".                                                                    | RW     |
| 13    | TSPOVIPV4 | Receive frames of Over UDP-IPv4 package type | <ul style="list-style-type: none"> <li>0b0: Ignore PTP frames in Over UDP-IPv4 package form.</li> <li>0b1: Receive PTP frames in Over UDP-IPv4 package form.</li> </ul>        | RW     |
| 12    | TSPOVIPV6 | Receive frames of Over UDP-IPv6 package type | <ul style="list-style-type: none"> <li>0b0: Ignore PTP frames in Over UDP-IPv6 package form.</li> <li>0b1: Receive PTP frames in Over UDP-IPv6 package form.</li> </ul>        | RW     |
| 11    | TSPOVETH  | Receive frames of Over Ethernet package type | <ul style="list-style-type: none"> <li>0b0: Ignore PTP frames in Over Ethernet package form.</li> <li>0b1: Receive PTP frames in Over Ethernet package form.</li> </ul>        | RW     |
| 10    | TSPVER    | PTP message processing version               | <ul style="list-style-type: none"> <li>0b0: Process received PTP frames using IEEE1588v1 format.</li> <li>0b1: Process received PTP frames using IEEE1588v2 format.</li> </ul> | RW     |

| Bit | Name     | Function                      | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | Access |
|-----|----------|-------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 9   | TSPSSR   | Subsecond counter granularity | <ul style="list-style-type: none"> <li>0b0: When the subsecond register value reaches 0x7FFFFFFFH, the timestamp second counting increments by 1 second and the subsecond register resets to zero and counts again.</li> <li>0b1: If the value of the subsecond register reaches 0x3B9AC9FFH, the timestamp second register is increased by 1 second and the subsecond register resets to zero and counts again.</li> </ul> <div>  <b>Note</b><br/> 0x3B9AC9FF=999999999<br/> Subsecond increments must be properly programmed according to the PTP reference clock frequency and the value of this bit. </div> | RW     |
| 8   | TSPEALL  | Enable all frames             | When this bit is set to 1, timestamp snapshots are taken of all frames received                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | RW     |
| 7:6 | Reserved | Reserved                      | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | -      |
| 5   | TSPADUP  | Count register update         | When this bit is set to 1, the value of timestamp increment register (ETH_PTP_TSPADDR) is updated to PTP for precise calibration. This bit is automatically cleared when update completes. <div>  <b>Note</b><br/> This register bit must be read as 0 before it can be set to 1. </div>                                                                                                                                                                                                                                                                                                                      | RW     |
| 4   | TSPINT   | Timestamp interrupt enabled   | If system time exceeds the value written in target time register, a timestamp interrupt is generated. <div>  <b>Note</b><br/> When PPS0 generates an interrupt (ETH_PTP_TSPSTSR.TSTAR0=1), this bit is automatically cleared and timestamp interrupt function is disabled. If this interrupt function is still needed, set this bit to 1 in the application. </div>                                                                                                                                                                                                                                           | RW     |
| 3   | TSPUP    | Timestamp update              | When this bit is set to 1, system time is updated (added/subtracted) by the values specified in timestamp update second register (ETH_PTP_TMUSECR) and timestamp update subsecond register (ETH_PTP_TMUNSER). This bit is automatically cleared when update completes. <div>  <b>Note</b><br/> This register bit can only be set to 1 after both itself and TSPINI bits are read as 0. </div>                                                                                                                                                                                                                 | RW     |

| Bit | Name     | Function                   | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | Access |
|-----|----------|----------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 2   | TSPINI   | Timestamp initialization   | <p>When this bit is set to 1, system time is initialized (overwritten) by the values specified in timestamp update second register (ETH_PTP_TMUSECR) and timestamp update subsecond register (ETH_PTP_TMUNSER). This bit is automatically cleared when initialization completes.</p> <div>  <b>Note</b><br/>           This register bit must be read as 0 before it can be set to 1.         </div> | RW     |
| 1   | TSPUPSEL | Timestamp calibration mode | <ul style="list-style-type: none"> <li>● 0b0: Coarse calibration mode</li> <li>● 0b1: Fine calibration mode</li> </ul>                                                                                                                                                                                                                                                                                                                                                                | RW     |
| 0   | TSPEN    | PTP function enabled       | <ul style="list-style-type: none"> <li>● 0b0: Timestamp function disabled, MAC won't add timestamps to transmitted or received frames.</li> <li>● 0b1: Timestamp function enabled, timestamps is added to transmitted and received frames.</li> </ul> <div>  <b>Note</b><br/>           Once this bit is set to 1, system time needs to be initialized.         </div>                               | RW     |

## 38.4.3.2 ETH\_PTP Timestamp Status Register (ETH\_PTP\_TSPSTSR)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |        |        |        |          |        |       |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|--------|--------|--------|----------|--------|-------|
| 0x0728 | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5      | 4      | 3      | 2        | 1      | 0     |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |        |        |        |          |        |       |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   | TSERR1 | TSTAR1 | TSERR0 | Reserved | TSTAR0 | TSOVF |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   | RC     | RC     | RC     |          | RC     | RC    |

| Bit  | Name     | Function                | Description                                                                                                                                                                                                                                                                                                               | Access |
|------|----------|-------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:6 | Reserved | Reserved                | -                                                                                                                                                                                                                                                                                                                         | -      |
| 5    | TSERR1   | Target Time 1 Error     | <ul style="list-style-type: none"> <li>0b0: Time to be written to target time register 1 is correct.</li> <li>0b1: Time to be written to target time register 1 has passed, the write value is incorrect.</li> </ul>                                                                                                      | RC     |
| 4    | TSTAR1   | PPS1 Interrupt Flag Bit | <ul style="list-style-type: none"> <li>0b0: No interrupt generated</li> <li>0b1: Generate an interrupt</li> </ul> Setting conditions: <ul style="list-style-type: none"> <li>System time reaches the time indicated by target time register 1 (interrupt must be enabled and interrupt output event selected).</li> </ul> | RC     |
| 3    | TSERR0   | Target Time 0 Error     | <ul style="list-style-type: none"> <li>0b0: Time to be written to target time register 0 is correct.</li> <li>0b1: Time to be written to target time register 0 has passed, the write value is incorrect.</li> </ul>                                                                                                      | RC     |
| 2    | Reserved | Reserved                | -                                                                                                                                                                                                                                                                                                                         | -      |
| 1    | TSTAR0   | PPS0 Interrupt Flag Bit | <ul style="list-style-type: none"> <li>0b0: No interrupt generated</li> <li>0b1: Generate an interrupt</li> </ul> Setting conditions: <ul style="list-style-type: none"> <li>System time reaches the time indicated by target time register 0 (interrupt must be enabled and interrupt output event selected).</li> </ul> | RC     |
| 0    | TSOVF    | System Time Overflow    | <ul style="list-style-type: none"> <li>0b0: System time has not overflowed.</li> <li>0b1: System time has overflowed.</li> </ul>                                                                                                                                                                                          | RC     |



## 38.4.3.3 ETH\_PTP Timestamp Basic Addend Register (ETH\_PTP\_TSPADDR)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0x0718 | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | TSPADD       |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access | RW           |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit  | Name   | Function           | Description                                                                                                                                                                                                                                                                                                                                                                                 | Access |
|------|--------|--------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:0 | TSPADD | Basic Addend Value | <p>This register is used by software to linearly adjust clock frequency to match the master clock frequency. The value of this register is added to a 32-bit accumulator in each clock period.</p> <div>  <b>Note</b><br/>           This register is only active in fine calibration mode.         </div> | RW     |

## 38.4.3.4 ETH\_PTP Timestamp Subsecond Addend Register (ETH\_PTP\_TSPNSAR)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |           |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|-----------|---|---|---|---|---|---|---|
| 0x0704 | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7         | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |           |   |   |   |   |   |   |   |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   | TSPNSEADD |   |   |   |   |   |   |   |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   | RW        |   |   |   |   |   |   |   |

| Bit  | Name      | Function               | Description                                                                                                                                                                                                                                                                                                                                                                                     | Access |
|------|-----------|------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:8 | Reserved  | Reserved               | -                                                                                                                                                                                                                                                                                                                                                                                               | -      |
| 7:0  | TSPNSEADD | Subsecond Addend Value | <p>This register contains the 8-bit value used for incrementing the system subsecond register.</p> <ul style="list-style-type: none"> <li>● In coarse calibration mode, the value of this register is added to system time in each clock period.</li> <li>● In fine calibration mode, the value of this register is added to system time only when the 32-bit accumulator overflows.</li> </ul> | RW     |

## 38.4.3.5 ETH\_PTP Timestamp System Second Register (ETH\_PTP\_TMSSECR)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0x0708 | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | TSPSYSSEC    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access | RO           |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit  | Name      | Function           | Description                                          | Access |
|------|-----------|--------------------|------------------------------------------------------|--------|
| 31:0 | TSPSYSSEC | System Second Time | This bit represents the second value of system time. | RO     |

## 38.4.3.6 ETH\_PTP Timestamp System Subsecond Register (ETH\_PTP\_TMSNSER)

| Offset | Bit Position |            |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0x070C | 31           | 30         | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |            |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | Reserved     | TSPSYSNSEC |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access |              | RO         |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit  | Name       | Function               | Description                                             | Access |
|------|------------|------------------------|---------------------------------------------------------|--------|
| 31   | Reserved   | Reserved               | -                                                       | -      |
| 30:0 | TSPSYSNSEC | System Sub-second Time | This bit represents the subsecond value of system time. | RO     |

38.4.3.7 ETH\_PTP Timestamp Update Second Register (ETH\_PTP\_TMUSECR)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0x0710 | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | TSPUPSEC     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access | RW           |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit  | Name     | Function                   | Description                                                          | Access |
|------|----------|----------------------------|----------------------------------------------------------------------|--------|
| 31:0 | TSPUPSEC | System Second Update Value | This bit represents the second value to be updated into system time. | RW     |

## 38.4.3.8 ETH\_PTP Timestamp Update Subsecond Register (ETH\_PTP\_TMUNSER)

| Offset | Bit Position |          |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0x0714 | 31           | 30       | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |          |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | TSPUPNS      | TSPUPNSE |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access | RW           | RW       |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit  | Name     | Function                       | Description                                                                                                                                                                                                                                           | Access |
|------|----------|--------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31   | TSPUPNS  | Update Mode                    | Time Update Sign (Positive/Negative) :<br><ul style="list-style-type: none"> <li>0b0: Subtract the value of update register from system time register value.</li> <li>0b1: Add the value of update register to system time register value.</li> </ul> | RW     |
| 30:0 | TSPUPNSE | System Sub-second Update Value | This bit represents the subsecond value to be updated into system time.                                                                                                                                                                               | RW     |

## 38.4.3.9 ETH\_PTP Timestamp Target Second Register 0(ETH\_PTP\_TMTSECR0)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0x071C | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | TSPTAGSEC0   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access | RW           |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit  | Name       | Function             | Description                                                                                                                                | Access |
|------|------------|----------------------|--------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:0 | TSPTAGSEC0 | Target Second Time 0 | When system time exceeds or equals the set value of this register and target subsecond register 0, an interrupt event or PPS event occurs. | RW     |

## 38.4.3.10 ETH\_PTP Timestamp Target Subsecond Register 0 (ETH\_PTP\_TMTNSER0)

| Offset | Bit Position |             |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|-------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0x0720 | 31           | 30          | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |             |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | Reserved     | TSPTAGNSECO |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access |              | RW          |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit  | Name         | Function                | Description                                                                                                                             | Access |
|------|--------------|-------------------------|-----------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31   | Reserved     | Reserved                | -                                                                                                                                       | -      |
| 30:0 | TSPTAGN-SEC0 | Target Subsecond Time 0 | When system time exceeds or equals the set value of this register and target second register 0, an interrupt event or PPS event occurs. | RW     |

## 38.4.3.11 ETH\_PTP Timestamp Target Second Register 1(ETH\_PTP\_TMTSECR1)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0x0780 | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | TSPTAGSEC1   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access | RW           |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit  | Name       | Function             | Description                                                                                                                                | Access |
|------|------------|----------------------|--------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:0 | TSPTAGSEC1 | Target Second Time 1 | When system time exceeds or equals the set value of this register and target subsecond register 1, an interrupt event or PPS event occurs. | RW     |

## 38.4.3.12 ETH\_PTP Timestamp Target Subsecond Register 1 (ETH\_PTP\_TMTNSER1)

| Offset | Bit Position |             |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|-------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0x0784 | 31           | 30          | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |             |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | Reserved     | TSPTAGNSEC1 |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access |              | RW          |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit  | Name         | Function                | Description                                                                                                                             | Access |
|------|--------------|-------------------------|-----------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31   | Reserved     | Reserved                | -                                                                                                                                       | -      |
| 30:0 | TSPTAGN-SEC1 | Target Subsecond Time 1 | When system time exceeds or equals the set value of this register and target second register 1, an interrupt event or PPS event occurs. | RW     |

## 38.4.3.13 ETH\_PTP PPS Output Control Register (ETH\_PTP\_PPSCTLR)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |        |          |         |          |        |        |         |    |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|--------|----------|---------|----------|--------|--------|---------|----|---|---|---|---|---|---|---|
| 0x072C | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14     | 13       | 12      | 11       | 10     | 9      | 8       | 7  | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |        |          |         |          |        |        |         |    |   |   |   |   |   |   |   |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | TT1SEL | Reserved | PPSFRE1 | Reserved | TT0SEL | PPSOMD | PPSFRE0 |    |   |   |   |   |   |   |   |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RW     |          | RW      |          | RW     | RW     | RW      | RW |   |   |   |   |   |   |   |

| Bit   | Name     | Function                         | Description                                                                                                                                                                                                                                                                                                                                  | Access |
|-------|----------|----------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:15 | Reserved | Reserved                         | -                                                                                                                                                                                                                                                                                                                                            | -      |
| 14:13 | TT1SEL   | Target Time 1 Function Selection | <ul style="list-style-type: none"> <li>0b00: Target register 1 is only used for interrupt output events.</li> <li>0b01: Meaningless, do not set.</li> <li>0b10: Target register 1 is used for interrupt output events and PPS1 single output events.</li> <li>0b11: Target register 1 is only used for PPS1 single output events.</li> </ul> | RW     |
| 12:11 | Reserved | Reserved                         | -                                                                                                                                                                                                                                                                                                                                            | -      |
| 10:8  | PPSFRE1  | PPS1 Output Frequency            | Read always returns 0<br>Write: <ul style="list-style-type: none"> <li>0b001: When system time equals target second register 1, generate one pulse and no more pulses afterwards.</li> <li>Others: Setting prohibited</li> </ul>                                                                                                             | RW     |
| 7     | Reserved | Reserved                         | -                                                                                                                                                                                                                                                                                                                                            | -      |
| 6:5   | TT0SEL   | Target Time 0 Function Selection | <ul style="list-style-type: none"> <li>0b00: Target register 0 is only used for interrupt output events.</li> <li>0b01: Meaningless, do not set.</li> <li>0b10: Target register 0 is used for interrupt output events and PPS0 single output events.</li> <li>0b11: Target register 0 is only used for PPS0 single output events.</li> </ul> | RW     |
| 4     | PPSOMD   | PPS Output Mode                  | <ul style="list-style-type: none"> <li>0b0: Continuous output mode</li> <li>0b1: Single output mode</li> </ul> <div>  <b>Note</b><br/>           PPS0 supports both modes above, PPS1 only supports single output mode.         </div>                    | RW     |
| 3:0   | PPSFRE0  | PPS0 Output Frequency            | <b>In continuous output mode:</b><br>When system time experiences second carry (sub-second time overflow), PPS0 generates different forms of output according to the following settings:                                                                                                                                                     | RW     |

| Bit | Name    | Function              | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | Access |
|-----|---------|-----------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 3:0 | PPSFRE0 | PPS0 Output Frequency | <ul style="list-style-type: none"> <li>● 0b0000: Continuous output with 1Hz pulse interval</li> <li>● 0b0001: Continuous output with 2Hz clock frequency</li> <li>● 0b0010: Continuous output with 4Hz clock frequency</li> <li>● 0b0011: Continuous output with 8Hz clock frequency</li> <li>● 0b0100: Continuous output with 16Hz clock frequency</li> <li>● .....</li> <li>● 0b1111: Continuous output with 32768Hz clock frequency</li> </ul> <div style="border: 1px solid black; padding: 5px; margin-top: 10px;">  <b>Note</b><br/>           In continuous output mode, the TSPSSR bit of time-stamp control register must be set to 0.         </div> <p><b>In single output mode:</b> read always returns 0<br/>           Write:</p> <ul style="list-style-type: none"> <li>● 0b0001: When system time equals target second register 0, generate one pulse and no more pulses afterwards.</li> <li>● Others: Setting prohibited</li> </ul> | RW     |



## 38.4.4 ETH\_DMA Register

### 38.4.4.1 ETH\_DMA Bus Mode Register (ETH\_DMA\_BUSMODR)

| Offset | Bit Position |    |    |    |      |      |     |       |      |      |    |    |    |      |      |      |    |    |    |      |     |    |   |   |      |     |   |   |   |   |   |   |
|--------|--------------|----|----|----|------|------|-----|-------|------|------|----|----|----|------|------|------|----|----|----|------|-----|----|---|---|------|-----|---|---|---|---|---|---|
| 0x1000 | 31           | 30 | 29 | 28 | 27   | 26   | 25  | 24    | 23   | 22   | 21 | 20 | 19 | 18   | 17   | 16   | 15 | 14 | 13 | 12   | 11  | 10 | 9 | 8 | 7    | 6   | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00020101   |    |    |    |      |      |     |       |      |      |    |    |    |      |      |      |    |    |    |      |     |    |   |   |      |     |   |   |   |   |   |   |
| Name   | Reserved     |    |    |    | TXPR | MBST | AAL | M8PBL | SPBL | RPBL |    |    |    | FBST | PRAT | TPBL |    |    |    | DSEN | DSL |    |   |   | DMAA | SWR |   |   |   |   |   |   |
| Access |              |    |    |    | RW   | RW   | RW  | RW    | RW   | RW   |    |    |    | RW   | RW   | RW   |    |    |    | RW   | RW  |    |   |   | RW   | RW  |   |   |   |   |   |   |

| Bit   | Name     | Function                     | Description                                                                                                                                                                                                                                                                                                  | Access |
|-------|----------|------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:28 | Reserved | Reserved                     | -                                                                                                                                                                                                                                                                                                            | -      |
| 27    | TXPR     | Transfer Priority            | <ul style="list-style-type: none"> <li>0b0: RxDMA transaction priority is higher than TxDMA.</li> <li>0b1: TxDMA transaction priority is higher than RxDMA.</li> </ul>                                                                                                                                       | R/W    |
| 26    | MBST     | Mixed Burst Transfer         | When this bit is set to 1 and FBST bit is 0, AHB Master port initiates all bursts longer than 16 with INCR, and bursts equal to or shorter than 16 with SINGLE and INCRx (x=4/8/16).                                                                                                                         | R/W    |
| 25    | AAL      | Address Alignment            | When this bit is set to 1: <ol style="list-style-type: none"> <li>If FBST=1, AHB interface generates all bursts aligned with the LS bit of start address.</li> <li>If FBST=0, the first burst (accessing start address of data buffer) is not aligned, but subsequent bursts are address-aligned.</li> </ol> | R/W    |
| 24    | M8PBL    | 8x PBL                       | When this bit is set to 1, the programmed PBL value (RPBL bit and TPBL bit) is multiplied by 8, so DMA transfers data with maximum 8, 16, 32, 64, 128 and 256 beats according to PBL value.                                                                                                                  | R/W    |
| 23    | SPBL     | Independent PBL Control      | <ul style="list-style-type: none"> <li>0b0: The TPBL setting of this register is used for both TxDMA and RxDMA transactions.</li> <li>0b1: The TPBL setting of this register is only used for TxDMA transactions, RPBL setting is used for RxDMA transactions.</li> </ul>                                    | R/W    |
| 22:17 | RPBL     | Rx Programmable Burst Length | Same meaning as TPBL bit of this register, only used for RxDMA transactions. <div>  <b>Note</b><br/>           This bit is only useful when SPBL bit of this register is enable.         </div>                           | R/W    |

| Bit   | Name | Function                  | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | Access |
|-------|------|---------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 16    | FBST | Fixed Burst Transfer      | <ul style="list-style-type: none"> <li>0b0: DMA Master interface only uses SINGLE and INCR access types.</li> <li>0b1: DMA Master interface uses SINGLE, INCR4, INCR8 and INCR16 access types.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | R/W    |
| 15:14 | PRAT | Priority Ratio            | <p>Preemption priority ratio between Rx and Tx for BUS:</p> <ul style="list-style-type: none"> <li>0b00: 1:1</li> <li>0b11: 2:1</li> <li>0b10: 3:1</li> <li>0b11: 4:1</li> </ul> <div>  <b>Note</b> <ol style="list-style-type: none"> <li>This bit is only active when DMAA bit of this register is 0.</li> <li>The selection of Tx:Rx or Rx:Tx is determined by TXPR bit of this register.</li> </ol> </div>                                                                                                                                                                                                                                                              | R/W    |
| 13:8  | TPBL | Programmable Burst Length | <p>This bit indicates the maximum number of beats to be transferred in one TxDMA or RxDMA transaction, which is the maximum value used in single block read/write operations. Whenever DMA starts burst transfer on host bus, it always attempts to burst according to the specified PBL. PBL can be programmed with values 1, 2, 4, 8, 16 and 32, any other values causes undefined behavior.</p> <div>  <b>Note</b> <p>PBL values have the following limitations:</p> <ol style="list-style-type: none"> <li>The possible maximum number of beats (PBL) is limited by Tx FIFO and Rx FIFO sizes.</li> <li>Do not program PBL values beyond the range.</li> </ol> </div> | R/W    |
| 7     | DSEN | Enhanced Descriptor       | <p>Enhanced Descriptor Format Enable:</p> <ul style="list-style-type: none"> <li>0b0: Normal descriptor</li> <li>0b1: Enhanced descriptor</li> </ul> <div>  <b>Note</b> <p>Enhanced descriptors must be used when PTP function is enabled (ETH_PTP_TSPCTLR.TSPEN=1) or checksum offload function is enabled (ETH_MAC_CONFIGR.IPCO=1).</p> </div>                                                                                                                                                                                                                                                                                                                          | R/W    |
| 6:2   | DSL  | Descriptor Skip Length    | <p>This bit specifies the number of Word to skip between two unlinked descriptors, with address jumping from current descriptor end to next descriptor start. When DSL value equals zero, in ring mode, DMA treats descriptor table as continuous.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | R/W    |
| 1     | DMAA | DMA Arbitration           | <ul style="list-style-type: none"> <li>0b0: Round-robin priority, arbitration method is determined by PRAT bit and TXPR bit of this register.</li> <li>0b1: Fixed priority, priority method is determined by TXPR bit of this register.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | R/W    |

| Bit | Name | Function       | Description                                                                                                                                                                                      | Access |
|-----|------|----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 0   | SWR  | Software Reset | When this bit is set to 1, DMA controller resets all internal registers and logic of MAC subsystem. This bit is automatically cleared after reset operation completes in all core clock domains. | R/W    |

## 38.4.4.2 ETH\_DMA Operation Mode Register (ETH\_DMA\_OPRMODR)

| Offset | Bit Position |    |    |    |       |    |     |    |      |    |          |    |     |    |     |    |          |    |     |    |     |    |          |   |   |   |     |   |     |   |     |   |     |  |     |  |     |  |          |  |  |
|--------|--------------|----|----|----|-------|----|-----|----|------|----|----------|----|-----|----|-----|----|----------|----|-----|----|-----|----|----------|---|---|---|-----|---|-----|---|-----|---|-----|--|-----|--|-----|--|----------|--|--|
| 0x1018 | 31           | 30 | 29 | 28 | 27    | 26 | 25  | 24 | 23   | 22 | 21       | 20 | 19  | 18 | 17  | 16 | 15       | 14 | 13  | 12 | 11  | 10 | 9        | 8 | 7 | 6 | 5   | 4 | 3   | 2 | 1   | 0 |     |  |     |  |     |  |          |  |  |
| Reset  | 0x00000000   |    |    |    |       |    |     |    |      |    |          |    |     |    |     |    |          |    |     |    |     |    |          |   |   |   |     |   |     |   |     |   |     |  |     |  |     |  |          |  |  |
| Name   | Reserved     |    |    |    | DTCOE |    | RSF |    | DFRF |    | Reserved |    | TSF |    | FTF |    | Reserved |    | TTC |    | STT |    | Reserved |   |   |   | FEF |   | FUF |   | DGF |   | RTC |  | OSF |  | STR |  | Reserved |  |  |
| Access |              |    |    |    | RW    |    | RW  |    | RW   |    |          |    | RW  |    | RW  |    |          |    | RW  |    | RW  |    |          |   |   |   | RW  |   | RW  |   | RW  |   | RW  |  | RW  |  | RW  |  | RW       |  |  |

| Bit   | Name     | Function                                            | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | Access |
|-------|----------|-----------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:27 | Reserved | Reserved                                            | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | -      |
| 26    | DTCOE    | Disable discarding TCP/IP checksum and error frames | <ul style="list-style-type: none"> <li>0b0: If the FEF bit of this register is set to 1, all error frames is discarded.</li> <li>0b1: If the received frame contains only errors detected by the receive checksum offload engine, the core does not discard the frame.</li> </ul> <div>  <b>Note</b><br/>           These frames have no errors (including CRC errors) in the Ethernet frames received by the MAC, only errors in the encapsulated payload.         </div> | RW     |
| 25    | RSF      | Receive store forward                               | <ul style="list-style-type: none"> <li>0b0: The read action for Rx FIFO is performed only when the frames in Rx FIFO reach the value specified by the RTC bit of this register.</li> <li>0b1: The read action is executed immediately after a complete frame is written to Rx FIFO, ignoring the RTC bit.</li> </ul>                                                                                                                                                                                                                                        | RW     |
| 24    | DFRF     | Disable refreshing received frames                  | When this bit is set to 1, RxDMA does not refresh any frames due to unavailable receive descriptors/buffers. For details, refer to " <a href="#">Receive Process Suspended</a> ".                                                                                                                                                                                                                                                                                                                                                                           | RW     |
| 23:22 | Reserved | Reserved                                            | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | -      |
| 21    | TSF      | Transmit store forward                              | <ul style="list-style-type: none"> <li>0b0: The read action for Tx FIFO is performed only when the frames in Tx FIFO reach the value specified by the TTC bit of this register, and MAC transmission starts accordingly.</li> <li>0b1: If there is a complete frame in Tx FIFO, MAC transmission starts, ignoring the TTC bit.</li> </ul> <div>  <b>Note</b><br/>           This bit can only be changed when the transmission is stopped.         </div>                | RW     |

| Bit   | Name     | Function            | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | Access |
|-------|----------|---------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 20    | FTF      | Flush Tx FIFO       | <p>When this bit is set to 1, Tx FIFO is reset to its default value, and all data in Tx FIFO is lost/flushed. This bit is automatically cleared when the flush operation ends.</p> <div>  <b>Note</b><br/>           Do not perform write operations on the operating mode register before this bit is cleared.         </div>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | RW     |
| 19:17 | Reserved | Reserved            | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | -      |
| 16:14 | TTC      | Transmit threshold  | <p>This bit controls the threshold level of Tx FIFO. DMA transmission starts when the frame size in Tx FIFO exceeds the threshold. Additionally, complete frames shorter than the threshold are transmitted automatically:</p> <ul style="list-style-type: none"> <li>● 0b000: 64</li> <li>● 0b001: 128</li> <li>● 0b010: 192</li> <li>● 0b011: 256</li> <li>● 0b100: 40</li> <li>● 0b101: 32</li> <li>● 0b110: 24</li> <li>● 0b111: 16</li> </ul> <div>  <b>Note</b><br/>           This bit is used only when the TSF bit is 0.         </div>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | RW     |
| 13    | STT      | Transmit start stop | <ul style="list-style-type: none"> <li>● 0b0: After completing the transmission of the current frame, the transmit process enters the stopped state and saves the next descriptor position in the transmit list. This descriptor position becomes the current position after re-starting transmission.</li> <li>● 0b1: Start the transmit process. DMA checks the current position in the transmit list, i.e., the address set by the transmit descriptor address list register (ETH_DMA_TXDLADR) or the reserved position from the last transmission stop, to retrieve descriptors and locate frames to be transmitted.</li> </ul> <p>If the current descriptor does not belong to TxDMA, the transmit process enters the suspended state. If this command is issued before setting the ETH_DMA_TXDLADR register, TxDMA behavior is unpredictable.</p> <div>  <b>Note</b><br/>           The start transmit command is enable only when transmission is stopped. The stop transmit command is enable only when the current frame transmission ends or the transmit process is in the suspended state.         </div> | RW     |

| Bit  | Name     | Function                        | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | Access |
|------|----------|---------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 12:8 | Reserved | Reserved                        | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | -      |
| 7    | FEF      | Forward error frames            | <ul style="list-style-type: none"> <li>0b0: Rx FIFO discards frames with error status (CRC error, collision error, giant frame, watchdog timeout, overflow).<br/>If the start byte pointer of a frame is transferred to the read controller side (in threshold mode), the frame is not discarded. If the start byte of the frame has not been transmitted on the bus, Rx FIFO discards this error frame.</li> <li>0b1: All frames except runt error frames are forwarded.</li> </ul>                                                                                                          | RW     |
| 6    | FUF      | Forward under-sized good frames | <ul style="list-style-type: none"> <li>0b0: Rx FIFO discards all frames shorter than 64 bytes unless such frames are transmitted due to a lower receive threshold (e.g., RTC=01).</li> <li>0b1: Rx FIFO forwards undersized frames (error-free but shorter than 64 bytes) containing PAD bytes and CRC.</li> </ul>                                                                                                                                                                                                                                                                            | RW     |
| 5    | DGF      | Discard giant frames            | <ul style="list-style-type: none"> <li>0b0: Rx FIFO does not discard giant frames.</li> <li>0b1: Rx FIFO discards giant frames.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                    | RW     |
| 4:3  | RTC      | Receive threshold               | <p>This bit controls the threshold level of Rx FIFO. DMA transfer (request) starts when the frame size in Rx FIFO exceeds the threshold. Additionally, complete frames shorter than the threshold are transmitted automatically:</p> <ul style="list-style-type: none"> <li>0b00: 64</li> <li>0b01: 32</li> <li>0b10: 96</li> <li>0b11: 128</li> </ul> <div style="border: 1px solid black; padding: 5px; margin-top: 10px;">  <b>Note</b><br/>This bit is used only when the RSF bit is 0.         </div> | RW     |
| 2    | OSF      | Process second frame            | When this bit is set to 1, DMA is commanded to process the second transmit data frame, even if the status of the first frame has not been obtained.                                                                                                                                                                                                                                                                                                                                                                                                                                           | RW     |

| Bit | Name            | Function           | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | Access |
|-----|-----------------|--------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 1   | STR             | Receive start stop | <ul style="list-style-type: none"> <li>0b0: RxDMA stops operation after transmitting the current frame and saves the next descriptor position in the receive list. This descriptor position becomes the current position after re-starting the receive process.</li> <li>0b1: Start the receive process. DMA attempts to retrieve descriptors from the current position in the receive list, i.e., the address set by the receive descriptor address list register (ETH_DMA_RXDLADR) or the reserved position from the last receive stop, and processes incoming frames.</li> </ul> <p>If RxDMA does not occupy any descriptors, the receive process enters the suspended state. If this command is issued before setting the ETH_DMA_RXDLADR register, RxDMA behavior is unpredictable.</p> <div>  <b>Note</b><br/>           The start receive command is enable only when reception is stopped. The stop receive command is enable only when the current receive process is running (waiting for receive data packets) or in the suspended state.         </div> | RW     |
| 0   | <i>Reserved</i> | <i>Reserved</i>    | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | -      |

## 38.4.4.3 ETH\_DMA Action Status Register (ETH\_DMA\_DMASTSR)

| Offset | Bit Position |      |    |      |    |      |    |          |      |    |    |      |    |    |      |    |    |     |    |     |    |     |   |     |   |          |     |   |     |   |     |   |     |  |     |  |     |  |     |  |     |  |     |  |     |  |     |  |
|--------|--------------|------|----|------|----|------|----|----------|------|----|----|------|----|----|------|----|----|-----|----|-----|----|-----|---|-----|---|----------|-----|---|-----|---|-----|---|-----|--|-----|--|-----|--|-----|--|-----|--|-----|--|-----|--|-----|--|
| 0x1014 | 31           | 30   | 29 | 28   | 27 | 26   | 25 | 24       | 23   | 22 | 21 | 20   | 19 | 18 | 17   | 16 | 15 | 14  | 13 | 12  | 11 | 10  | 9 | 8   | 7 | 6        | 5   | 4 | 3   | 2 | 1   | 0 |     |  |     |  |     |  |     |  |     |  |     |  |     |  |     |  |
| Reset  | 0x00000000   |      |    |      |    |      |    |          |      |    |    |      |    |    |      |    |    |     |    |     |    |     |   |     |   |          |     |   |     |   |     |   |     |  |     |  |     |  |     |  |     |  |     |  |     |  |     |  |
| Name   | Reserved     | PTPS |    | PMTS |    | MMCS |    | Reserved | EBUS |    |    | TSTS |    |    | RSTS |    |    | NIS |    | AIS |    | ERS |   | FBS |   | Reserved | ETS |   | RWS |   | RSS |   | RUS |  | RIS |  | UNS |  | OVS |  | TJS |  | TUS |  | TSS |  | TIS |  |
| Access |              | RO   |    | RO   |    | RO   |    |          | RO   |    |    | RO   |    |    | RO   |    |    | RW  |    | RW  |    | RW  |   | RW  |   |          | RW  |   | RW  |   | RW  |   | RW  |  | RW  |  | RW  |  | RW  |  | RW  |  | RW  |  | RW  |  |     |  |

| Bit   | Name     | Function         | Description                                                                                                                                                                                                                                                                                                                                                                                     | Access |
|-------|----------|------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:30 | Reserved | Reserved         | -                                                                                                                                                                                                                                                                                                                                                                                               | -      |
| 29    | PTPS     | PTP Event Status | <ul style="list-style-type: none"> <li>0b0: No timestamp event occurs in the PTP module.</li> <li>0b1: A timestamp event occurs in the PTP module.</li> </ul> <div>  <b>Note</b><br/>           This bit is automatically cleared when the event status that set it is cleared.         </div>                 | RO     |
| 28    | PMTS     | PMT Event Status | <ul style="list-style-type: none"> <li>0b0: No wakeup event occurs in the PMT module.</li> <li>0b1: A wakeup event occurs in the PMT module.</li> </ul> <div>  <b>Note</b><br/>           This bit is automatically cleared when the event status that set it is cleared.         </div>                     | RO     |
| 27    | MMCS     | MMC Event Status | <ul style="list-style-type: none"> <li>0b0: No frame statistics event occurs in the MMC module.</li> <li>0b1: A frame statistics event occurs in the MMC module.</li> </ul> <div>  <b>Note</b><br/>           This bit is automatically cleared when the event status that set it is cleared.         </div> | RO     |
| 26    | Reserved | Reserved         | -                                                                                                                                                                                                                                                                                                                                                                                               | -      |



| Bit   | Name | Function            | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | Access |
|-------|------|---------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 25:23 | EBUS | Error Bit Status    | <p>This bit indicates the type of error causing a bus error (error response on the AHB interface) and does not generate an interrupt.</p> <ul style="list-style-type: none"> <li>● 0b000: Error occurred during RxDMA write data.</li> <li>● 0b011: Error occurred during TxDMA read data.</li> <li>● 0b100: Error occurred during RxDMA write descriptor.</li> <li>● 0b101: Error occurred during TxDMA write descriptor.</li> <li>● 0b110: Error occurred during RxDMA read descriptor.</li> <li>● 0b111: Error occurred during TxDMA read descriptor.</li> <li>● Other: Reserved</li> </ul> <div>  <b>Note</b><br/>           This bit is enable only when the FBS bit of this register is set to 1.         </div> | RO     |
| 22:20 | TSTS | Transmitting Status | <p>This bit indicates the state of the TxDMA FSM and does not trigger an interrupt.</p> <ul style="list-style-type: none"> <li>● 0b000: Stopped, issued reset or stop transmit command.</li> <li>● 0b001: Running, fetching transmit descriptors.</li> <li>● 0b010: Running, waiting for status.</li> <li>● 0b011: Running, reading data from the host memory buffer and adding it to the transmit buffer (Tx FIFO) queue.</li> <li>● 0b100: Timestamp writing</li> <li>● 0b101: Reserved</li> <li>● 0b110: Suspended, transmit descriptor unavailable or transmit buffer underflow.</li> <li>● 0b111: Running, closing transmit descriptors.</li> </ul>                                                                                                                                                | RO     |
| 19:17 | RSTS | Receiving Status    | <p>This bit indicates the status of the RxDMA FSM and does not generate an interrupt.</p> <ul style="list-style-type: none"> <li>● 0b000: Stopped, issued reset or stop receive command.</li> <li>● 0b001: Running, fetching receive descriptors.</li> <li>● 0b010: Reserved</li> <li>● 0b011: Running, waiting for receive data packets.</li> <li>● 0b100: Suspended, receive descriptor unavailable.</li> <li>● 0b101: Running, closing receive descriptors.</li> <li>● 0b110: Timestamp writing</li> <li>● 0b111: Running, transferring data from receive buffers to host memory.</li> </ul>                                                                                                                                                                                                         | RO     |

| Bit   | Name     | Function                          | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | Access |
|-------|----------|-----------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 16    | NIS      | Normal Interrupt Status Summary   | <p>This bit is set to 1 when any of the following conditions are met:</p> <ul style="list-style-type: none"> <li>● TIS=1 &amp;&amp; ETH_DMA_INTENAR.TIE=1</li> <li>● TUS=1 &amp;&amp; ETH_DMA_INTENAR.TUE=1</li> <li>● RIS=1 &amp;&amp; ETH_DMA_INTENAR.RIE=1</li> <li>● ERS=1 &amp;&amp; ETH_DMA_INTENAR.ERE=1</li> </ul> <div>  <b>Note</b> <ol style="list-style-type: none"> <li>1. Only unmasked interrupt enable bits affect this normal interrupt status summary bit.</li> <li>2. Whenever the corresponding bit that caused this bit to be set to 1 is cleared, this bit must also be cleared (by writing 1 to it).</li> </ol> </div>                                                                                                                                                                                                                                                               | RW     |
| 15    | AIS      | Abnormal Interrupt Status Summary | <p>This bit is set to 1 when any of the following conditions are met:</p> <ul style="list-style-type: none"> <li>● TSS=1 &amp;&amp; ETH_DMA_INTENAR.TSE=1</li> <li>● TJS=1 &amp;&amp; ETH_DMA_INTENAR.TJE=1</li> <li>● OVS=1 &amp;&amp; ETH_DMA_INTENAR.OVE=1</li> <li>● UNS=1 &amp;&amp; ETH_DMA_INTENAR.UNE=1</li> <li>● RUS=1 &amp;&amp; ETH_DMA_INTENAR.RUE=1</li> <li>● RSS=1 &amp;&amp; ETH_DMA_INTENAR.RSE=1</li> <li>● RWS=1 &amp;&amp; ETH_DMA_INTENAR.RWE=1</li> <li>● ETS=1 &amp;&amp; ETH_DMA_INTENAR.ETE=1</li> <li>● FBS=1 &amp;&amp; ETH_DMA_INTENAR.FBE=1</li> </ul> <div>  <b>Note</b> <ol style="list-style-type: none"> <li>1. Only unmasked interrupt enable bits affect this abnormal interrupt status summary bit.</li> <li>2. Whenever the corresponding bit that caused this bit to be set to 1 is cleared, this bit must also be cleared (by writing 1 to it).</li> </ol> </div> | RW     |
| 14    | ERS      | Early Receive Status              | <p>This bit indicates that DMA has filled the first data buffer of the packet.</p> <div>  <b>Note</b> <p>This bit can be cleared by writing 1 to it or automatically cleared after the RIS bit of this register is set.</p> </div>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | RW     |
| 13    | FBS      | Fatal Bus Error Status            | <p>This bit indicates a bus error has occurred. For the specific error type, refer to the EBUS bit of this register. When this bit is set to 1, the corresponding DMA engine disables all its bus accesses.</p> <div>  <b>Note</b> <p>This bit can be cleared by writing 1 to it.</p> </div>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | RW     |
| 12:11 | Reserved | Reserved                          | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | -      |

| Bit | Name | Function                          | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | Access |
|-----|------|-----------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 10  | ETS  | Early Transmit Status             | <p>This bit indicates that the frame to be transmitted is fully transferred to Tx FIFO.</p> <div>  <b>Note</b><br/>           This bit can be cleared by writing 1 to it.         </div>                                                                                                                                                                                                                                                                                                                                                                                                                                                       | RW     |
| 9   | RWS  | Receive Watch-dog Status          | <p>This bit is set to 1 when the length of the received frame exceeds 2048 bytes.</p> <div>  <b>Note</b><br/>           This bit can be cleared by writing 1 to it.         </div>                                                                                                                                                                                                                                                                                                                                                                                                                                                             | RW     |
| 8   | RSS  | Receive Stopped Status            | <p>This bit is set to 1 when the receive process enters the stopped state.</p> <div>  <b>Note</b><br/>           This bit can be cleared by writing 1 to it.         </div>                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | RW     |
| 7   | RUS  | Receive Buffer Unavailable Status | <p>This bit indicates that the next descriptor in the receive list is owned by the host, DMA cannot acquire it, and the receive process enters the suspended state.</p> <p>To resume processing receive descriptors, the host must modify the ownership of the descriptor and issue a receive poll demand command.</p> <p>If no receive poll demand command is issued, the receive process resumes when the next recognized input frame is received.</p> <p>This bit is set to 1 only when the previous receive descriptor is owned by DMA.</p> <div>  <b>Note</b><br/>           This bit can be cleared by writing 1 to it.         </div> | RW     |
| 6   | RIS  | Receive Status                    | <p>This bit indicates that frame reception is complete, specific frame status information is published in the descriptor, and reception remains in the running state.</p> <div>  <b>Note</b><br/>           This bit can be cleared by writing 1 to it.         </div>                                                                                                                                                                                                                                                                                                                                                                       | RW     |
| 5   | UNS  | Transmit Underflow Status         | <p>This bit indicates that during frame transmission, a transmit buffer underflow occurred, transmission enters the suspended state, and the underflow error flag bit TDES0[1] in the descriptor is set to 1.</p> <div>  <b>Note</b><br/>           This bit can be cleared by writing 1 to it.         </div>                                                                                                                                                                                                                                                                                                                               | RW     |

| Bit | Name | Function                           | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | Access |
|-----|------|------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 4   | OVS  | Receive Overflow Status            | <p>This bit indicates that during frame reception, a receive buffer overflow occurred. If part of the frame is transferred to the application, the overflow error bit RDES0[11] in the descriptor is set to 1.</p> <div>  <b>Note</b><br/>           This bit can be cleared by writing 1 to it.         </div>                                                                                                           | RW     |
| 3   | TJS  | Transmit Jabber Timeout Status     | <p>This bit indicates that the transmit Jabber timer has expired, meaning the transmitter is overly active. The transmit process is aborted and placed in the stopped state, and the transmit Jabber timeout flag bit TDES0[14] in the descriptor is set to 1.</p> <div>  <b>Note</b><br/>           This bit can be cleared by writing 1 to it.         </div>                                                           | RW     |
| 2   | TUS  | Transmit Buffer Unavailable Status | <p>This bit indicates that the next descriptor in the transmit list is owned by the host, DMA cannot acquire it, and transmission enters the suspended state.</p> <p>To resume processing transmit descriptors, the host should change the ownership of the descriptors and then issue a transmit poll demand command.</p> <div>  <b>Note</b><br/>           This bit can be cleared by writing 1 to it.         </div> | RW     |
| 1   | TSS  | Transmit Stop Status               | <p>This bit is set to 1 when the transmit process enters the stopped state.</p> <div>  <b>Note</b><br/>           This bit can be cleared by writing 1 to it.         </div>                                                                                                                                                                                                                                            | RW     |
| 0   | TIS  | Transmit Interrupt Status          | <p>This bit indicates that frame transmission is complete, and specific frame status information is published in the descriptor.</p> <div>  <b>Note</b><br/>           This bit can be cleared by writing 1 to it.         </div>                                                                                                                                                                                       | RW     |

## 38.4.4.4 ETH\_DMA Interrupt Enable Register (ETH\_DMA\_INTENAR)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |     |     |     |     |          |     |     |     |     |     |     |     |     |     |     |     |    |    |    |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-----|-----|-----|-----|----------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|----|----|----|
| 0x101C | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15  | 14  | 13  | 12  | 11       | 10  | 9   | 8   | 7   | 6   | 5   | 4   | 3   | 2   | 1   | 0   |    |    |    |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |     |     |     |     |          |     |     |     |     |     |     |     |     |     |     |     |    |    |    |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | NIE | AIE | ERE | FBE | Reserved | ETE | RWE | RSE | RUE | RIE | UNE | OVE | TJE | TUE | TSE | TIE |    |    |    |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RW  | RW  | RW  | RW  |          | RW  | RW  | RW  | RW  | RW  | RW  | RW  | RW  | RW  | RW  | RW  | RW | RW | RW |

| Bit   | Name     | Function                          | Description                                                                                                                                                                                                                                                                                                                                                                                                                                | Access |
|-------|----------|-----------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:17 | Reserved | Reserved                          | -                                                                                                                                                                                                                                                                                                                                                                                                                                          | -      |
| 16    | NIE      | Normal Interrupt Enable Summary   | <ul style="list-style-type: none"> <li>0b0: Normal event interrupt disabled.</li> <li>0b1: Normal event interrupt enabled.</li> </ul> <div>  <b>Note</b><br/>           Normal events refer to events indicated by bits Bit0, Bit2, Bit6, and Bit14 of the action status register ETH_DMA_DMASTSAR.         </div>                                        | RW     |
| 15    | AIE      | Abnormal Interrupt Enable Summary | <ul style="list-style-type: none"> <li>0b0: Abnormal event interrupt disabled.</li> <li>0b1: Abnormal event interrupt enabled.</li> </ul> <div>  <b>Note</b><br/>           Abnormal events refer to events indicated by bits Bit1, Bit3, Bit4, Bit5, Bit7, Bit8, Bit9, Bit10, and Bit13 of the action status register ETH_DMA_DMASTSAR.         </div> | RW     |
| 14    | ERE      | Early Receive Interrupt Enabled   | <ul style="list-style-type: none"> <li>0b0: Early receive interrupt disabled.</li> <li>0b1: Early receive interrupt enabled.</li> </ul> <div>  <b>Note</b><br/>           This bit is enable after the NIE bit is set to 1.         </div>                                                                                                              | RW     |
| 13    | FBE      | Fatal Bus Error Interrupt Enabled | <ul style="list-style-type: none"> <li>0b0: Fatal bus error interrupt disabled.</li> <li>0b1: Fatal bus error interrupt enabled.</li> </ul> <div>  <b>Note</b><br/>           This bit is enable after the AIE bit is set to 1.         </div>                                                                                                          | RW     |
| 12:11 | Reserved | Reserved                          | -                                                                                                                                                                                                                                                                                                                                                                                                                                          | -      |
| 10    | ETE      | Early Transmit Interrupt Enabled  | <ul style="list-style-type: none"> <li>0b0: Early transmit interrupt disabled.</li> <li>0b1: Early transmit interrupt enabled.</li> </ul> <div>  <b>Note</b><br/>           This bit is enable after the AIE bit is set to 1.         </div>                                                                                                            | RW     |

| Bit | Name | Function                                     | Description                                                                                                                                                                                                                                                                                                                                           | Access |
|-----|------|----------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 9   | RWE  | Receive Watchdog Overflow Interrupt Enabled  | <ul style="list-style-type: none"> <li>0b0: Receive watchdog overflow interrupt disabled.</li> <li>0b1: Receive watchdog overflow interrupt enabled.</li> </ul> <div>  <b>Note</b><br/>           This bit is enable after the AIE bit is set to 1.         </div>   | RW     |
| 8   | RSE  | Receive Stop Interrupt Enabled               | <ul style="list-style-type: none"> <li>0b0: Receive stop interrupt disabled.</li> <li>0b1: Receive stop interrupt enabled.</li> </ul> <div>  <b>Note</b><br/>           This bit is enable after the AIE bit is set to 1.         </div>                             | RW     |
| 7   | RUE  | Receive Buffer Unavailable Interrupt Enabled | <ul style="list-style-type: none"> <li>0b0: Receive buffer unavailable interrupt disabled.</li> <li>0b1: Receive buffer unavailable interrupt enabled.</li> </ul> <div>  <b>Note</b><br/>           This bit is enable after the AIE bit is set to 1.         </div> | RW     |
| 6   | RIE  | Receive Interrupt Enabled                    | <ul style="list-style-type: none"> <li>0b0: Receive interrupt disabled.</li> <li>0b1: Receive interrupt enabled.</li> </ul> <div>  <b>Note</b><br/>           This bit is enable after the NIE bit is set to 1.         </div>                                     | RW     |
| 5   | UNE  | Transmit Underflow Interrupt Enabled         | <ul style="list-style-type: none"> <li>0b0: Transmit underflow interrupt disabled.</li> <li>0b1: Transmit underflow interrupt enabled.</li> </ul> <div>  <b>Note</b><br/>           This bit is enable after the AIE bit is set to 1.         </div>               | RW     |
| 4   | OVE  | Receive Overflow Interrupt Enabled           | <ul style="list-style-type: none"> <li>0b0: Receive overflow interrupt disabled.</li> <li>0b1: Receive overflow interrupt enabled.</li> </ul> <div>  <b>Note</b><br/>           This bit is enable after the AIE bit is set to 1.         </div>                   | RW     |
| 3   | TJE  | Transmit Jabber Timeout Interrupt Enabled    | <ul style="list-style-type: none"> <li>0b0: Transmit Jabber timeout interrupt disabled.</li> <li>0b1: Transmit Jabber timeout interrupt enabled.</li> </ul> <div>  <b>Note</b><br/>           This bit is enable after the AIE bit is set to 1.         </div>     | RW     |

| Bit | Name | Function                                      | Description                                                                                                                                                                                                                                                                                                                                                 | Access |
|-----|------|-----------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 2   | TUE  | Transmit Buffer Unavailable Interrupt Enabled | <ul style="list-style-type: none"> <li>● 0b0: Transmit buffer unavailable interrupt disabled.</li> <li>● 0b1: Transmit buffer unavailable interrupt enabled.</li> </ul> <div>  <b>Note</b><br/>           This bit is enable after the NIE bit is set to 1.         </div> | RW     |
| 1   | TSE  | Transmit Stopped Interrupt Enabled            | <ul style="list-style-type: none"> <li>● 0b0: Transmit stop interrupt disabled.</li> <li>● 0b1: Transmit stop interrupt enabled.</li> </ul> <div>  <b>Note</b><br/>           This bit is enable after the AIE bit is set to 1.         </div>                             | RW     |
| 0   | TIE  | Transmit Interrupt Enabled                    | <ul style="list-style-type: none"> <li>● 0b0: Transmit interrupt disabled.</li> <li>● 0b1: Transmit interrupt enabled.</li> </ul> <div>  <b>Note</b><br/>           This bit is enable after the NIE bit is set to 1.         </div>                                       | RW     |

## 38.4.4.5 ETH\_DMA Frame Loss Statistics Register (ETH\_DMA\_RFR CNTR)

| Offset | Bit Position |    |        |        |    |    |    |    |    |    |    |    |    |    |        |        |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |
|--------|--------------|----|--------|--------|----|----|----|----|----|----|----|----|----|----|--------|--------|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|--|
| 0x1020 | 31           | 30 | 29     | 28     | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17     | 16     | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |  |
| Reset  | 0x00000000   |    |        |        |    |    |    |    |    |    |    |    |    |    |        |        |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |
| Name   | Reserved     |    | OVFOVF | OVFCNT |    |    |    |    |    |    |    |    |    |    | UNAOVF | UNACNT |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |
| Access |              |    | RO     | RC     |    |    |    |    |    |    |    |    |    |    | RO     | RC     |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |

| Bit   | Name     | Function                                     | Description                                                                                                                                                                                                                                                                                     | Access |
|-------|----------|----------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:29 | Reserved | Reserved                                     | -                                                                                                                                                                                                                                                                                               | -      |
| 28    | OVFOVF   | Overflow Lost Frame Count Overflow           | This bit is set to 1 when the OVFCNT bit of this register counts the overflow of the lost frame counter.<br> <b>Note</b><br>This bit is automatically cleared after reading the OVFCNT bit of this register.   | RO     |
| 27:17 | OVFCNT   | Overflow Lost Frame Statistics               | This bit indicates the number of frames lost due to Rx FIFO overflow and frames that are too short (good frames shorter than 64 bytes).                                                                                                                                                         | RC     |
| 16    | UNAOVF   | Buffer Unavailable Lost Frame Count Overflow | This bit is set to 1 when the UNACNT bit of this register counts the overflow of the lost frame counter.<br> <b>Note</b><br>This bit is automatically cleared after reading the UNACNT bit of this register. | RO     |
| 15:0  | UNACNT   | Buffer Unavailable Lost Frame Statistics     | This bit indicates the number of frames lost due to buffer unavailability (no available receive descriptors).                                                                                                                                                                                   | RC     |



## 38.4.4.6 ETH\_DMA Watchdog Timer Register (ETH\_DMA\_REVWDTR)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |      |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|------|---|---|---|---|---|---|---|
| 0x1024 | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7    | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |      |   |   |   |   |   |   |   |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   | RIWT |   |   |   |   |   |   |   |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   | RW   |   |   |   |   |   |   |   |

| Bit  | Name     | Function               | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | Access |
|------|----------|------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:8 | Reserved | Reserved               | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | -      |
| 7:0  | RIWT     | Receive Watchdog Count | <p>This bit indicates the time value obtained by multiplying the system clock periods by 256, which is the set time value of the watchdog timer.</p> <p>The watchdog timer is triggered by the programmed value after RxDMA completes frame transmission. At this time, the RIS bit of the action status register ETH_DMA_DMASTR is not set to 1 because the corresponding RDES1.RIS bit in the descriptor is set to 1. The RIS bit is set to 1 and the watchdog timer stops only when the watchdog timer expires. The watchdog timer is reset when the RIS bit of the action status register ETH_DMA_DMASTR is set to high level.</p> | RW     |

## 38.4.4.7 ETH\_DMA Transmit Poll Demand Register (ETH\_DMA\_TXPOLLR)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0x1004 | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | TXPOLL       |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access | RW           |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit  | Name   | Function                        | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | Access |
|------|--------|---------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:0 | TXPOLL | Transmit Poll Demand Initiation | <p>Writing any value to this bit causes DMA to read the current descriptor pointed to by the ETH_DMA_CHTXDER register. If the descriptor is unavailable (owned by the host), TxDMA returns to the suspended state and sets the TUS bit of the ETH_DMA_DMASTR register. If the descriptor is available, transmission continues.</p> <p>This register is used to instruct TxDMA to poll the transmit descriptor list and check whether the current descriptor is owned by DMA. If TxDMA is in suspend mode, issue a transmit poll demand command to wake it up. If an underflow error occurs in the transmit frame or the descriptor owned by TxDMA is unavailable, TxDMA enters suspend mode.</p> <p>The user can issue this command at any time. Once the command starts reacquiring the current descriptor from host memory, TxDMA resets it.</p> <div>  <b>Note</b><br/>           This bit reads always returns 0.         </div> | RW     |

## 38.4.4.8 ETH\_DMA Receive Poll Demand Register (ETH\_DMA\_RXPOLLR)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0x1008 | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | RX POLL      |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access | RW           |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit  | Name   | Function                       | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | Access |
|------|--------|--------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:0 | RXPOLL | Receive Poll Demand Initiation | <p>Writing any value to this bit causes DMA to read the current descriptor pointed to by the ETH_DMA_CHRXDER register. If the descriptor is unavailable (owned by the host), RxDMA returns to the suspended state and does not set the RUS bit of the ETH_DMA_DMASTS register. If the descriptor is available, RxDMA returns to the active state.</p> <p>This register is used to instruct RxDMA to poll the receive descriptor list and check for new descriptors to wake RxDMA from the suspended state. RxDMA enters the suspended state only when the descriptor it owns is unavailable.</p> <div>  <b>Note</b><br/>           This bit reads always returns 0.         </div> | RW     |

## 38.4.4.9 ETH\_DMA Transmit Descriptor List Address Register (ETH\_DMA\_TXDLADR)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0x1010 | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | TXDLAD       |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access | RW           |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit  | Name   | Function                          | Description                                                                                                                                                                                                                                                                  | Access |
|------|--------|-----------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:0 | TXDLAD | Transmit Descriptor Start Address | <p>This bit represents the address of the first descriptor in the transmit descriptor list.</p> <div>  <b>Note</b><br/>           The lowest 2 bits read always returns 0.         </div> | RW     |

## 38.4.4.10 ETH\_DMA Receive Descriptor List Address Register (ETH\_DMA\_RXDLADR)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0x100C | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | RXDLAD       |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access | RW           |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit  | Name   | Function                         | Description                                                                                                                                                                                                                                                           | Access |
|------|--------|----------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:0 | RXDLAD | Receive Descriptor Start Address | This bit represents the address of the first descriptor in the receive descriptor list.<br><div>  <b>Note</b><br/>           The lowest 2 bits read always returns 0.         </div> | RW     |

## 38.4.4.11 ETH\_DMA Current Host Transmit Descriptor Register (ETH\_DMA\_CHTXDER)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0x1048 | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | CHTXDE       |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access | RO           |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit  | Name   | Function                            | Description                                                                          | Access |
|------|--------|-------------------------------------|--------------------------------------------------------------------------------------|--------|
| 31:0 | CHTXDE | Transmit Descriptor Address Pointer | This bit points to the start address of the current transmit descriptor read by DMA. | RO     |

## 38.4.4.12 ETH\_DMA Current Host Receive Descriptor Register (ETH\_DMA\_CHRXDER)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0x104C | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | CHRXDE       |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access | RO           |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit  | Name   | Function                           | Description                                                                         | Access |
|------|--------|------------------------------------|-------------------------------------------------------------------------------------|--------|
| 31:0 | CHRXDE | Receive Descriptor Address Pointer | This bit points to the start address of the current receive descriptor read by DMA. | RO     |

## 38.4.4.13 ETH\_DMA Current Host Transmit Buffer Register (ETH\_DMA\_CHTXBF)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0x1050 | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | CHTXBF       |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access | RO           |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit  | Name   | Function                        | Description                                                         | Access |
|------|--------|---------------------------------|---------------------------------------------------------------------|--------|
| 31:0 | CHTXBF | Transmit Buffer Address Pointer | This bit points to the current transmit buffer address read by DMA. | RO     |

## 38.4.4.14 ETH\_DMA Current Host Receive Buffer Register (ETH\_DMA\_CHRXBFR)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0x1054 | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | CHRXBF       |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access | RO           |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit  | Name   | Function                       | Description                                                        | Access |
|------|--------|--------------------------------|--------------------------------------------------------------------|--------|
| 31:0 | CHRXBF | Receive Buffer Address Pointer | This bit points to the current receive buffer address read by DMA. | RO     |

### 38.4.5 ETH\_MMC Register

#### 38.4.5.1 ETH\_MMC MMC Control Register (ETH\_MMC\_MMCCTLR)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |        |        |     |     |     |      |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|--------|--------|-----|-----|-----|------|
| 0x0100 | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5      | 4      | 3   | 2   | 1   | 0    |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |        |        |     |     |     |      |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   | MCPSEL | MCPSET | MCF | ROR | COS | CRST |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   | RW     | RW     | RW  | RW  | RW  | RW   |

| Bit  | Name     | Function                       | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | Access |
|------|----------|--------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:6 | Reserved | Reserved                       | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | -      |
| 5    | MCPSEL   | Counter Preset Value Selection | <ul style="list-style-type: none"> <li>0b0: Preset all MMC statistic register values to "preset half value" (0x7FF0).</li> <li>0b1: Preset all MMC statistic register values to "preset full value" (0xFFF0).</li> </ul> <div>  <b>Note</b> <ul style="list-style-type: none"> <li>The preset full value refers to 0x0000-0x0010=0xFFF0.</li> <li>The preset half value refers to 0x8000-0x0010=0x7FF0.</li> </ul> </div> | RW     |
| 4    | MCPSET   | Counter Preset Value Setting   | After this bit is set to 1, all MMC statistic registers are preset to full or half values based on the setting of the MCPSEL bit of this register. <div>  <b>Note</b> <p>This bit is automatically cleared 1 clock period after being set.</p> </div>                                                                                                                                                                    | RW     |
| 3    | MCF      | Counter Freeze                 | After this bit is set to 1, the values of all MMC statistic counters are frozen and no longer change due to receiving or transmitting frames.                                                                                                                                                                                                                                                                                                                                                               | RW     |
| 2    | ROR      | Counter Read Reset             | After this bit is set to 1, all MMC statistic counters are reset to zero after being read. <div>  <b>Note</b> <ul style="list-style-type: none"> <li>This bit is enable regardless of whether the MCF bit of this register is enable.</li> <li>Reading the lower 8 bits of the statistic counter is considered a read action.</li> </ul> </div>                                                                          | RW     |
| 1    | COS      | Counter Full Stop              | <ul style="list-style-type: none"> <li>0b0: After the counter reaches FFFFH, it resets to zero and resumes counting.</li> <li>0b1: After the counter reaches FFFFH, it stops and no longer counts.</li> </ul>                                                                                                                                                                                                                                                                                               | RW     |

| Bit | Name | Function      | Description                                                                                                                                                                                                                                                         | Access |
|-----|------|---------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 0   | CRST | Counter Reset | <div>After being set to 1, the values of all MMC statistic counters are reset.</div> <div> <b>Note</b><br/>This bit is automatically cleared 1 clock period after being set.</div> | RW     |

## 38.4.5.2 ETH\_MMC Rx Statistics Status Register (ETH\_MMC\_REVSTSR)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |        |        |        |          |    |    |    |    |    |    |   |   |   |   |   |        |        |        |        |        |          |  |  |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|--------|--------|--------|----------|----|----|----|----|----|----|---|---|---|---|---|--------|--------|--------|--------|--------|----------|--|--|
| 0x0104 | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19     | 18     | 17     | 16       | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4      | 3      | 2      | 1      | 0      |          |  |  |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |        |        |        |          |    |    |    |    |    |    |   |   |   |   |   |        |        |        |        |        |          |  |  |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    | RXOEIS | RXLEIS | RXUGIS | Reserved |    |    |    |    |    |    |   |   |   |   |   | RXREIS | RXAEIS | RXCEIS | RXMGIS | RXBGIS | Reserved |  |  |
| Access |              |    |    |    |    |    |    |    |    |    |    |    | RO     | RO     | RO     |          |    |    |    |    |    |    |   |   |   |   |   | RO     | RO     | RO     | RO     | RO     |          |  |  |

| Bit   | Name     | Function                               | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                    | Access |
|-------|----------|----------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:20 | Reserved | Reserved                               | -                                                                                                                                                                                                                                                                                                                                                                                                                                                              | -      |
| 19    | RXOEIS   | Oversize Error Frame Statistics Status | <ul style="list-style-type: none"> <li>0b0: The number of received oversize error frames has not reached half or full value.</li> <li>0b1: The number of received oversize error frames has reached half or full value.</li> </ul> <div>  <b>Note</b><br/>           This bit is automatically cleared after reading the corresponding MMC statistic register.         </div> | RO     |
| 18    | RXLEIS   | Frame Length Error Statistics Status   | <ul style="list-style-type: none"> <li>0b0: The number of received length error frames has not reached half or full value.</li> <li>0b1: The number of received length error frames has reached half or full value.</li> </ul> <div>  <b>Note</b><br/>           This bit is automatically cleared after reading the corresponding MMC statistic register.         </div>   | RO     |
| 17    | RXUGIS   | Unicast Good Frame Statistics Status   | <ul style="list-style-type: none"> <li>0b0: The number of received unicast good frames has not reached half or full value.</li> <li>0b1: The number of received unicast good frames has reached half or full value.</li> </ul> <div>  <b>Note</b><br/>           This bit is automatically cleared after reading the corresponding MMC statistic register.         </div>   | RO     |
| 16:8  | Reserved | Reserved                               | -                                                                                                                                                                                                                                                                                                                                                                                                                                                              | -      |
| 7     | RXREIS   | Dwarf Error Frame Statistics Status    | <ul style="list-style-type: none"> <li>0b0: The number of received dwarf error frames has not reached half or full value.</li> <li>0b1: The number of received dwarf error frames has reached half or full value.</li> </ul> <div>  <b>Note</b><br/>           This bit is automatically cleared after reading the corresponding MMC statistic register.         </div>     | RO     |



| Bit | Name     | Function                                | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                      | Access |
|-----|----------|-----------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 6   | RXAEIS   | Alignment Error Frame Statistics Status | <ul style="list-style-type: none"> <li>0b0: The number of received alignment error frames has not reached half or full value.</li> <li>0b1: The number of received alignment error frames has reached half or full value.</li> </ul> <div>  <b>Note</b><br/>           This bit is automatically cleared after reading the corresponding MMC statistic register.         </div> | RO     |
| 5   | RXCEIS   | CRC Error Frame Statistics Status       | <ul style="list-style-type: none"> <li>0b0: The number of received CRC error frames has not reached half or full value.</li> <li>0b1: The number of received CRC error frames has reached half or full value.</li> </ul> <div>  <b>Note</b><br/>           This bit is automatically cleared after reading the corresponding MMC statistic register.         </div>             | RO     |
| 4   | RXMGIS   | Multicast Good Frame Statistics Status  | <ul style="list-style-type: none"> <li>0b0: The number of received multicast good frames has not reached half or full value.</li> <li>0b1: The number of received multicast good frames has reached half or full value.</li> </ul> <div>  <b>Note</b><br/>           This bit is automatically cleared after reading the corresponding MMC statistic register.         </div> | RO     |
| 3   | RXBGIS   | Broadcast Good Frame Statistics Status  | <ul style="list-style-type: none"> <li>0b0: The number of received broadcast good frames has not reached half or full value.</li> <li>0b1: The number of received broadcast good frames has reached half or full value.</li> </ul> <div>  <b>Note</b><br/>           This bit is automatically cleared after reading the corresponding MMC statistic register.         </div> | RO     |
| 2:0 | Reserved | Reserved                                | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                | -      |

## 38.4.5.3 ETH\_MMC Tx Statistics Status Register (ETH\_MMC\_TRSSTSRR)

| Offset | Bit Position |    |    |    |    |    |    |    |         |        |          |         |         |         |         |          |    |    |    |    |    |    |   |        |        |          |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|---------|--------|----------|---------|---------|---------|---------|----------|----|----|----|----|----|----|---|--------|--------|----------|---|---|---|---|---|---|
| 0x0108 | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23      | 22     | 21       | 20      | 19      | 18      | 17      | 16       | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8      | 7      | 6        | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |         |        |          |         |         |         |         |          |    |    |    |    |    |    |   |        |        |          |   |   |   |   |   |   |
| Name   | Reserved     |    |    |    |    |    |    |    | TXEDEIS | TXUGIS | Reserved | TXCAEIS | TXECEIS | TXLCEIS | TXDEEIS | Reserved |    |    |    |    |    |    |   | TXMGIS | TXBGIS | Reserved |   |   |   |   |   |   |
| Access |              |    |    |    |    |    |    |    | RO      | RO     |          | RO      | RO      | RO      | RO      |          |    |    |    |    |    |    |   | RO     | RO     |          |   |   |   |   |   |   |

| Bit   | Name     | Function                                      | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | Access |
|-------|----------|-----------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:23 | Reserved | Reserved                                      | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | -      |
| 22    | TXEDEIS  | Excessive Delay Error Frame Statistics Status | <ul style="list-style-type: none"> <li>0b0: The number of transmitted excessive delay error frames has not reached half or full value.</li> <li>0b1: The number of transmitted excessive delay error frames has reached half or full value.</li> </ul> <div>  <b>Note</b><br/>           This bit is automatically cleared after reading the corresponding MMC statistic register.         </div> | RO     |
| 21    | TXUGIS   | Unicast Good Frame Statistics Status          | <ul style="list-style-type: none"> <li>0b0: The number of transmitted unicast good frames has not reached half or full value.</li> <li>0b1: The number of transmitted unicast good frames has reached half or full value.</li> </ul> <div>  <b>Note</b><br/>           This bit is automatically cleared after reading the corresponding MMC statistic register.         </div>                 | RO     |
| 20    | Reserved | Reserved                                      | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | -      |
| 19    | TXCAEIS  | Carrier Error Frame Statistics Status         | <ul style="list-style-type: none"> <li>0b0: The number of transmitted carrier error frames has not reached half or full value.</li> <li>0b1: The number of transmitted carrier error frames has reached half or full value.</li> </ul> <div>  <b>Note</b><br/>           This bit is automatically cleared after reading the corresponding MMC statistic register.         </div>               | RO     |

| Bit  | Name     | Function                                          | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | Access |
|------|----------|---------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 18   | TXECEIS  | Excessive Collision Error Frame Statistics Status | <ul style="list-style-type: none"> <li>0b0: The number of transmitted excessive collision error frames has not reached half or full value.</li> <li>0b1: The number of transmitted excessive collision error frames has reached half or full value.</li> </ul> <div>  <b>Note</b><br/>           This bit is automatically cleared after reading the corresponding MMC statistic register.         </div> | RO     |
| 17   | TXLCEIS  | Delay Collision Error Frame Statistics Status     | <ul style="list-style-type: none"> <li>0b0: The number of transmitted delay collision error frames has not reached half or full value.</li> <li>0b1: The number of transmitted delay collision error frames has reached half or full value.</li> </ul> <div>  <b>Note</b><br/>           This bit is automatically cleared after reading the corresponding MMC statistic register.         </div>         | RO     |
| 16   | TXDEEIS  | Deferred Error Frame Statistics Status            | <ul style="list-style-type: none"> <li>0b0: The number of transmitted deferred error frames has not reached half or full value.</li> <li>0b1: The number of transmitted deferred error frames has reached half or full value.</li> </ul> <div>  <b>Note</b><br/>           This bit is automatically cleared after reading the corresponding MMC statistic register.         </div>                     | RO     |
| 15:4 | Reserved | Reserved                                          | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | -      |
| 3    | TXMGIS   | Multicast Good Frame Statistics Status            | <ul style="list-style-type: none"> <li>0b0: The number of transmitted multicast good frames has not reached half or full value.</li> <li>0b1: The number of transmitted multicast good frames has reached half or full value.</li> </ul> <div>  <b>Note</b><br/>           This bit is automatically cleared after reading the corresponding MMC statistic register.         </div>                     | RO     |
| 2    | TXBGIS   | Broadcast Good Frame Statistics Status            | <ul style="list-style-type: none"> <li>0b0: The number of transmitted broadcast good frames has not reached half or full value.</li> <li>0b1: The number of transmitted broadcast good frames has reached half or full value.</li> </ul> <div>  <b>Note</b><br/>           This bit is automatically cleared after reading the corresponding MMC statistic register.         </div>                     | RO     |
| 1:0  | Reserved | Reserved                                          | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | -      |

## 38.4.5.4 ETH\_MMC Rx Interrupt Control Register (ETH\_MMC\_RITCTLR)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |        |        |        |          |    |    |    |    |    |    |   |   |   |   |   |        |        |        |        |        |          |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|--------|--------|--------|----------|----|----|----|----|----|----|---|---|---|---|---|--------|--------|--------|--------|--------|----------|
| 0x010C | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19     | 18     | 17     | 16       | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4      | 3      | 2      | 1      | 0      |          |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |        |        |        |          |    |    |    |    |    |    |   |   |   |   |   |        |        |        |        |        |          |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    | RXOEIM | RXLEIM | RXUGIM | Reserved |    |    |    |    |    |    |   |   |   |   |   | RXREIM | RXAEIM | RXCEIM | RXMGIM | RXBGIM | Reserved |
| Access |              |    |    |    |    |    |    |    |    |    |    |    | RW     | RW     | RW     |          |    |    |    |    |    |    |   |   |   |   |   | RW     | RW     | RW     | RW     | RW     |          |

| Bit   | Name     | Function                                | Description                                                                                                                                                                                                                                      | Access |
|-------|----------|-----------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:20 | Reserved | Reserved                                | -                                                                                                                                                                                                                                                | -      |
| 19    | RXOEIM   | Oversize Error Frame Interrupt Mask     | <ul style="list-style-type: none"> <li>0b0: When RXOEIS is set, oversize error frame MMC statistic interrupts are not masked.</li> <li>0b1: When RXOEIS is set, oversize error frame MMC statistic interrupts are masked.</li> </ul>             | RW     |
| 18    | RXLEIM   | Length Error Frame Interrupt Mask       | <ul style="list-style-type: none"> <li>0b0: When RXLEIS is set, length error frame MMC statistic interrupts are not masked.</li> <li>0b1: When RXLEIS is set, the length error frame MMC statistics interrupt is masked.</li> </ul>              | RW     |
| 17    | RXUGIM   | Unicast Good Frame Interrupt Mask.      | <ul style="list-style-type: none"> <li>0b0: When RXUGIS is set, the unicast good frame MMC statistics interrupt is not masked.</li> <li>0b1: When RXUGIS is set, the unicast good frame MMC statistics interrupt is masked.</li> </ul>           | RW     |
| 16:8  | Reserved | Reserved                                | -                                                                                                                                                                                                                                                | -      |
| 7     | RXREIM   | Dwarf Frame Error Frame Interrupt Mask. | <ul style="list-style-type: none"> <li>0b0: When RXREIS is set, the dwarf frame error frame MMC statistics interrupt is not masked.</li> <li>0b1: When RXREIS is set, the dwarf frame error frame MMC statistics interrupt is masked.</li> </ul> | RW     |
| 6     | RXAEIM   | Alignment Error Frame Interrupt Mask.   | <ul style="list-style-type: none"> <li>0b0: When RXAEIS is set, the alignment error frame MMC statistics interrupt is not masked.</li> <li>0b1: When RXAEIS is set, the alignment error frame MMC statistics interrupt is masked.</li> </ul>     | RW     |
| 5     | RXCEIM   | CRC Error Frame Interrupt Mask.         | <ul style="list-style-type: none"> <li>0b0: When RXCEIS is set, the CRC error frame MMC statistics interrupt is not masked.</li> <li>0b1: When RXCEIS is set, the CRC error frame MMC statistics interrupt is masked.</li> </ul>                 | RW     |
| 4     | RXMGIM   | Multicast Good Frame Interrupt Mask.    | <ul style="list-style-type: none"> <li>0b0: When RXMGIS is set, the multicast good frame MMC statistics interrupt is not masked.</li> <li>0b1: When RXMGIS is set, the multicast good frame MMC statistics interrupt is masked.</li> </ul>       | RW     |

| Bit | Name            | Function                             | Description                                                                                                                                                                                                                                 | Access |
|-----|-----------------|--------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 3   | RXBGIM          | Broadcast Good Frame Interrupt Mask. | <ul style="list-style-type: none"><li>● 0b0: When RXBGIS is set, the broadcast good frame MMC statistics interrupt is not masked.</li><li>● 0b1: When RXBGIS is set, the broadcast good frame MMC statistics interrupt is masked.</li></ul> | RW     |
| 2:0 | <i>Reserved</i> | <i>Reserved</i>                      | -                                                                                                                                                                                                                                           | -      |

## 38.4.5.5 ETH\_MMC Tx Interrupt Control Register (ETH\_MMC\_TITCLR)

| Offset | Bit Position |    |    |    |    |    |    |    |    |         |        |          |         |         |         |         |          |    |    |    |    |    |   |   |   |        |        |          |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|---------|--------|----------|---------|---------|---------|---------|----------|----|----|----|----|----|---|---|---|--------|--------|----------|---|---|---|---|
| 0x0110 | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22      | 21     | 20       | 19      | 18      | 17      | 16      | 15       | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6      | 5      | 4        | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |         |        |          |         |         |         |         |          |    |    |    |    |    |   |   |   |        |        |          |   |   |   |   |
| Name   | Reserved     |    |    |    |    |    |    |    |    | TXEDEIM | TXUGIM | Reserved | TXCAEIM | TXECEIM | TXLCEIM | TXDEEIM | Reserved |    |    |    |    |    |   |   |   | TXMGIM | TXBGIM | Reserved |   |   |   |   |
| Access |              |    |    |    |    |    |    |    |    | RW      | RW     |          | RW      | RW      | RW      | RW      |          |    |    |    |    |    |   |   |   | RW     | RW     |          |   |   |   |   |

| Bit   | Name     | Function                                       | Description                                                                                                                                                                                                                                                        | Access |
|-------|----------|------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:23 | Reserved | Reserved                                       | -                                                                                                                                                                                                                                                                  | -      |
| 22    | TXEDEIM  | Excessive delay error frame interrupt mask     | <ul style="list-style-type: none"> <li>0b0: When TXEDEIS is set, the excessive delay error frame MMC statistics interrupt is not masked.</li> <li>0b1: When TXEDEIS is set, the excessive delay error frame MMC statistics interrupt is masked.</li> </ul>         | RW     |
| 21    | TXUGIM   | Unicast good frame interrupt mask              | <ul style="list-style-type: none"> <li>0b0: When TXUGIS is set, the unicast good frame MMC statistics interrupt is not masked.</li> <li>0b1: When TXUGIS is set, the unicast good frame MMC statistics interrupt is masked.</li> </ul>                             | RW     |
| 20    | Reserved | Reserved                                       | -                                                                                                                                                                                                                                                                  | -      |
| 19    | TXCAEIM  | Carrier error frame interrupt mask             | <ul style="list-style-type: none"> <li>0b0: When TXCAEIS is set, the carrier error frame MMC statistics interrupt is not masked.</li> <li>0b1: When TXCAEIS is set, the carrier error frame MMC statistics interrupt is masked.</li> </ul>                         | RW     |
| 18    | TXECEIM  | Excessive collision error frame interrupt mask | <ul style="list-style-type: none"> <li>0b0: When TXECEIS is set, the excessive collision error frame MMC statistics interrupt is not masked.</li> <li>0b1: When TXECEIS is set, the excessive collision error frame MMC statistics interrupt is masked.</li> </ul> | RW     |
| 17    | TXLCEIM  | Delay collision error frame interrupt mask     | <ul style="list-style-type: none"> <li>0b0: When TXLCEIS is set, the delay collision error frame MMC statistics interrupt is not masked.</li> <li>0b1: When TXLCEIS is set, the delay collision error frame MMC statistics interrupt is masked.</li> </ul>         | RW     |
| 16    | TXDEEIM  | Delay error frame interrupt mask               | <ul style="list-style-type: none"> <li>0b0: When TXDEEIS is set, the delay error frame MMC statistics interrupt is not masked.</li> <li>0b1: When TXDEEIS is set, the delay error frame MMC statistics interrupt is masked.</li> </ul>                             | RW     |
| 15:4  | Reserved | Reserved                                       | -                                                                                                                                                                                                                                                                  | -      |

| Bit | Name            | Function                            | Description                                                                                                                                                                                                                                 | Access |
|-----|-----------------|-------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 3   | TXMGIM          | Multicast good frame interrupt mask | <ul style="list-style-type: none"><li>● 0b0: When TXMGIS is set, the multicast good frame MMC statistics interrupt is not masked.</li><li>● 0b1: When TXMGIS is set, the multicast good frame MMC statistics interrupt is masked.</li></ul> | RW     |
| 2   | TXBGIM          | Broadcast good frame interrupt mask | <ul style="list-style-type: none"><li>● 0b0: When TXBGIS is set, the broadcast good frame MMC statistics interrupt is not masked.</li><li>● 0b1: When TXBGIS is set, the broadcast good frame MMC statistics interrupt is masked.</li></ul> | RW     |
| 1:0 | <i>Reserved</i> | <i>Reserved</i>                     | -                                                                                                                                                                                                                                           | -      |

## 38.4.5.6 ETH\_MMC Rx Unicast Good Frame Statistics Register (ETH\_MMC\_RXUNGFR)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |         |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0x01C4 | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15      | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |         |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RXUNCNT |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RO      |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit   | Name     | Function                 | Description                           | Access |
|-------|----------|--------------------------|---------------------------------------|--------|
| 31:16 | Reserved | Reserved                 | -                                     | -      |
| 15:0  | RXUNCNT  | Unicast Good Frame Count | Number of Unicast Good Frame Received | RO     |

## 38.4.5.7 ETH\_MMC Rx Multicast Good Frame Statistics Register (ETH\_MMC\_RXMUGFR)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |          |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0x0190 | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15       | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |          |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RXMUGCNT |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RO       |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit   | Name     | Function                   | Description                             | Access |
|-------|----------|----------------------------|-----------------------------------------|--------|
| 31:16 | Reserved | Reserved                   | -                                       | -      |
| 15:0  | RXMUGCNT | Multicast Good Frame Count | Number of Multicast Good Frame Received | RO     |



## 38.4.5.8 ETH\_MMC Rx Broadcast Good Frame Statistics Registers (ETH\_MMC\_RXBRGFR)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |         |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0x018C | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15      | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |         |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RXBRCNT |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RO      |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit   | Name     | Function                   | Description                             | Access |
|-------|----------|----------------------------|-----------------------------------------|--------|
| 31:16 | Reserved | Reserved                   | -                                       | -      |
| 15:0  | RXBRGCNT | Broadcast Good Frame Count | Number of Broadcast Good Frame Received | RO     |

## 38.4.5.9 ETH\_MMC Rx CRC Error Frame Statistics Register (ETH\_MMC\_RXCREFR)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |          |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0x0194 | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15       | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |          |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RXCRECNT |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RO       |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit   | Name     | Function              | Description                        | Access |
|-------|----------|-----------------------|------------------------------------|--------|
| 31:16 | Reserved | Reserved              | -                                  | -      |
| 15:0  | RXCRECNT | CRC Error Frame Count | Number of CRC Error Frame Received | RO     |

## 38.4.5.10 ETH\_MMC Rx Alignment Error Frame Statistics Registers (ETH\_MMC\_RXALEFR)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |          |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0x0198 | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15       | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |          |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RXALECNT |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RO       |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit   | Name     | Function                    | Description                              | Access |
|-------|----------|-----------------------------|------------------------------------------|--------|
| 31:16 | Reserved | Reserved                    | -                                        | -      |
| 15:0  | RXALECNT | Alignment Error Frame Count | Number of Alignment Error Frame Received | RO     |

## 38.4.5.11 ETH\_MMC Rx Dwarf Frame Error Statistics Registers (ETH\_MMC\_RXRUEFR)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |          |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0x019C | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15       | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |          |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RXRUECNT |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RO       |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit   | Name     | Function                | Description                          | Access |
|-------|----------|-------------------------|--------------------------------------|--------|
| 31:16 | Reserved | Reserved                | -                                    | -      |
| 15:0  | RXRUECNT | Dwarf Frame Error Count | Number of Dwarf Frame Error Received | RO     |

## 38.4.5.12 ETH\_MMC Rx Length Error Frame Statistics Registers (ETH\_MMC\_RXLEEFR)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |          |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0x01C8 | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15       | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |          |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RXLEECNT |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RO       |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit   | Name     | Function                 | Description                           | Access |
|-------|----------|--------------------------|---------------------------------------|--------|
| 31:16 | Reserved | Reserved                 | -                                     | -      |
| 15:0  | RXLEECNT | Length Error Frame Count | Number of Length Error Frame Received | RO     |

## 38.4.5.13 ETH\_MMC Rx Oversize Error Frame Statistics Registers (ETH\_MMC\_RXOREFR)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |          |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0x01CC | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15       | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |          |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RXORECNT |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RO       |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit   | Name     | Function                   | Description                             | Access |
|-------|----------|----------------------------|-----------------------------------------|--------|
| 31:16 | Reserved | Reserved                   | -                                       | -      |
| 15:0  | RXORECNT | Oversize Error Frame Count | Number of Oversize Error Frame Received | RO     |

## 38.4.5.14 ETH\_MMC Tx Unicast Good Frame Statistics Register (ETH\_MMC\_TXUNGFR)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |         |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0x0168 | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15      | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |         |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | TXUNCNT |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RO      |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit   | Name     | Function                 | Description                              | Access |
|-------|----------|--------------------------|------------------------------------------|--------|
| 31:16 | Reserved | Reserved                 | -                                        | -      |
| 15:0  | TXUNCNT  | Unicast Good Frame Count | Number of Transmitted Unicast Good Frame | RO     |

## 38.4.5.15 ETH\_MMC Tx Multicast Good Frame Statistics Register (ETH\_MMC\_TXMUGFR)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |          |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0x0120 | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15       | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |          |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | TXMUGCNT |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RO       |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit   | Name     | Function                   | Description                                | Access |
|-------|----------|----------------------------|--------------------------------------------|--------|
| 31:16 | Reserved | Reserved                   | -                                          | -      |
| 15:0  | TXMUGCNT | Multicast Good Frame Count | Number of Transmitted Multicast Good Frame | RO     |

## 38.4.5.16 ETH\_MMC Tx Broadcast Good Frame Statistics Register (ETH\_MMC\_TXBRGFR)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |          |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0x011C | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15       | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |          |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | TXBRGCNT |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |          |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit   | Name     | Function                   | Description                                | Access |
|-------|----------|----------------------------|--------------------------------------------|--------|
| 31:16 | Reserved | Reserved                   | -                                          | -      |
| 15:0  | TXBRGCNT | Broadcast Good Frame Count | Number of Transmitted Broadcast Good Frame | RO     |

## 38.4.5.17 ETH\_MMC Tx Deferred Error Frame Statistics Register (ETH\_MMC\_TXDEEFR)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |          |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0x0154 | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15       | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |          |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | TXDEECNT |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |          |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit   | Name     | Function                   | Description                                | Access |
|-------|----------|----------------------------|--------------------------------------------|--------|
| 31:16 | Reserved | Reserved                   | -                                          | -      |
| 15:0  | TXDEECNT | Deferred Error Frame Count | Number of Transmitted Deferred Error Frame | RO     |

## 38.4.5.18 ETH\_MMC Tx Carrier Error Frame Statistics Register (ETH\_MMC\_TXCAEFR)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |          |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |  |  |  |  |  |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|--|--|--|--|--|--|
| 0x0160 | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15       | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |  |  |  |  |  |  |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |          |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |  |  |  |  |  |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | TXCAECNT |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |  |  |  |  |  |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RO       |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |  |  |  |  |  |

| Bit   | Name     | Function                  | Description                               | Access |
|-------|----------|---------------------------|-------------------------------------------|--------|
| 31:16 | Reserved | Reserved                  | -                                         | -      |
| 15:0  | TXCAECNT | Carrier Error Frame Count | Number of Transmitted Carrier Error Frame | RO     |

## 38.4.5.19 ETH\_MMC Tx Delay Collision Error Frame Statistics Register (ETH\_MMC\_TXLCEFR)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |          |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |  |  |  |  |  |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|--|--|--|--|--|--|
| 0x0158 | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15       | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |  |  |  |  |  |  |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |          |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |  |  |  |  |  |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | TXLCECNT |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |  |  |  |  |  |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RO       |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |  |  |  |  |  |

| Bit   | Name     | Function                          | Description                                       | Access |
|-------|----------|-----------------------------------|---------------------------------------------------|--------|
| 31:16 | Reserved | Reserved                          | -                                                 | -      |
| 15:0  | TXLCECNT | Delay Collision Error Frame Count | Number of transmitted Delay Collision Error Frame | RO     |

## 38.4.5.20 ETH\_MMC Tx Excessive Collision Error Frame Statistics Register (ETH\_MMC\_TXECEFR)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |         |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0x015C | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15      | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |         |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | TXECCNT |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RO      |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit   | Name     | Function                              | Description                                           | Access |
|-------|----------|---------------------------------------|-------------------------------------------------------|--------|
| 31:16 | Reserved | Reserved                              | -                                                     | -      |
| 15:0  | TXECECNT | Excessive Collision Error Frame Count | Number of Transmitted Excessive Collision Error Frame | RO     |

## 38.4.5.21 ETH\_MMC Tx Excessive Delay Error Frame Statistics Register (ETH\_MMC\_TXEDEFR)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |          |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0x016C | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15       | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |          |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | TXEDECNT |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RO       |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit   | Name     | Function                          | Description                                       | Access |
|-------|----------|-----------------------------------|---------------------------------------------------|--------|
| 31:16 | Reserved | Reserved                          | -                                                 | -      |
| 15:0  | TXEDECNT | Excessive Delay Error Frame Count | Number of Transmitted Excessive Delay Error Frame | RO     |

## 39 SDIO Controller (SDIOC)

### 39.1 Introduction

SDIOC provides an SD host interface and an MMC host interface for communicating with SD cards supporting the SD2.0 protocol, SDIO devices and MMC devices supporting the eMMC4.2 protocol. This product has 2 SDIO controllers, which can communicate with 2 SD/MMC/SDIO devices at the same time.

### 39.2 Features

- Support SDSC, SDHC, SDXC format SD card and SDIO device
- Supports 1-bit and 4-bit SD bus
- Supports 1-bit, 4-bit and 8-bit MMC buses
- SD clock up to 50MHz
- With card identification and hardware write protection function



## 39.3 Functional Description

### 39.3.1 Port Assignment

Table 39-1 Port Allocation Table

| Port Name | IO  | Function                             |
|-----------|-----|--------------------------------------|
| SDIOx_CK  | O   | SD clock output signal               |
| SDIOx_CMD | I/O | SD command and response signal       |
| SDIOx_D0  | I/O | SD data signal                       |
| SDIOx_D1  | I/O | SD data signal                       |
| SDIOx_D2  | I/O | SD data signal                       |
| SDIOx_D3  | I/O | SD data signal                       |
| SDIOx_D4  | I/O | SD data signal                       |
| SDIOx_D5  | I/O | SD data signal                       |
| SDIOx_D6  | I/O | SD data signal                       |
| SDIOx_D7  | I/O | SD data signal                       |
| SDIOx_CD  | I   | SD card identification status signal |
| SDIOx_WP  | I   | SD write protection status signal    |



#### Note

x represents the unit number, which is 1-2. When x is mentioned in this chapter later, it refers to the unit number and will not be repeated here.

The above ports are based on the SD2.0 protocol and the eMMC4.51 protocol. The SD clock output signal (SDIOx\_CK) is used to output the SD clock when the SDIOC communicates with the SD/MMC device. SD data signal (SDIOx\_Dy (y=0-7) ) is used for SDIOC and devices to send and receive data during communication. When one-wire (1-bit) communication is used, only SDIOx\_D0 is enabled, and SDIOx\_Dy (y=1-7) remains high. When four-wire (4-bit) communication is used, SDIOx\_Dy (y=0-3) is enabled, and SDIOx\_Dy (y=4-7) remains high. When eight-wire (8-bit) communication is used, SDIOx\_Dy (y=0-7) is enable. Eight-wire (8-bit) communication is limited to MMC device communication.

### 39.3.2 Basic Access Method

The user starts the SDIOC to communicate with the off-chip SD/MMC device by reading and writing the SDIOC register. Since writing the command register will trigger the SDIOC's send command action, writing the command register must be done last. Before this, the user needs to set the transmission mode through the transmission mode register (TRANSMODE), set the command parameters through the parameter registers (ARG0, ARG1), and set the correct command number (command index), type, response (response) type, etc. Once the write command register is executed, the SDIOC will send the command, and the user can read the interrupt status register (NORINTST, ERRINTST) to check whether the response information has been received, whether there is an error message, etc. When a command is executed, the user can get the response of the command by reading the response register (RESP0-7).

### 39.3.3 Data Transmission

The data buffer register of SDIOC is used for data exchange between host devices such as CPU/DMA and SD devices. SDIOC has built-in FIFO to speed up data exchange. When the command includes sending data, the user will write the data into the data buffer registers (BUF0, BUF1) sequentially, and these data will be buffered in the FIFO of SDIOC first, when the amount of written data reaches the set value of the data length register (BLKSIZE) or 512 bytes, SDIOC will send data through SDIOx\_Dy (y=0-7). During this period, SDIOC can continue to write data to the data buffer register through FIFO. Similarly, when the command includes data reception, SDIOC receives data through SDIOx\_Dy (y=0-7), and buffers the data in the BUFFER of SDIOC, and the user reads the data buffer registers (BUF0, BUF1) get data. For the data format, please refer to the SD and MMC protocols.

### 39.3.4 SD Clock

The SD clock (SDIOx\_CK) is generated by the host and output to the device for communication between the two. The SD clock is generated by dividing the action clock (PCLK1) of the SDIOC module through a divider. The frequency division coefficient of the frequency divider is set by CLKCON register. According to the requirements of the SD2.0 protocol, the fastest value should be 50MHz in the data transfer mode, so the user needs to set it according to the frequency of PCLK1 reasonable frequency division coefficient.

### 39.3.5 Interrupts and DMA Start Requests

#### SD Interrupt

SDIOC provides two types of interrupts, normal interrupt and error interrupt. Normal interrupts are interrupts that occur when various events occur during the communication between SDIOC and SD/MMC card. Error interrupts are interrupts that are generated when various errors occur during SDIOC operation communication. Normal interrupts and error interrupts have their own interrupt status register, interrupt status enable register, and interrupt signal enable register, respectively.

- The interrupt status register is used to indicate the cause of the interrupt.
- The interrupt status enable register is used to enable each status bit of the interrupt status register. Each status bit of the interrupt status register needs to be active when the enable bit of the interrupt status enable register is enabled.
- The interrupt signal enable register is used to Enable Each interrupt factor to apply for an interrupt to the CPU.

#### SDIO Interrupt

The SDIO device can send a card interrupt request to the host when the transmission is idle. When using SDIO interrupt, you need to enable the NORINTSTEN.CINTEN bit. If you need to apply for an interrupt to the CPU, you also need to enable the NORINTSGEN.CINTEN bit. The SDIO device requests an interrupt from the host by pulling the SDIOx\_D1 data line low. SDIOC detects that SDIOx\_D1 is pulled low, sets the NORINTST.CINT bit, and applies for an interrupt to the CPU according to the setting. The SDIO device will release SDIOx\_D1 after judging that the interrupt processing is completed.

#### DMA Request

SDIOC can read and write data in communication through DMA. When using DMA to transfer and write data to the device, set the start source of a channel of DMA to the write request of SDIOC, and then set the transfer target address of DMA to the data buffer register (BUF0, BUF1) and it is a fixed address. When the SDIOC sends the write data command, if the data FIFO is empty at this time, a write request signal is sent to the DMA, and the DMA is started to write data to the data buffer register (BUF0, BUF1),

and these data will enter the data FIFO in turn. When the data in the FIFO reaches the set value of the data length register (BLKSIZE) or 512 bytes, SDIOC will send the data through SDIOx\_Dy (y=0-7). If it is a multi-block write command, SDIOC will continue to send a start request to the DMA to write data at the same time. When using DMA to transmit data read from the device, set the start source of another channel of DMA to the read request of SDIOC, and then set the transmission source address of DMA to the data buffer register (BUF0, BUF1) and it is a fixed address. After SDIOC sends the read data command, the device will send data through SDIOx\_Dy (y=0-7), and the data FIFO of SDIOC starts to receive data. When the data in FIFO reaches the set value of the data length register (BLKSIZE) or 512 bytes, SDIOC sends a read request signal to DMA, and starts DMA to read data from the data buffer register (BUF0, BUF1). If it is a multi-block read command, SDIOC will continue to read the next data block from the device at the same time.

### Card Insert and Removal

The insert and removal of SD/MMC/SDIO devices are identified by the SDIOx\_CD signal line. When there is no device in the card slot, the card slot will pull up the SDIOx\_CD signal through a resistor. When a device is inserted, the SDIOx\_CD signal will be pulled low, and will become high again after the device is removed. SDIOC judges whether there is a device by the level of SDIOx\_CD. Users can judge whether there is a device plugged in by reading the PSTAT.CDL bit. SDIOC can generate corresponding interrupts when the device is inserted and removed, and is enabled through the NORINTSTEN register and NORINTSGEN register.

## 39.4 Operation Example

### 39.4.1 Host and Device Initialization

#### 39.4.1.1 Host Initialization

##### Context

SDIOC needs to be initialized first when using it.

##### Procedure

- Step1.** Read PSTAT register, query clock status and device insertion status.
- Step2.** Configure PWRCON register, enable SDIOC.
- Step3.** Configure CLKCON register to make SDIOC output the SD clock, and configure the frequency division of the SD clock according to the frequency of PCLK1 to ensure that the frequency of the SD clock in the card identification mode does not exceed 400kHz.
- Step4.** Configure HOSTCON register, select 1-bit mode and disable high speed mode.
- Step5.** Configure TOUTCON register according to the device characteristics, so that the host ends the communication and reports an error when the communication times out.
- Step6.** Configure normal and error interrupt status enable registers, interrupt signal enable register to enable SDIOC to interrupt when needed and set the flag bit.

#### 39.4.1.2 SD Card Initialization

##### Context

After completing the SDIOC initialization configuration, if an SD card is connected, it needs to be initialized according to the SD protocol.

##### Procedure

- Step1.** Card reset, send reset command CMD0 to the device, CMD0 has no response information, at this time the card will enter the card identification mode.

- Step2.** Confirm the operation condition of the card, send CMD8 to the device and wait for the response information.
- Step3.** Send the initialization command ACMD41 (send CMD55 first, then send CMD41), judge whether the device is initialized according to the response information, otherwise continue to send ACMD41 until the initialization is completed.
- Step4.** Send CMD2 to get the CID information of the card.
- Step5.** Send CMD3 to get the RCA information of the card.

#### Results

At this point the card will enter the data transfer mode and the initialization is complete.

#### 39.4.1.3 MMC Card Initialization

##### Context

After completing the SDIOC initialization configuration, if an MMC card is connected, it needs to be initialized according to the MMC protocol.

##### Procedure

- Step1.** Card reset, send reset command CMD0 to the device, CMD0 has no response information, at this time the card will enter the card identification mode.
- Step2.** Confirm the operation condition of the card, send CMD1 to the device and wait for the response message, judge whether the device is initialized according to the response message, otherwise continue to send CMD1 until the initialization is completed.
- Step3.** Send CMD2 to get the CID information of the card.
- Step4.** Send CMD3 to get the RCA information of the card.

#### Results

At this point the card will enter the data transfer mode and the initialization is complete.

#### 39.4.1.4 SDIO Initialization

##### Context

After completing the SDIOC initialization configuration, if an SDIO device is connected, it needs to be initialized according to the SDIO protocol.

##### Procedure

- Step1.** Device reset, send reset command CMD0 to the device, CMD0 has no response information.
- Step2.** Confirm the operation condition of the SDIO device, send CMD5 to the device and wait for the response information.
- Step3.** Send CMD3 to get the RCA information of the device, and the initialization is complete.

#### 39.4.2 SD/MMC Single Block Read and Write

##### Context

After the SD/MMC card enters the data transmission mode, SDIOC can access the data of the SD/MMC card through read and write commands. The order of reading and writing a single data block is as follows:

##### Procedure

- Step1.** Configure CLKCON register to make SDIOC output the SD clock, and configure the frequency division of the SD clock according to the frequency of PCLK1 to ensure that the frequency of the SD clock in the default speed mode does not exceed the maximum clock speed (Default Speed mode,  $f_{pp} \leq 25\text{MHz}$ ).
- Step2.** Send CMD7, SD/MMC card will enter the data transmission state.

- Step3.** If necessary, for the SD card, configure HOSTCON register to set the host bus width, and send ACMD6 to set the SD bus width (1-bit or 4-bit), and for the MMC card, configure HOSTCON register to set the host bus width, and rewrite the Ext\_CSD register of the MMC card through the CMD6 (SWITCH) command to set the bus width (1-bit, 4-bit or 8-bit).
- Step4.** If necessary, for the SD card, you can configure BLKSIZE register to set the data block length, and set the size (bytes) of the data block through the CMD16 command, and the configuration range is 1-512 bytes. For SDHC/SDXC and MMC cards, the data block size is fixed at 512 bytes.
- Step5.** If necessary, for the SD card, configure HOSTCON register to set the host to high speed mode, and send CMD6 to switch the SD card to high speed mode ( $f_{pp} \leq 50\text{MHz}$ ). SDIOx\_CK can be set up to 50MHz after switch success.
- Step6.** When writing data, first configure TRANSMODE register, and set the transfer mode to write, single block transfer. Send single block write command CMD24. If using the CPU to write data, after confirming that the NORINTST.BWR bit is 1, write data to the data buffer register (BUF0, BUF1). If using DMA to write data, wait for the DMA transfer to complete, SDIOC will send data through the data signal SDIOx\_Dy. After the transmission is completed, the NORINTST.TC bit will be set to 1 and an interrupt request will be generated. If an error occurs during the transmission process, the corresponding error flag will be set and an interrupt request will be generated.
- Step7.** When reading data, first configure TRANSMODE register, and set the transfer mode to read, single block transfer. Send single block read command CMD17, SDIOC will receive data through data signal SDIOx\_Dy. If using the CPU to read data, after confirming that the NORINTST.BRR bit is 1, read the data from the data buffer register (BUF0, BUF1). If using DMA to read data, wait for the DMA transfer to complete, after the reading is completed, the NORINTST.TC bit will be set to 1 and an interrupt request will be generated. If an error occurs during the transmission process, the corresponding error flag will be set and an interrupt request will be generated.

### 39.4.3 SD/MMC Multi Block Read and Write

#### Procedure

- Step1.** Configure CLKCON register to make SDIOC output the SD clock, and configure the frequency division of the SD clock according to the frequency of PCLK1 to ensure that the frequency of the SD clock in the default speed mode does not exceed the maximum clock speed (Default Speed mode,  $f_{pp} \leq 25\text{MHz}$ ).
- Step2.** Send CMD7, SD/MMC card will enter the data transmission state.
- Step3.** If necessary, for the SD card, configure HOSTCON register to set the host bus width, and send ACMD6 to set the SD bus width (1-bit or 4-bit), and for the MMC card, configure HOSTCON register to set the host bus width, and rewrite the Ext\_CSD register of the MMC card through the CMD6 (SWITCH) command to set the bus width (1-bit, 4-bit or 8-bit).
- Step4.** If necessary, for the SD card, you can configure BLKSIZE register to set the data block length, and set the size (bytes) of the data block through the CMD16 command, and the configuration range is 1-512 bytes. For SDHC/SDXC and MMC cards, the data block size is fixed at 512 bytes.
- Step5.** Configure TRANSMODE register, set the transfer mode to read/write, and multi-block transfer. If the data block count is enabled, the number of data blocks to be transmitted need to be set in BLKCNT register, multi-block transmission can not start when the BLKCNT register is not

set. If the data block counting enable is disabled, there is no need to set BLKCNT register, meanwhile, an unlimited number of multi-block transmissions can be performed. When the host decides to terminate the transmission, it needs to send a CMD12 to the device after the completion of the last data block transmission to notify the device that the data transmission has ended.

- Step6.** If necessary, for the SD card, configure HOSTCON register to set the host to high speed mode, and send CMD6 to switch the SD card to high speed mode ( $f_{pp} \leq 50\text{MHz}$ ). SDIOx\_CK can be set up to 50MHz after switch success.
- Step7.** When writing data, first send the multi-block write command CMD25. If using the CPU to write data, after confirming that the NORINTST.BWR bit is 1, write data to the data buffer register (BUF0, BUF1). During the sending process, BWR will keep 0, reset to 1 after sending, the user can write the data of the second block at this time. If using DMA to write data, wait for the DMA transfer to complete, SDIOC will send data through the data signal SDIOx\_Dy. The NORINTST.TC bit will be set to 1 and an interrupt request will be generated. If an error occurs in the transmission process, the corresponding error flag will be set and an interrupt request will be generated. After all the data is sent, if the automatic sending of CMD12 is set, SDIOC will automatically send a CMD12 to end the transmission, otherwise it is necessary to manually send CMD12 to the device to notify the device that the data transmission has ended.
- Step8.** When reading data, send the multi-block read command CMD18, and SDIOC will receive data through the data signal SDIOx\_Dy. If using the CPU to read data, after confirming that the NORINTST.BRR bit is 1, read the data from the data buffer register (BUF0, BUF1). The data of the second block can be read at this time. If using DMA to read data, wait for the DMA transfer to complete, after the reading is completed, the NORINTST.TC bit will be set to 1 and an interrupt request will be generated. If an error occurs during the transmission process, the corresponding error flag will be set and an interrupt will be generated. After all the data is received, if the automatic sending of CMD12 is set, the SDIOC will automatically send a CMD12 to end the transmission, otherwise it is necessary to manually send a CMD12 to the device to notify the device that the data transmission has ended.

#### 39.4.4 Transfer Abort, Suspend and Resume

When performing multi-block transfer, it can be aborted or suspended by software control. The abort operation can be performed regardless of whether or not the number of transfer data blocks is set. Abort operations are divided into asynchronous abort and synchronous abort. The asynchronous abort operation needs to send CMD12 to terminate the transmission by writing CMD register while the transmission is in progress, and the transmission will be terminated immediately at this time. For the write operation, the SD/MMC card will discard the current data block data and enter the program mode, and write the previously received data block into the FLASH. For read operations, SD/MMC will stop transferring data. Synchronous abort means that the transmission stops at the data block interval by setting the BLKGAP registe during the transmission. After the setting is completed, the transmission will stop after the current data block transmission ends. At this time, CMD12 needs to be sent to end the transmission.

When executing the suspend operation, first set the BLKGAP.SABGR bit to stop the transmission at the data block interval, and enable the read wait function. After the setting is complete, the transmission will stop transmission after the current data block transmission ends. At this time, send CMD52 by writing CMD register to suspend the transmission. After sending the suspend command, it is need to continue to read the CCCR.BS bit of SDIO through CMD52. If BS=0, it means that the transmission is suspended, and the SD bus turns to an idle state. After the transfer is suspended, the host can perform



operations on other functions of SDIO. But if you want to resume operation later, you need to back up the SDIOC register (offset address 0x00-0x0D) after suspending, and restore these register settings after performing other operations. When performing a resume operation, first clear the BLKGAP register to stop the transmission at the data block interval, and then send CMD52 to resume transmission by writing CMD register. After sending the recovery command, it is necessary to continue to read the CCCR.DF bit of SDIO through CMD52. If DF=1, it means that there is data to be transmitted after the recovery is executed. If DF=0, it means that there is no data to be transmitted. At this time, the data line should be reset by writing SFTRST register.

**Notice**

Suspend and resume operations require SDIO devices and combo card devices to support such operations and read wait operation. Access to the CIA area (common I/O area) of SDIO does not support suspend and resume operations, and only the read wait function can be used.

### 39.4.5 Read Wait

Read wait operation allows the host to insert CMD52 during continuous data transfers to access other functions of the SDIO device. When performing read wait, firstly, the transmission is stopped at the data block interval by setting the BLKGAP.SABGR bit, and the read wait operation is enabled. After the setting is completed, the host will pull the SDIOx\_D2 data line low after the current data block transmission is completed. The transmission will be suspended at this time, and the host can insert the CMD52 at this time to access other functions that do not require data transmission. When it is necessary to end the read wait operation, set the BLKGAP.CR bit and clear SABGR to resume transmission.

**Notice**

Read wait operation requires support from SDIO devices and combo card devices, and needs to be performed under four-wire bus transmission.

### 39.4.6 Wake-up

When not working for a long time, the system can be transferred to a low power consumption state to reduce power consumption. In the low power consumption state, the system can be woken up by SD/MMC/SDIO device insert/removal and card interrupt to continue working. When using the wake-up function, you need to enable the corresponding enable bit in NORINTSTEN register and NORINTSGEN register for insert/removal or card interrupt. At the same time, enable the corresponding wake-up function of the SDIOx\_CD/ SDIOx\_D1 port. After the configuration is complete, the system can be transferred to a low power consumption mode. When insert/removal and card interrupt, the wake-up function of the SDIOx\_CD/SDIOx\_D1 port will wake up the system and generate interrupts of insert/removal and card interrupt.

#### Use the Insert to Wake-up the Stop Mode Process

- Step1.** Configure the SDIOC port, select SDIOx\_CD as PA10.
- Step2.** Configure the PCR register of PA10, and select EIRQ10 to be active.
- Step3.** Configure the low power consumption mode of the power control module to stop mode. For configuration methods, please refer to the relevant description in the "Power Control (PWC)" chapter.
- Step4.** Configure the interrupt control module WUPEN register to enable the wake-up function of EIRQ10.

- Step5.** Configure the EIRQCR10 register of the interrupt control module and select the falling edge trigger.
- Step6.** Configure PWRCON to enable SDIOC.
- Step7.** Configure NORINTSTEN register and NORINTSGEN register to enable the corresponding enable bit of insert.
- Step8.** Configure the interrupt source selection register to select the SDIOC interrupt as the interrupt source and enable the interrupt.
- Step9.** Execute the WFI command to put the system into Stop mode.
- Step10.** When the device is plugged in, PA10 will be pulled low, wake up the CPU through EIRQ10, and enter the interrupt subroutine according to the SD interrupt request.
- Step11.** Clear the corresponding bit status of insert in NORINTST register, exit the interrupt subroutine, and perform subsequent operations.

#### Use the Insert to Wake-up the Power-down Mode Process

- Step1.** When the device is inserted, PA10 will be pulled low, and the system will be woken up through EIRQ10. Re-power on and configure the SDIOC port, and select SDIOx\_CD as PA10.
- Step2.** Configure the PWRC0 registers of the power control module to enable the CPU to enter the Power-down mode after executing the WFI command.
- Step3.** Configure the low power consumption mode of the power control module to Power-down mode. For configuration methods, please refer to the relevant description in the "Power Control (PWC)" chapter.
- Step4.** Configure the PDWKES register of the power control module and select the falling edge.
- Step5.** Execute the WFI command to put the system into Power-down mode.
- Step6.** When the device is plugged in, PA10 will be pulled low, wake up the system through WKUP2\_2 and power on again.



## 39.5 Register Description

The SDIOC module is designed according to the SD Host Controller Standard Specification, so the registers are the same as the standard description, and the unused addresses and bits are changed to reserved bits (Reserved).

### 39.5.1 Register List

SDIOC System Control Register base address: 0x40055400

Table 39-2 SDIOC System Control Register Offset

| Offset | Register                       | Description                   | Width |
|--------|--------------------------------|-------------------------------|-------|
| 0x04   | <a href="#">SDIOC_SYCTLREG</a> | SDIOC System Control Register | 32    |

- SDIOC1 base address: 0x40070000
- SDIOC2 base address: 0x40078400

Table 39-3 SDIO Controller (SDIOC) Register Offset

| Offset | Register                        | Description                | Width |
|--------|---------------------------------|----------------------------|-------|
| 0x04   | <a href="#">SDIOC_BLKSIZE</a>   | Data Block Size Register   | 16    |
| 0x06   | <a href="#">SDIOC_BLKCNT</a>    | Data Block Count Register  | 16    |
| 0x08   | <a href="#">SDIOC_ARG0</a>      | Argument Register 0        | 16    |
| 0x0A   | <a href="#">SDIOC_ARG1</a>      | Argument Register 1        | 16    |
| 0x0C   | <a href="#">SDIOC_TRANSMODE</a> | Transmission Mode Register | 16    |
| 0x0E   | <a href="#">SDIOC_CMD</a>       | Command Register           | 16    |
| 0x10   | <a href="#">SDIOC_RESP0</a>     | Response Register 0        | 16    |
| 0x12   | <a href="#">SDIOC_RESP1</a>     | Response Register 1        | 16    |
| 0x14   | <a href="#">SDIOC_RESP2</a>     | Response Register 2        | 16    |
| 0x16   | <a href="#">SDIOC_RESP3</a>     | Response Register 3        | 16    |
| 0x18   | <a href="#">SDIOC_RESP4</a>     | Response Register 4        | 16    |
| 0x1A   | <a href="#">SDIOC_RESP5</a>     | Response Register 5        | 16    |
| 0x1C   | <a href="#">SDIOC_RESP6</a>     | Response Register 6        | 16    |
| 0x1E   | <a href="#">SDIOC_RESP7</a>     | Response Register 7        | 16    |
| 0x20   | <a href="#">SDIOC_BUF0</a>      | Data Buffer Register 0     | 16    |
| 0x22   | <a href="#">SDIOC_BUF1</a>      | Data Buffer Register 1     | 16    |
| 0x24   | <a href="#">SDIOC_PSTAT</a>     | Host Status Register       | 32    |
| 0x28   | <a href="#">SDIOC_HOSTCON</a>   | Host Control Register      | 8     |
| 0x29   | <a href="#">SDIOC_PWRCON</a>    | Power Control Register     | 8     |

| Offset | Register                         | Description                                       | Width |
|--------|----------------------------------|---------------------------------------------------|-------|
| 0x2A   | <a href="#">SDIOC_BLKGP CON</a>  | Data Block Gap Control Register                   | 8     |
| 0x2C   | <a href="#">SDIOC_CLKCON</a>     | Clock Control Register                            | 16    |
| 0x2E   | <a href="#">SDIOC_TOUTCON</a>    | Timeout Control Register                          | 8     |
| 0x2F   | <a href="#">SDIOC_SFTRST</a>     | Software Reset Register                           | 8     |
| 0x30   | <a href="#">SDIOC_NORINTST</a>   | Normal Interrupt Status Register                  | 16    |
| 0x32   | <a href="#">SDIOC_ERRINTST</a>   | Error Interrupt Status Register                   | 16    |
| 0x34   | <a href="#">SDIOC_NORINTSTEN</a> | Normal Interrupt Status Enable Register           | 16    |
| 0x36   | <a href="#">SDIOC_ERRINTSTEN</a> | Error Interrupt Status Enable Register            | 16    |
| 0x38   | <a href="#">SDIOC_NORINTSGEN</a> | Normal Interrupt Signal Enable Register           | 16    |
| 0x3A   | <a href="#">SDIOC_ERRINTSGEN</a> | Error Interrupt Signal Enable Register            | 16    |
| 0x3C   | <a href="#">SDIOC_ATCERRST</a>   | Auto-Command Error Status Register                | 16    |
| 0x50   | <a href="#">SDIOC_FEA</a>        | Forced Auto-Command Error Status Control Register | 16    |
| 0x52   | <a href="#">SDIOC_FEE</a>        | Forced Error Status Control Register              | 16    |

### 39.5.2 Data Block Size Register (SDIOC\_BLKSIZE)

| Offset | Bit Position |    |    |    |     |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|-----|----|---|---|---|---|---|---|---|---|---|---|
| 0x04   | 15           | 14 | 13 | 12 | 11  | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x0000       |    |    |    |     |    |   |   |   |   |   |   |   |   |   |   |
| Name   | Reserved     |    |    |    | TBS |    |   |   |   |   |   |   |   |   |   |   |
| Access |              |    |    |    |     |    |   |   |   |   |   |   |   |   |   |   |
|        |              |    |    |    | RW  |    |   |   |   |   |   |   |   |   |   |   |

| Bit   | Name     | Function            | Description                                                                                  | Access |
|-------|----------|---------------------|----------------------------------------------------------------------------------------------|--------|
| 15:12 | Reserved | Reserved            | -                                                                                            | -      |
| 11:0  | TBS      | Transfer block size | Set the length of the data block to be transmitted, in bytes, with a value range of 1 to 512 | RW     |

### 39.5.3 Data Block Count Register (SDIOC\_BLKCNT)

| Offset | Bit Position |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0x06   | 15           | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x0000       |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | BC           |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access | RW           |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit  | Name | Function    | Description                                                                                                                                                        | Access |
|------|------|-------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 15:0 | BC   | Block Count | Set the number of transmission data blocks, set this register to be performed when the transmission is stopped, and the TRANSMODE.BCE bit is required to be active | RW     |

### 39.5.4 Argument Register 0 (SDIOC\_ARG0)

| Offset | Bit Position |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0x08   | 15           | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x0000       |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | ARG0         |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access | RW           |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit  | Name | Function         | Description                                                                                                   | Access |
|------|------|------------------|---------------------------------------------------------------------------------------------------------------|--------|
| 15:0 | ARG0 | Command argument | Set the argument contained in the current sending command, this register is the lower 16 bits of the argument | RW     |

### 39.5.5 Argument Register 1 (SDIOC\_ARG1)

| Offset | Bit Position |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0x0A   | 15           | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x0000       |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | ARG1         |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access | RW           |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit  | Name | Function         | Description                                                                                                   | Access |
|------|------|------------------|---------------------------------------------------------------------------------------------------------------|--------|
| 15:0 | ARG1 | Command argument | Set the argument contained in the current sending command, this register is the upper 16 bits of the argument | RW     |

### 39.5.6 Transmission Mode Register (SDIOC\_TRANSMODE)

| Offset | Bit Position |    |    |    |    |    |   |   |   |   |      |      |       |     |          |   |
|--------|--------------|----|----|----|----|----|---|---|---|---|------|------|-------|-----|----------|---|
| 0x0C   | 15           | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5    | 4    | 3     | 2   | 1        | 0 |
| Reset  | 0x0000       |    |    |    |    |    |   |   |   |   |      |      |       |     |          |   |
| Name   | Reserved     |    |    |    |    |    |   |   |   |   | MULB | DDIR | ATCEN | BCE | Reserved |   |
| Access |              |    |    |    |    |    |   |   |   |   | RW   | RW   | RW    | RW  |          |   |

| Bit  | Name     | Function                    | Description                                                                                                                                                                                                                 | Access |
|------|----------|-----------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 15:6 | Reserved | Reserved                    | -                                                                                                                                                                                                                           | -      |
| 5    | MULB     | Multi-block                 | <ul style="list-style-type: none"> <li>0b0: Current transmission is single block transmission (single block)</li> <li>0b1: Current transmission is multi block transmission (multi block)</li> </ul>                        | RW     |
| 4    | DDIR     | Data transmission direction | <ul style="list-style-type: none"> <li>0b0: Write operation (host sends data)</li> <li>0b1: Read operation (host receives data)</li> </ul>                                                                                  | RW     |
| 3:2  | ATCEN    | Autocommand enabled         | <ul style="list-style-type: none"> <li>0b00: Do not send auto command</li> <li>0b01: Automatically send CMD12 after multi-block transmission</li> <li>0b10: Setting prohibited</li> <li>0b11: Setting prohibited</li> </ul> | RW     |
| 1    | BCE      | Block count enabled         | <ul style="list-style-type: none"> <li>0b0: Disable data block count enable</li> <li>0b1: Enable data block count enable</li> </ul>                                                                                         | RW     |
| 0    | Reserved | Reserved                    | -                                                                                                                                                                                                                           | -      |

## 39.5.7 Command Register (SDIOC\_CMD)

| Offset | Bit Position |    |     |    |    |    |   |     |   |   |     |     |     |          |   |        |
|--------|--------------|----|-----|----|----|----|---|-----|---|---|-----|-----|-----|----------|---|--------|
| 0x0E   | 15           | 14 | 13  | 12 | 11 | 10 | 9 | 8   | 7 | 6 | 5   | 4   | 3   | 2        | 1 | 0      |
| Reset  | 0x0000       |    |     |    |    |    |   |     |   |   |     |     |     |          |   |        |
| Name   | Reserved     |    | IDX |    |    |    |   | TYP |   |   | DAT | ICE | CCE | Reserved |   | RESTYP |
| Access |              |    | RW  |    |    |    |   | RW  |   |   | RW  | RW  | RW  |          |   | RW     |

| Bit   | Name     | Function          | Description                                                                                                                                                                                                                                                                | Access |
|-------|----------|-------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 15:14 | Reserved | Reserved          | -                                                                                                                                                                                                                                                                          | -      |
| 13:8  | IDX      | Command number    | The number of the command sent                                                                                                                                                                                                                                             | RW     |
| 7:6   | TYP      | Command type      | <ul style="list-style-type: none"> <li>0b00: Normal command</li> <li>0b01: Suspend command</li> <li>0b10: Resume command</li> <li>0b11: Abort command</li> </ul>                                                                                                           | RW     |
| 5     | DAT      | Command with data | <ul style="list-style-type: none"> <li>0b0: The current command only uses the SDIOx_CMD command line</li> <li>0b1: The current command needs to use the SDIOx_Dy data line</li> </ul>                                                                                      | RW     |
| 4     | ICE      | Number check      | <ul style="list-style-type: none"> <li>0b0: Do not check the command number in the response</li> <li>0b1: Check the command number in the response</li> </ul>                                                                                                              | RW     |
| 3     | CCE      | CRC check         | <ul style="list-style-type: none"> <li>0b0: Do not check the CRC checksum in the response</li> <li>0b1: Check the CRC checksum in the response</li> </ul>                                                                                                                  | RW     |
| 2     | Reserved | Reserved          | -                                                                                                                                                                                                                                                                          | -      |
| 1:0   | RESTYP   | Response type     | <ul style="list-style-type: none"> <li>0b00: This command has no response</li> <li>0b01: This command has a response of 136 bits</li> <li>0b10: This command has a response of 48 bits</li> <li>0b11: This command has a response of 48 bits with a busy status</li> </ul> | RW     |

39.5.8 Response Register 0 (SDIOC\_RESP0)

| Offset | Bit Position |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0x10   | 15           | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x0000       |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | RESP0        |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access | RO           |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit  | Name  | Function              | Description                         | Access |
|------|-------|-----------------------|-------------------------------------|--------|
| 15:0 | RESP0 | Response mes-<br>sage | 15-0 digits of response information | RO     |

### 39.5.9 Response Register 1 (SDIOC\_RESP1)

| Offset | Bit Position |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0x12   | 15           | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x0000       |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | RESP1        |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access | RO           |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit  | Name  | Function         | Description                          | Access |
|------|-------|------------------|--------------------------------------|--------|
| 15:0 | RESP1 | Response message | 31-16 digits of response information | RO     |

### 39.5.10 Response Register 2 (SDIOC\_RESP2)

| Offset | Bit Position |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0x14   | 15           | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x0000       |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | RESP2        |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access | RO           |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit  | Name  | Function         | Description                          | Access |
|------|-------|------------------|--------------------------------------|--------|
| 15:0 | RESP2 | Response message | 47-32 digits of response information | RO     |

### 39.5.11 Response Register 3 (SDIOC\_RESP3)

| Offset | Bit Position |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0x16   | 15           | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x0000       |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | RESP3        |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access | RO           |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit  | Name  | Function         | Description                          | Access |
|------|-------|------------------|--------------------------------------|--------|
| 15:0 | RESP3 | Response message | 63-48 digits of response information | RO     |

## 39.5.12 Response Register 4 (SDIOC\_RESP4)

| Offset | Bit Position |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0x18   | 15           | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x0000       |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | RESP4        |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access | RO           |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit  | Name  | Function         | Description                          | Access |
|------|-------|------------------|--------------------------------------|--------|
| 15:0 | RESP4 | Response message | 79-64 digits of response information | RO     |

## 39.5.13 Response Register 5 (SDIOC\_RESP5)

| Offset | Bit Position |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0x1A   | 15           | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x0000       |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | RESP5        |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access | RO           |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit  | Name  | Function         | Description                          | Access |
|------|-------|------------------|--------------------------------------|--------|
| 15:0 | RESP5 | Response message | 95-80 digits of response information | RO     |

## 39.5.14 Response Register 6 (SDIOC\_RESP6)

| Offset | Bit Position |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0x1C   | 15           | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x0000       |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | RESP6        |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access | RO           |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit  | Name  | Function         | Description                           | Access |
|------|-------|------------------|---------------------------------------|--------|
| 15:0 | RESP6 | Response Message | 111-96 digits of response information | RO     |



## 39.5.15 Response Register 7 (SDIOC\_RESP7)

| Offset | Bit Position |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0x1E   | 15           | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x0000       |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | RESP7        |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access | RO           |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit  | Name  | Function         | Description                            | Access |
|------|-------|------------------|----------------------------------------|--------|
| 15:0 | RESP7 | Response Message | 127-112 digits of response information | RO     |

## 39.5.16 Data Buffer Register 0 (SDIOC\_BUF0)

| Offset | Bit Position |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0x20   | 15           | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x0000       |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | BUF0         |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access | RW           |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit  | Name | Function    | Description                                                                       | Access |
|------|------|-------------|-----------------------------------------------------------------------------------|--------|
| 15:0 | BUF0 | Data Buffer | Write send data and read receive data, this register is the lower 16 bits of data | RW     |

## 39.5.17 Data Buffer Register 1 (SDIOC\_BUF1)

| Offset | Bit Position |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0x22   | 15           | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x0000       |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | BUF1         |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access | RW           |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit  | Name | Function    | Description                                                                       | Access |
|------|------|-------------|-----------------------------------------------------------------------------------|--------|
| 15:0 | BUF1 | Data buffer | Write send data and read receive data, this register is the upper 16 bits of data | RW     |

## 39.5.18 Host Status Register (SDIOC\_PSTAT)

| Offset | Bit Position |    |    |    |    |    |    |      |      |    |    |    |     |     |     |     |          |    |    |    |    |     |     |     |     |          |   |   |   |   |   |   |    |     |     |
|--------|--------------|----|----|----|----|----|----|------|------|----|----|----|-----|-----|-----|-----|----------|----|----|----|----|-----|-----|-----|-----|----------|---|---|---|---|---|---|----|-----|-----|
| 0x24   | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24   | 23   | 22 | 21 | 20 | 19  | 18  | 17  | 16  | 15       | 14 | 13 | 12 | 11 | 10  | 9   | 8   | 7   | 6        | 5 | 4 | 3 | 2 | 1 | 0 |    |     |     |
| Reset  | 0x00000000   |    |    |    |    |    |    |      |      |    |    |    |     |     |     |     |          |    |    |    |    |     |     |     |     |          |   |   |   |   |   |   |    |     |     |
| Name   | Reserved     |    |    |    |    |    |    | CMDL | DATL |    |    |    | WPL | CDL | CSS | CIN | Reserved |    |    |    |    | BRE | BWE | RTA | WTA | Reserved |   |   |   |   |   |   | DA | CID | CIC |
| Access |              |    |    |    |    |    |    | RO   | RO   |    |    |    | RO  | RO  | RO  | RO  |          |    |    |    |    | RO  | RO  | RO  | RO  |          |   |   |   |   |   |   | RO | RO  | RO  |

| Bit   | Name     | Function                        | Description                                                                                                                                                                                                    | Access |
|-------|----------|---------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:25 | Reserved | Reserved                        | -                                                                                                                                                                                                              | -      |
| 24    | CMDL     | Command line status             | Command line (SDIOx_CMD) status                                                                                                                                                                                | RO     |
| 23:20 | DATL     | Data line status                | Command line (SDIOx_Dy (y=0-3) status                                                                                                                                                                          | RO     |
| 19    | WPL      | Write protect line status       | Command line (SDIOx_WP) status                                                                                                                                                                                 | RO     |
| 18    | CDL      | Card identification line status | Card identification line (SDIOx_CD) status                                                                                                                                                                     | RO     |
| 17    | CSS      | Device steady state             | <ul style="list-style-type: none"> <li>0b0: The state of the card identification line is unstable</li> <li>0b1: The status of the card identification line is stable, the device is inserted or not</li> </ul> | RO     |
| 16    | CIN      | Device inserted                 | <ul style="list-style-type: none"> <li>0b0: No device inserted</li> <li>0b1: There is a device inserted</li> </ul>                                                                                             | RO     |
| 15:12 | Reserved | Reserved                        | -                                                                                                                                                                                                              | -      |
| 11    | BRE      | Data buffer full                | <ul style="list-style-type: none"> <li>0b0: The data buffer does not have enough data to read</li> <li>0b1: The data buffer has enough data to read</li> </ul>                                                 | RO     |
| 10    | BWE      | Data buffer empty               | <ul style="list-style-type: none"> <li>0b0: The data buffer cannot write data</li> <li>0b1: Data buffer can write data</li> </ul>                                                                              | RO     |
| 9     | RTA      | Read operation status           | <ul style="list-style-type: none"> <li>0b0: No read operation in progress</li> <li>0b1: Read operation in progress</li> </ul>                                                                                  | RO     |
| 8     | WTA      | Write status                    | <ul style="list-style-type: none"> <li>0b0: No write operation in progress</li> <li>0b1: Write operation in progress</li> </ul>                                                                                | RO     |
| 7:3   | Reserved | Reserved                        | -                                                                                                                                                                                                              | -      |
| 2     | DA       | Data line transmission status   | <ul style="list-style-type: none"> <li>0b0: Data line idle</li> <li>0b1: Data transmission in progress</li> </ul>                                                                                              | RO     |
| 1     | CID      | With data command suppression   | <ul style="list-style-type: none"> <li>0b0: Enable sending commands with data</li> <li>0b1: Disable sending commands with data</li> </ul>                                                                      | RO     |

| Bit | Name | Function            | Description                                                                                                            | Access |
|-----|------|---------------------|------------------------------------------------------------------------------------------------------------------------|--------|
| 0   | CIC  | command suppression | <ul style="list-style-type: none"><li>● 0b0: Enable sending commands</li><li>● 0b1: Disable sending commands</li></ul> | RO     |

## 39.5.19 Host Control Register (SDIOC\_HOSTCON)

| Offset | Bit Position |      |      |          |   |      |    |          |
|--------|--------------|------|------|----------|---|------|----|----------|
| 0x28   | 7            | 6    | 5    | 4        | 3 | 2    | 1  | 0        |
| Reset  | 0x00         |      |      |          |   |      |    |          |
| Name   | CDSS         | CDTL | EXDW | Reserved |   | HSEN | DW | Reserved |
| Access | RW           | RW   | RW   |          |   | RW   | RW |          |

| Bit | Name     | Function                               | Description                                                                                                                                                                                                                                                                                                                                                                | Access |
|-----|----------|----------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 7   | CDSS     | Card identification line selection     | <ul style="list-style-type: none"> <li>0b0: Select the real SDIOx_CD line to reflect the card identification status <ul style="list-style-type: none"> <li>PSTAT.CDL=0 means the card is removed</li> <li>PSTAT.CDL=1 means the card is inserted</li> </ul> </li> <li>0b1: Select the card identification test signal to reflect the card identification status</li> </ul> | RW     |
| 6   | CDTL     | Card identification test signal status | <ul style="list-style-type: none"> <li>0b0: Card identification test signal is low level (no device is inserted)</li> <li>0b1: Card identification test signal is high level (device is inserted)</li> </ul>                                                                                                                                                               | RW     |
| 5   | EXDW     | Extended data width                    | <ul style="list-style-type: none"> <li>0b0: Data line width uses the setting of DW bit</li> <li>0b1: Data line width is 8-bit</li> </ul>                                                                                                                                                                                                                                   | RW     |
| 4:3 | Reserved | Reserved                               | -                                                                                                                                                                                                                                                                                                                                                                          | -      |
| 2   | HSEN     | High speed mode enable                 | <ul style="list-style-type: none"> <li>0b0: Disable high speed mode</li> <li>0b1: Enable high speed mode</li> </ul>                                                                                                                                                                                                                                                        | RW     |
| 1   | DW       | Data bit width selection               | <ul style="list-style-type: none"> <li>0b0: Data line width is 1-bit</li> <li>0b1: Data line width is 4-bit</li> </ul>                                                                                                                                                                                                                                                     | RW     |
| 0   | Reserved | Reserved                               | -                                                                                                                                                                                                                                                                                                                                                                          | -      |

## 39.5.20 Power Control Register (SDIOC\_PWRCON)

| Offset | Bit Position |   |   |   |   |   |   |      |
|--------|--------------|---|---|---|---|---|---|------|
| 0x29   | 7            | 6 | 5 | 4 | 3 | 2 | 1 | 0    |
| Reset  | 0x00         |   |   |   |   |   |   |      |
| Name   | Reserved     |   |   |   |   |   |   | PWON |
| Access |              |   |   |   |   |   |   | RW   |

| Bit | Name     | Function      | Description                                                                                     | Access |
|-----|----------|---------------|-------------------------------------------------------------------------------------------------|--------|
| 7:1 | Reserved | Reserved      | -                                                                                               | -      |
| 0   | PWON     | SDIOC enabled | <ul style="list-style-type: none"> <li>0b0: Disable SDIOC</li> <li>0b1: Enable SDIOC</li> </ul> | RW     |

## 39.5.21 Data Block Gap Control Register (SDIOC\_BLKGPCON)

| Offset | Bit Position |   |   |   |      |     |    |       |
|--------|--------------|---|---|---|------|-----|----|-------|
| 0x2A   | 7            | 6 | 5 | 4 | 3    | 2   | 1  | 0     |
| Reset  | 0x00         |   |   |   |      |     |    |       |
| Name   | Reserved     |   |   |   | IABG | RWC | CR | SABGR |
| Access |              |   |   |   | RW   | RW  | RW | RW    |

| Bit | Name            | Function                         | Description                                                                                                                                                                                                                   | Access |
|-----|-----------------|----------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 7:4 | <i>Reserved</i> | <i>Reserved</i>                  | -                                                                                                                                                                                                                             | -      |
| 3   | IABG            | Block Gap Interrupt Control      | <ul style="list-style-type: none"> <li>0b0: Disable receive SDIO device interrupt (card interrupt) during data block gap</li> <li>0b1: Enable receive SDIO device interrupt (card interrupt) during data block gap</li> </ul> | RW     |
| 2   | RWC             | Read Wait Control                | <ul style="list-style-type: none"> <li>0b0: Disable read wait function</li> <li>0b1: Enable read wait function</li> </ul>                                                                                                     | RW     |
| 1   | CR              | Continue Transmission            | <ul style="list-style-type: none"> <li>0b0: No effect</li> <li>0b1: Release a transfer that was stopped by setting the SABGR bit</li> </ul>                                                                                   | RW     |
| 0   | SABGR           | Data Block Gap Stop Transmission | <ul style="list-style-type: none"> <li>0b0: Stop transmission when there is no data block gap</li> <li>0b1: Stop transmission when there is a data block gap</li> </ul>                                                       | RW     |

## 39.5.22 Clock Control Register (SDIOC\_CLKCON)

| Offset | Bit Position |    |    |    |    |    |   |   |          |   |   |   |    |          |     |    |
|--------|--------------|----|----|----|----|----|---|---|----------|---|---|---|----|----------|-----|----|
| 0x2C   | 15           | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7        | 6 | 5 | 4 | 3  | 2        | 1   | 0  |
| Reset  | 0x0002       |    |    |    |    |    |   |   |          |   |   |   |    |          |     |    |
| Name   | FS           |    |    |    |    |    |   |   | Reserved |   |   |   | CE | Reserved | ICE |    |
| Access | RW           |    |    |    |    |    |   |   |          |   |   |   | RW |          |     | RW |

| Bit  | Name     | Function                | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | Access |
|------|----------|-------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 15:8 | FS       | Frequency Selection     | SDIOx_CK clock frequency division selection, the reference clock is PCLK1 <ul style="list-style-type: none"> <li>0x00: PCLK1</li> <li>0x01: PCLK1 divided by 2</li> <li>0x02: PCLK1 divided by 4</li> <li>0x04: PCLK1 divided by 8</li> <li>0x08: PCLK1 divided by 16</li> <li>0x10: PCLK1 divided by 32</li> <li>0x20: PCLK1 divided by 64</li> <li>0x40: PCLK1 divided by 128</li> <li>0x80: PCLK1 divided by 256</li> <li>0x81: PCLK1 divided by 512</li> <li>Others: Setting prohibited</li> </ul> | RW     |
| 7:3  | Reserved | Reserved                | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | -      |
| 2    | CE       | SDIOx_CK Output Control | <ul style="list-style-type: none"> <li>0b0: SDIOx_CK stop output</li> <li>0b1: SDIOx_CK output</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                              | RW     |
| 1    | Reserved | Reserved                | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | -      |
| 0    | ICE      | Clock Enabled           | <ul style="list-style-type: none"> <li>0b0: Enable SDIOC action clock</li> <li>0b1: Disable SDIOC action clock</li> </ul>                                                                                                                                                                                                                                                                                                                                                                              | RW     |

## 39.5.23 Timeout Control Register (SDIOC\_TOUTCON)

| Offset | Bit Position |   |   |   |     |   |   |   |
|--------|--------------|---|---|---|-----|---|---|---|
| 0x2E   | 7            | 6 | 5 | 4 | 3   | 2 | 1 | 0 |
| Reset  | 0x00         |   |   |   |     |   |   |   |
| Name   | Reserved     |   |   |   | DTO |   |   |   |
| Access |              |   |   |   | RW  |   |   |   |

| Bit | Name     | Function     | Description                                                                                                                                                                                                                                                                                                                                                  | Access |
|-----|----------|--------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 7:4 | Reserved | Reserved     | -                                                                                                                                                                                                                                                                                                                                                            | -      |
| 3:0 | DTO      | Data Timeout | Set the data line SDIOx_Dy (y=0-7) overtime judgment time, the unit is the clock period of PCLK1 <ul style="list-style-type: none"> <li>● 0b0000: <math>PCLK1 \times 2^{13}</math></li> <li>● 0b0001: <math>PCLK1 \times 2^{14}</math></li> <li>● .....</li> <li>● 0b1110: <math>PCLK1 \times 2^{27}</math></li> <li>● 0b1111: Setting prohibited</li> </ul> | RW     |



## 39.5.24 Software Reset Register (SDIOC\_SFTRST)

| Offset | Bit Position |   |   |   |   |      |      |      |
|--------|--------------|---|---|---|---|------|------|------|
| 0x2F   | 7            | 6 | 5 | 4 | 3 | 2    | 1    | 0    |
| Reset  | 0x00         |   |   |   |   |      |      |      |
| Name   | Reserved     |   |   |   |   | RSTD | RSTC | RSTA |
| Access |              |   |   |   |   | RW   | RW   | RW   |

| Bit | Name            | Function        | Description                                                                                                                                                                                                                                                                                                                                    | Access |
|-----|-----------------|-----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 7:3 | <i>Reserved</i> | <i>Reserved</i> | -                                                                                                                                                                                                                                                                                                                                              | -      |
| 2   | RSTD            | Data Reset      | Resets all data-related registers, including the following register bits: BUF0, BUF1, PSTAT.BRE, PSTAT.BWE, PSTAT.RTA, PSTAT.WTA, PSTAT.DLA, PSTAT.CID, BLKGPCON.CR, BLKGPCON.SABGR, NORINTST.BRR, NORINTST.BWR, NORINTST.BGE, NORINTST.TC <ul style="list-style-type: none"> <li>0b0: Normal operation</li> <li>0b1: Execute reset</li> </ul> | RW     |
| 1   | RSTC            | Command Reset   | Resets all command-related registers, including the following register bits: PSTAT.CIC, NORINTST.CC <ul style="list-style-type: none"> <li>0b0: Normal operation</li> <li>0b1: Execute reset</li> </ul>                                                                                                                                        | RW     |
| 0   | RSTA            | Reset All       | Reset all SDIOC registers except card identification function <ul style="list-style-type: none"> <li>0b0: Normal operation</li> <li>0b1: Execute reset</li> </ul>                                                                                                                                                                              | RW     |

## 39.5.25 Normal Interrupt Status Register (SDIOC\_NORINTST)

| Offset | Bit Position |          |    |    |    |    |   |      |     |      |     |     |          |     |    |    |
|--------|--------------|----------|----|----|----|----|---|------|-----|------|-----|-----|----------|-----|----|----|
| 0x30   | 15           | 14       | 13 | 12 | 11 | 10 | 9 | 8    | 7   | 6    | 5   | 4   | 3        | 2   | 1  | 0  |
| Reset  | 0x0000       |          |    |    |    |    |   |      |     |      |     |     |          |     |    |    |
| Name   | EI           | Reserved |    |    |    |    |   | CINT | CRM | CIST | BRR | BWR | Reserved | BGE | TC | CC |
| Access | RO           |          |    |    |    |    |   | RO   | RW  | RW   | RW  | RW  |          | RW  | RW | RW |

| Bit  | Name            | Function                         | Description                                                                                                           | Access |
|------|-----------------|----------------------------------|-----------------------------------------------------------------------------------------------------------------------|--------|
| 15   | EI              | Error interrupt                  | Set when any error occurs in ERRINTST Register                                                                        | RO     |
| 14:9 | <i>Reserved</i> | <i>Reserved</i>                  | -                                                                                                                     | -      |
| 8    | CINT            | Card interrupt                   | Set when the SDIO device issues a card interrupt application, and reset after the SDIO device cancels the application | RO     |
| 7    | CRM             | Card removal                     | Set when the device is removed, write 1 to reset                                                                      | RW     |
| 6    | CIST            | Card insertion                   | Set when the device is inserted, write 1 to reset                                                                     | RW     |
| 5    | BRR             | Buffer readable                  | Set when the data in the buffer can be read (PSTAT.BRE=1), write 1 to reset                                           | RW     |
| 4    | BWR             | Buffer writable                  | Set when the buffer can write data (PSTAT.BWE=1), write 1 to reset                                                    | RW     |
| 3    | <i>Reserved</i> | <i>Reserved</i>                  | -                                                                                                                     | -      |
| 2    | BGE             | Data block gap stop transmission | Set when transfer stops between data blocks, write 1 to reset                                                         | RW     |
| 1    | TC              | Transmission completion          | Set when read and write transfer is complete, write 1 to reset                                                        | RW     |
| 0    | CC              | Command completed                | Set when no response command is sent or the response of the response command is received, write 1 to reset            | RW     |

## 39.5.26 Error Interrupt Status Register (SDIOC\_ERRINTST)

| Offset | Bit Position |    |    |    |    |    |   |     |          |      |     |      |     |      |     |      |
|--------|--------------|----|----|----|----|----|---|-----|----------|------|-----|------|-----|------|-----|------|
| 0x32   | 15           | 14 | 13 | 12 | 11 | 10 | 9 | 8   | 7        | 6    | 5   | 4    | 3   | 2    | 1   | 0    |
| Reset  | 0x0000       |    |    |    |    |    |   |     |          |      |     |      |     |      |     |      |
| Name   | Reserved     |    |    |    |    |    |   | ACE | Reserved | DEBE | DCE | DTOE | CIE | CEBE | CCE | CTOE |
| Access |              |    |    |    |    |    |   | RW  |          | RW   | RW  | RW   | RW  | RW   | RW  | RW   |

| Bit  | Name     | Function                | Description                                                                                                                                   | Access |
|------|----------|-------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 15:9 | Reserved | Reserved                | -                                                                                                                                             | -      |
| 8    | ACE      | Auto send command error | Set when an error occurs in the automatic sending command (auto CMD), the type of error can be queried in ATCERRST register, write 1 to reset | RW     |
| 7    | Reserved | Reserved                | -                                                                                                                                             | -      |
| 6    | DEBE     | Data stop bit error     | Set when the data line detects a low level in the stop bit, write 1 to reset                                                                  | RW     |
| 5    | DCE      | Data CRC check error    | Set when a CRC check error occurs on the data line, write 1 to reset                                                                          | RW     |
| 4    | DTOE     | Data timeout error      | Set when data timeout occurs, write 1 to reset, data timeout time is set by TOUTCON register                                                  | RW     |
| 3    | CIE      | Wrong command number    | Set when the command number contained in the received response is wrong, write 1 to reset                                                     | RW     |
| 2    | CEBE     | Command stop bit error  | Set when the command line detects a low level in the stop bit, write 1 to reset                                                               | RW     |
| 1    | CCE      | Command CRC check error | Set when a CRC check error occurs on the command line, write 1 to reset                                                                       | RW     |
| 0    | CTOE     | Command timeout error   | Set when no response is received for more than 64 SDIOx_CLK periods after the command is sent, write 1 to reset                               | RW     |

## 39.5.27 Normal Interrupt State Enable Register (SDIOC\_NORINTSGEN)

| Offset | Bit Position |    |    |    |    |    |   |        |       |        |       |       |          |       |      |      |
|--------|--------------|----|----|----|----|----|---|--------|-------|--------|-------|-------|----------|-------|------|------|
| 0x34   | 15           | 14 | 13 | 12 | 11 | 10 | 9 | 8      | 7     | 6      | 5     | 4     | 3        | 2     | 1    | 0    |
| Reset  | 0x0000       |    |    |    |    |    |   |        |       |        |       |       |          |       |      |      |
| Name   | Reserved     |    |    |    |    |    |   | CINTEN | CRMEN | CISTEN | BRREN | BWREN | Reserved | BGEEN | TCEN | CCEN |
| Access |              |    |    |    |    |    |   | RW     | RW    | RW     | RW    | RW    |          | RW    | RW   | RW   |

| Bit  | Name     | Function                                        | Description                                                                                                                     | Access |
|------|----------|-------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------|--------|
| 15:9 | Reserved | Reserved                                        | -                                                                                                                               | -      |
| 8    | CINTEN   | Card Interrupt Status Enabled                   | <ul style="list-style-type: none"> <li>0b0: Disable NORINTST.CINT setting</li> <li>0b1: Enable NORINTST.CINT setting</li> </ul> | RW     |
| 7    | CRMEN    | Card Removed Status Enabled                     | <ul style="list-style-type: none"> <li>0b0: Disable NORINTST.CRM setting</li> <li>0b1: Enable NORINTST.CRM setting</li> </ul>   | RW     |
| 6    | CISTEN   | Card Insertion Status Enabled                   | <ul style="list-style-type: none"> <li>0b0: Disable NORINTST.CIST setting</li> <li>0b1: Enable NORINTST.CIST setting</li> </ul> | RW     |
| 5    | BRREN    | Buffer Read Status Enabled                      | <ul style="list-style-type: none"> <li>0b0: Disable NORINTST.BRR setting</li> <li>0b1: Enable NORINTST.BRR setting</li> </ul>   | RW     |
| 4    | BWREN    | Buffer Writable Status Enabled                  | <ul style="list-style-type: none"> <li>0b0: Disable NORINTST.BWR setting</li> <li>0b1: Enable NORINTST.BWR setting</li> </ul>   | RW     |
| 3    | Reserved | Reserved                                        | -                                                                                                                               | -      |
| 2    | BGEEN    | Data block gap Stop Transmission Status Enabled | <ul style="list-style-type: none"> <li>0b0: Disable NORINTST.BGE setting</li> <li>0b1: Enable NORINTST.BGE setting</li> </ul>   | RW     |
| 1    | TCEN     | Transfer Complete Status Enabled                | <ul style="list-style-type: none"> <li>0b0: Disable NORINTST.TC setting</li> <li>0b1: Enable NORINTST.TC setting</li> </ul>     | RW     |
| 0    | CCEN     | Command Completion Status Enabled               | <ul style="list-style-type: none"> <li>0b0: Disable NORINTST.CC setting</li> <li>0b1: Enable NORINTST.CC setting</li> </ul>     | RW     |

## 39.5.28 Error Interrupt Status Enable Register (SDIOC\_ERRINTSTEN)

| Offset | Bit Position |    |    |    |    |    |   |       |          |        |       |        |       |        |       |        |
|--------|--------------|----|----|----|----|----|---|-------|----------|--------|-------|--------|-------|--------|-------|--------|
| 0x36   | 15           | 14 | 13 | 12 | 11 | 10 | 9 | 8     | 7        | 6      | 5     | 4      | 3     | 2      | 1     | 0      |
| Reset  | 0x0000       |    |    |    |    |    |   |       |          |        |       |        |       |        |       |        |
| Name   | Reserved     |    |    |    |    |    |   | ACEEN | Reserved | DEBEEN | DCEEN | DTOEEN | CIEEN | CEBEEN | CCEEN | CTOEEN |
| Access |              |    |    |    |    |    |   | RW    |          | RW     | RW    | RW     | RW    | RW     | RW    | RW     |

| Bit  | Name     | Function                                        | Description                                                                                                                     | Access |
|------|----------|-------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------|--------|
| 15:9 | Reserved | Reserved                                        | -                                                                                                                               | -      |
| 8    | ACEEN    | Automatically send command error status enabled | <ul style="list-style-type: none"> <li>0b0: Disable ERRINTST.ACE setting</li> <li>0b1: Enable ERRINTST.ACE setting</li> </ul>   | RW     |
| 7    | Reserved | Reserved                                        | -                                                                                                                               | -      |
| 6    | DEBEEN   | Data stop bit error status enabled              | <ul style="list-style-type: none"> <li>0b0: Disable ERRINTST.DEBE setting</li> <li>0b1: Enable ERRINTST.DEBE setting</li> </ul> | RW     |
| 5    | DCEEN    | Data CRC check error status enabled             | <ul style="list-style-type: none"> <li>0b0: Disable ERRINTST.DCE setting</li> <li>0b1: Enable ERRINTST.DCE setting</li> </ul>   | RW     |
| 4    | DTOEEN   | Data timeout error status enabled               | <ul style="list-style-type: none"> <li>0b0: Disable ERRINTST.DTOE setting</li> <li>0b1: Enable ERRINTST.DTOE setting</li> </ul> | RW     |
| 3    | CIEEN    | Command number error status enabled             | <ul style="list-style-type: none"> <li>0b0: Disable ERRINTST.CIE setting</li> <li>0b1: Enable ERRINTST.CIE setting</li> </ul>   | RW     |
| 2    | CEBEEN   | Command stop bit error status enabled           | <ul style="list-style-type: none"> <li>0b0: Disable ERRINTST.CEBE setting</li> <li>0b1: Enable ERRINTST.CEBE setting</li> </ul> | RW     |
| 1    | CCEEN    | Command CRC check error status enabled          | <ul style="list-style-type: none"> <li>0b0: Disable ERRINTST.CCE setting</li> <li>0b1: Enable ERRINTST.CCE setting</li> </ul>   | RW     |
| 0    | CTOEEN   | Command timeout error status enabled            | <ul style="list-style-type: none"> <li>0b0: Disable ERRINTST.CTOE setting</li> <li>0b1: Enable ERRINTST.CTOE setting</li> </ul> | RW     |

## 39.5.29 Normal Interrupt Signal Enable Register (SDIOC\_NORINTSGEN)

| Offset | Bit Position |    |    |    |    |    |   |         |         |         |        |        |          |        |        |       |
|--------|--------------|----|----|----|----|----|---|---------|---------|---------|--------|--------|----------|--------|--------|-------|
| 0x38   | 15           | 14 | 13 | 12 | 11 | 10 | 9 | 8       | 7       | 6       | 5      | 4      | 3        | 2      | 1      | 0     |
| Reset  | 0x0000       |    |    |    |    |    |   |         |         |         |        |        |          |        |        |       |
| Name   | Reserved     |    |    |    |    |    |   | CINTSEN | CRMSSEN | CISTSEN | BRRSEN | BWRSEN | Reserved | BGESEN | TCSSEN | CCSEN |
| Access |              |    |    |    |    |    |   | RW      | RW      | RW      | RW     | RW     |          | RW     | RW     | RW    |

| Bit  | Name     | Function                                        | Description                                                                                                                                               | Access |
|------|----------|-------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 15:9 | Reserved | Reserved                                        | -                                                                                                                                                         | -      |
| 8    | CINTSEN  | Card interrupt signal enabled                   | <ul style="list-style-type: none"> <li>0: Disable NORINTST.CINT application interrupt</li> <li>1: Enable NORINTST.CINT application interrupt</li> </ul>   | RW     |
| 7    | CRMSSEN  | Card removal signal enabled                     | <ul style="list-style-type: none"> <li>0: Disable NORINTST.CRM application interrupt</li> <li>1: Enable NORINTST.CRM application interrupt</li> </ul>     | RW     |
| 6    | CISTSEN  | Card insertion signal enabled                   | <ul style="list-style-type: none"> <li>0: Disable NORINTST.CIST application interrupt</li> <li>1: Enable NORINTST.CIST application interrupt</li> </ul>   | RW     |
| 5    | BRRSEN   | Buffer readable signal enabled                  | <ul style="list-style-type: none"> <li>0: Disable NORINTST.BRR application interrupt</li> <li>1: Enable NORINTST.BRR application interrupt</li> </ul>     | RW     |
| 4    | BWRSEN   | Buffer writable signal enabled                  | <ul style="list-style-type: none"> <li>0: Disable NORINTST.BWR application interrupt</li> <li>1: Enable NORINTST.BWR application interrupt</li> </ul>     | RW     |
| 3    | Reserved | Reserved                                        | -                                                                                                                                                         | -      |
| 2    | BGESEN   | Data block gap stop transmission signal enabled | <ul style="list-style-type: none"> <li>0b0: Disable NORINTST.BGE application interrupt</li> <li>0b1: Enable NORINTST.BGE application interrupt</li> </ul> | RW     |
| 1    | TCSSEN   | Transfer complete signal enabled                | <ul style="list-style-type: none"> <li>0b0: Disable NORINTST.TC application interrupt</li> <li>0b1: Enable NORINTST.TC application interrupt</li> </ul>   | RW     |
| 0    | CCSEN    | Command complete signal enabled                 | <ul style="list-style-type: none"> <li>0b0: Disable NORINTST.CC application interrupt</li> <li>0b1: Enable NORINTST.CC application interrupt</li> </ul>   | RW     |

## 39.5.30 Error Interrupt Signal Enable Register (SDIOC\_ERRINTSGEN)

| Offset | Bit Position |    |    |    |    |    |   |        |          |         |        |         |        |         |         |         |
|--------|--------------|----|----|----|----|----|---|--------|----------|---------|--------|---------|--------|---------|---------|---------|
| 0x3A   | 15           | 14 | 13 | 12 | 11 | 10 | 9 | 8      | 7        | 6       | 5      | 4       | 3      | 2       | 1       | 0       |
| Reset  | 0x0000       |    |    |    |    |    |   |        |          |         |        |         |        |         |         |         |
| Name   | Reserved     |    |    |    |    |    |   | ACESEN | Reserved | DEBESEN | DCESEN | DTOESEN | CIESEN | CEBESEN | CCESSEN | CTOESEN |
| Access |              |    |    |    |    |    |   | RW     |          | RW      | RW     | RW      | RW     | RW      | RW      | RW      |

| Bit  | Name     | Function                                        | Description                                                                                                                                                 | Access |
|------|----------|-------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 15:9 | Reserved | Reserved                                        | -                                                                                                                                                           | -      |
| 8    | ACESEN   | Automatically send command error signal enabled | <ul style="list-style-type: none"> <li>0b0: Disable ERRINTST.ACE application interrupt</li> <li>0b1: Enable ERRINTST.ACE application interrupt</li> </ul>   | RW     |
| 7    | Reserved | Reserved                                        | -                                                                                                                                                           | -      |
| 6    | DEBESEN  | Data stop bit error signal enabled              | <ul style="list-style-type: none"> <li>0b0: Disable ERRINTST.DEBE application interrupt</li> <li>0b1: Enable ERRINTST.DEBE application interrupt</li> </ul> | RW     |
| 5    | DCESEN   | Data CRC check error signal enabled             | <ul style="list-style-type: none"> <li>0b0: Disable ERRINTST.DCE application interrupt</li> <li>0b1: Enable ERRINTST.DCE application interrupt</li> </ul>   | RW     |
| 4    | DTOESEN  | Data timeout error signal enabled               | <ul style="list-style-type: none"> <li>0b0: Disable ERRINTST.DTOE application interrupt</li> <li>0b1: Enable ERRINTST.DTOE application interrupt</li> </ul> | RW     |
| 3    | CIESEN   | Command number error signal enabled             | <ul style="list-style-type: none"> <li>0b0: Disable ERRINTST.CIE application interrupt</li> <li>0b1: Enable ERRINTST.CIE application interrupt</li> </ul>   | RW     |
| 2    | CEBESEN  | Command stop bit error signal enabled           | <ul style="list-style-type: none"> <li>0b0: Disable ERRINTST.CEBE application interrupt</li> <li>0b1: Enable ERRINTST.CEBE application interrupt</li> </ul> | RW     |
| 1    | CCESSEN  | Command CRC check error signal enabled          | <ul style="list-style-type: none"> <li>0b0: Disable ERRINTST.CCE application interrupt</li> <li>0b1: Enable ERRINTST.CCE application interrupt</li> </ul>   | RW     |
| 0    | CTOESEN  | Command timeout error signal enabled            | <ul style="list-style-type: none"> <li>0b0: Disable ERRINTST.CTOE application interrupt</li> <li>0b1: Enable ERRINTST.CTOE application interrupt</li> </ul> | RW     |

## 39.5.31 Auto-Command Error Status Register (SDIOC\_ATCERRST)

| Offset | Bit Position |    |    |    |    |    |   |   |      |          |   |    |     |    |     |    |
|--------|--------------|----|----|----|----|----|---|---|------|----------|---|----|-----|----|-----|----|
| 0x3C   | 15           | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7    | 6        | 5 | 4  | 3   | 2  | 1   | 0  |
| Reset  | 0x0000       |    |    |    |    |    |   |   |      |          |   |    |     |    |     |    |
| Name   | Reserved     |    |    |    |    |    |   |   | CMDE | Reserved |   | IE | EBE | CE | TOE | NE |
| Access |              |    |    |    |    |    |   |   | RO   |          |   | RO | RO  | RO | RO  | RO |

| Bit  | Name     | Function              | Description                                                                                             | Access |
|------|----------|-----------------------|---------------------------------------------------------------------------------------------------------|--------|
| 15:8 | Reserved | Reserved              | -                                                                                                       | -      |
| 7    | CMDE     | Error not sent        | Set when the error corresponding to Bit4-Bit0 of this register occurs and other commands are not sent   | RO     |
| 6:5  | Reserved | Reserved              | -                                                                                                       | -      |
| 4    | IE       | Wrong command number  | Set when the received reply contains the wrong autocommand number                                       | RO     |
| 3    | EBE      | Stop bit error        | Set when the command line detects a low level on the stop bit                                           | RO     |
| 2    | CE       | Data timeout error    | Set when a CRC check error occurs on the command line                                                   | RO     |
| 1    | TOE      | Command timeout error | Set when no response is received for more than 64 SDIOx_CLK periods after the automatic command is sent | RO     |
| 0    | NE       | Not executed error    | Set when an autocommand was not sent for other reasons                                                  | RO     |



## 39.5.32 Forced Auto-Command Error Status Control Register (SDIOC\_FEA)

| Offset | Bit Position |    |    |    |    |    |   |   |       |          |   |     |      |     |      |     |
|--------|--------------|----|----|----|----|----|---|---|-------|----------|---|-----|------|-----|------|-----|
| 0x50   | 15           | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7     | 6        | 5 | 4   | 3    | 2   | 1    | 0   |
| Reset  | 0x0000       |    |    |    |    |    |   |   |       |          |   |     |      |     |      |     |
| Name   | Reserved     |    |    |    |    |    |   |   | FCMDE | Reserved |   | FIE | FEBE | FCE | FTOE | FNE |
| Access |              |    |    |    |    |    |   |   | RW    |          |   | RW  | RW   | RW  | RW   | RW  |

| Bit  | Name     | Function                       | Description                                                                                                                                                  | Access |
|------|----------|--------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 15:8 | Reserved | Reserved                       | -                                                                                                                                                            | -      |
| 7    | FCMDE    | Force not send error           | Write:<br><ul style="list-style-type: none"> <li>0b0: No effect</li> <li>0b1: Force an ATCERRST.CMDE error</li> </ul> Reading this bit has no effect         | RW     |
| 6:5  | Reserved | Reserved                       | -                                                                                                                                                            | -      |
| 4    | FIE      | Mandatory command number error | Write:<br><ul style="list-style-type: none"> <li>0b0: No effect</li> <li>0b1: Force the ATCERRST.IE error to occur</li> </ul> Reading this bit has no effect | RW     |
| 3    | FEBE     | Forced stop bit error          | Write:<br><ul style="list-style-type: none"> <li>0b0: No effect</li> <li>0b1: Force an ATCERRST.EBE error</li> </ul> Reading this bit has no effect          | RW     |
| 2    | FCE      | Force data timeout error       | Write:<br><ul style="list-style-type: none"> <li>0b0: No effect</li> <li>0b1: Force ATCERRST.CE error</li> </ul> Reading this bit has no effect              | RW     |
| 1    | FTOE     | Force command timeout error    | Write:<br><ul style="list-style-type: none"> <li>0b0: No effect</li> <li>0b1: Force an ATCERRST.TOE error</li> </ul> Reading this bit has no effect          | RW     |
| 0    | FNE      | force not executed error       | Write:<br><ul style="list-style-type: none"> <li>0b0: No effect</li> <li>0b1: Force an ATCERRST.NE error</li> </ul> Reading this bit has no effect           | RW     |

## 39.5.33 Forced Error Status Control Register (SDIOC\_FEE)

| Offset | Bit Position |    |    |    |    |    |   |      |          |       |      |       |      |       |      |       |
|--------|--------------|----|----|----|----|----|---|------|----------|-------|------|-------|------|-------|------|-------|
| 0x52   | 15           | 14 | 13 | 12 | 11 | 10 | 9 | 8    | 7        | 6     | 5    | 4     | 3    | 2     | 1    | 0     |
| Reset  | 0x0000       |    |    |    |    |    |   |      |          |       |      |       |      |       |      |       |
| Name   | Reserved     |    |    |    |    |    |   | FACE | Reserved | FDEBE | FDCE | FDTOE | FCIE | FCEBE | FCCE | FCTOE |
| Access |              |    |    |    |    |    |   | RW   |          | RW    | RW   | RW    | RW   | RW    | RW   | RW    |

| Bit  | Name     | Function                          | Description                                                                                                                                          | Access |
|------|----------|-----------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 15:9 | Reserved | Reserved                          | -                                                                                                                                                    | -      |
| 8    | FACE     | Force auto send command error     | Write:<br><ul style="list-style-type: none"> <li>0b0: No effect</li> <li>0b1: Force an ERRINTST.ACE error</li> </ul> Reading this bit has no effect  | RW     |
| 7    | Reserved | Reserved                          | -                                                                                                                                                    | -      |
| 6    | FDEBE    | Force data stop bit error         | Write:<br><ul style="list-style-type: none"> <li>0b0: No effect</li> <li>0b1: Force an ERRINTST.DEBE error</li> </ul> Reading this bit has no effect | RW     |
| 5    | FDCE     | Mandatory data CRC check error    | Write:<br><ul style="list-style-type: none"> <li>0b0: No effect</li> <li>0b1: Force an ERRINTST.DCE error</li> </ul> Reading this bit has no effect  | RW     |
| 4    | FDTOE    | Force data timeout error          | Write:<br><ul style="list-style-type: none"> <li>0b0: No effect</li> <li>0b1: Force an ERRINTST.DTOE error</li> </ul> Reading this bit has no effect | RW     |
| 3    | FCIE     | Mandatory command number error    | Write:<br><ul style="list-style-type: none"> <li>0b0: No effect</li> <li>0b1: Force an ERRINTST.CIE error</li> </ul> Reading this bit has no effect  | RW     |
| 2    | FCEBE    | Force command stop bit error      | Write:<br><ul style="list-style-type: none"> <li>0b0: No effect</li> <li>0b1: Force an ERRINTST.CEBE error</li> </ul> Reading this bit has no effect | RW     |
| 1    | FCCE     | Mandatory command CRC check error | Write:<br><ul style="list-style-type: none"> <li>0b0: No effect</li> <li>0b1: Force an ERRINTST.CCE error</li> </ul> Reading this bit has no effect  | RW     |

| Bit | Name  | Function                    | Description                                                                                                                                        | Access |
|-----|-------|-----------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 0   | FCTOE | Force command timeout error | Write: <ul style="list-style-type: none"><li>● 0b0: No effect</li><li>● 0b1: Force an ERRINTST.CTOE error</li></ul> Reading this bit has no effect | RW     |

## 39.5.34 System Control Register (SDIOC\_SYCTLREG)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |         |          |         |          |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---------|----------|---------|----------|
| 0x04   | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3       | 2        | 1       | 0        |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |         |          |         |          |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   | SELMMC2 | Reserved | SELMMC1 | Reserved |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   | RW      |          | RW      |          |

| Bit  | Name     | Function                | Description                                                                                                         | Access |
|------|----------|-------------------------|---------------------------------------------------------------------------------------------------------------------|--------|
| 31:4 | Reserved | Reserved                | -                                                                                                                   | -      |
| 3    | SELMMC2  | SDIOC2 MMC mode enabled | <ul style="list-style-type: none"> <li>0b0: SDIOC2 selects SD mode</li> <li>0b1: SDIOC2 selects MMC mode</li> </ul> | RW     |
| 2    | Reserved | Reserved                | -                                                                                                                   | -      |
| 1    | SELMMC1  | SDIOC1 MMC mode enabled | <ul style="list-style-type: none"> <li>0b0: SDIOC1 selects SD mode</li> <li>0b1: SDIOC1 selects MMC mode</li> </ul> | RW     |
| 0    | Reserved | Reserved                | -                                                                                                                   | -      |

## 40 External Memory Controller (EXMC)

### 40.1 Introduction

The External Memory Controller (EXMC) is an independent block used to access various off-chip storage and achieve data exchange. Through configuration, EXMC can convert the internal AMBA protocol interface into various types of dedicated off-chip memory communication protocol interfaces, including SRAM, PSRAM, NOR Flash, NAND Flash, and SDRAM. EXMC is internally divided into multiple blocks, each supporting specific memory types. Users can control external corresponding memory types by configuring the registers of the block.

### 40.2 Features

Supported features are as follows:

- Supported external memory types: SRAM, PSRAM, NOR Flash, NAND Flash, SDRAM
- Interface protocol: Supports AHB protocol conversion with various external storage.
- Interface timing: Programmable interface timing for various memory types.
- Bus bit width:
  - ▶ SMC supports 8-bit, 16-bit, and 32-bit MEM bus width.
  - ▶ DMC supports 16-bit and 32-bit MEM bus width.
  - ▶ NFC supports 8-bit and 16-bit MEM bus width.
- Bus multiplexing: NOR Flash, and PSRAM support address line and data line multiplexing.
- Automatic splitting: When AHB bus bit width and external memory bit width do not match, automatic splitting and byte selection control are supported.
- Interrupt Type:
  - ▶ NFC ECC calculation completion interrupt
  - ▶ NFC ECC error interrupt
  - ▶ NFC device access end interrupt.

## 40.3 Functional Description

### 40.3.1 Controller Architecture

Various types of external memory controllers independently generate corresponding protocol interfaces and send them to the port MUX logic. The port MUX logic shares the addresses, data, and control signals of various external storage on the same port before outputting them from the chip's port. In this case, the EXMC controller can only access one external device at a time. Alternatively, partial signals of various external storage can be shared to access different external devices simultaneously.

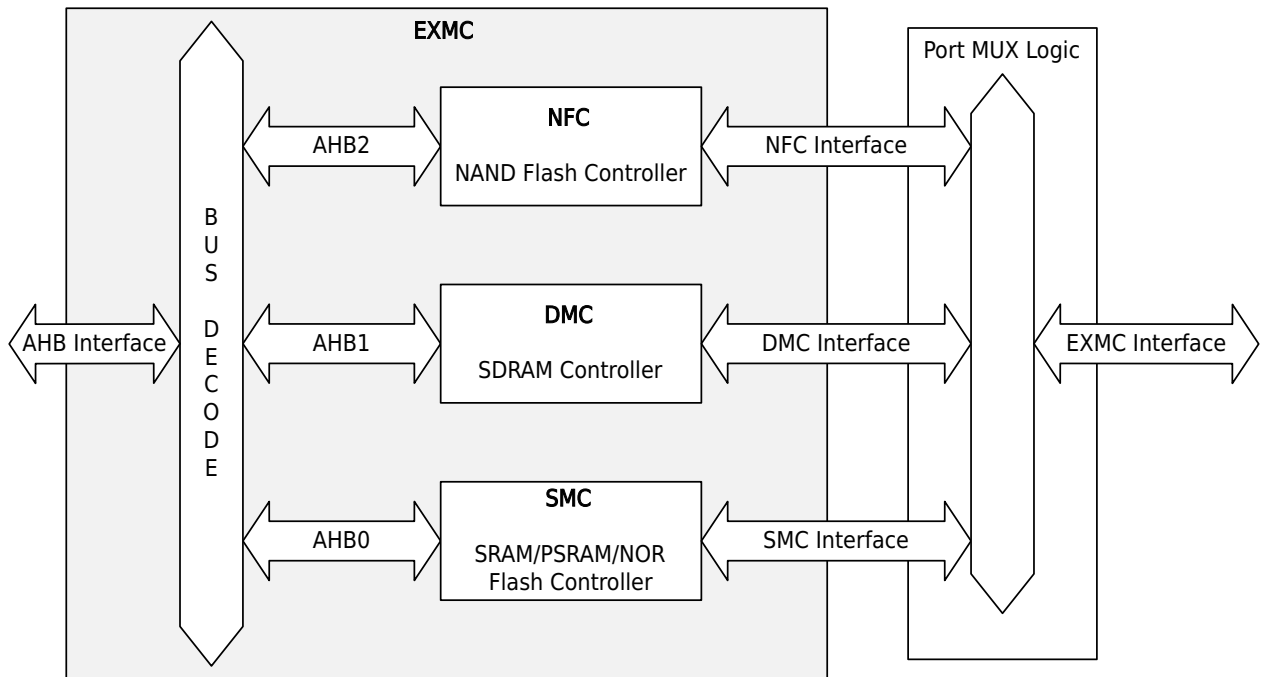


Figure 40-1 EXMC Architecture Diagram



#### Note

In the architecture diagram, the SRAM/PSRAM/NOR Flash controller is defined as SMC (Static Memory Controller), the SDRAM controller is defined as DMC (Dynamic Memory Controller), and the NAND Flash controller is defined as NFC (NAND Flash Memory Controller). Subsequent mentions of SMC, DMC, and NFC have the same meaning as their corresponding controllers.

### 40.3.2 Basic Access Specification

EXMC is the conversion interface between the AHB bus protocol and external memory protocols. During data transmission in SMC and DMC, the AHB data width and memory data width may differ. To ensure data transmission consistency, read and write accesses must comply with the following specifications:

1. If the AHB access width equals the memory width, data is accessed normally.
2. If the AHB access width is greater than the memory width, the AHB access is automatically split into several consecutive memory data width transmissions.
3. If the AHB access width is less than the memory width, the corresponding bytes are accessed through the byte selection control signals SMC\_BLS [3:0] or DMC\_DQM[3:0].

For specific access methods between AHB access width and storage of various data bit widths, refer to the table below.

Table 40-1 Table of AHB Access Width and Memory Bit Width Corresponding Access Methods

| RW | AHB Access Width | External Memory Width | Description                                  |
|----|------------------|-----------------------|----------------------------------------------|
| R  | 8                | 8                     | -                                            |
|    | 16               |                       | Converted into 2 EXMC read operations.       |
|    | 32               |                       | Converted into 4 EXMC read operations.       |
|    | 8                | 16                    | -                                            |
|    | 16               |                       | -                                            |
|    | 32               |                       | Converted into 2 EXMC read operations.       |
|    | 8                | 32                    | -                                            |
|    | 16               |                       | -                                            |
|    | 32               |                       | -                                            |
| W  | 8                | 8                     | Use byte control signal BLS[0]               |
|    | 16               |                       | Converted into 2 EXMC write operations.      |
|    | 32               |                       | Converted into 4 EXMC write operations.      |
|    | 8                | 16                    | Use Byte Control Signal BLS[1:0] or DQM[1:0] |
|    | 16               |                       | -                                            |
|    | 32               |                       | Converted into 2 EXMC write operations.      |
|    | 8                | 32                    | Use Byte Control Signal BLS[3:0] or DQM[3:0] |
|    | 16               |                       | Use Byte Control Signal BLS[3:0] or DQM[3:0] |
|    | 32               |                       | -                                            |

**Note**

The above table only applies data access SMC and DMC, where SMC supports external memory bit width of 8/16/32, and DMC supports external memory bit width of 16/32.

### 40.3.3 Address Mapping

This product defines a total of 1GB of external memory access space for data exchange between different types of external storage and internal data, including EXMC and QSPI. The 1GB space is allocated as shown in the figure below: SRAM/PSRAM/NOR Flash (SMC) data access maps to a 512MB address space, SDRAM (DMC) data access maps to a 128MB address space, NAND Flash (NFC) control and data access maps to a 1MB address space, and QSPI control and data access maps to a 128MB address space.

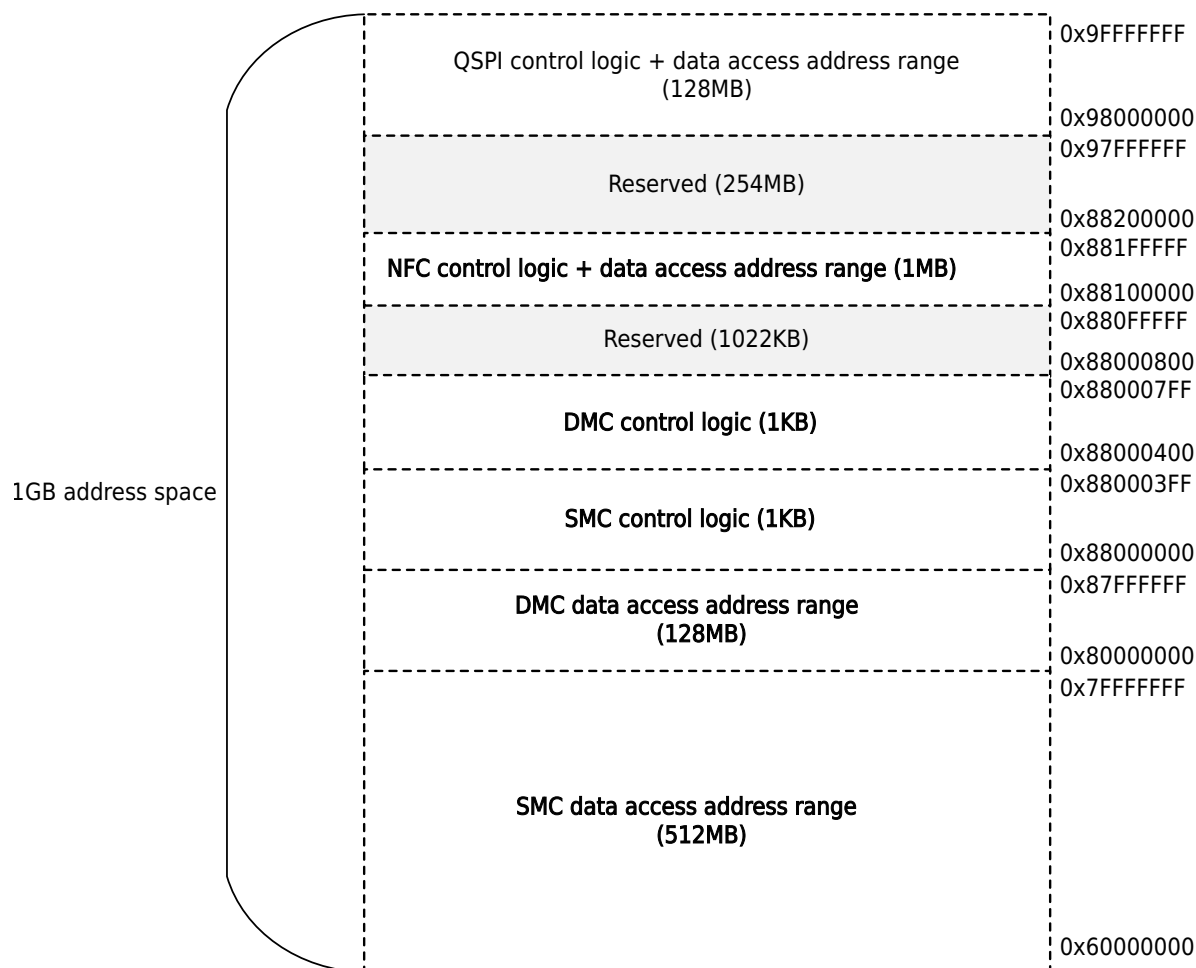


Figure 40-2 External Address Space Allocation



#### Note

1. Access control for the address space 0x98000000-0x9FFFFFFF is implemented by the QSPI block. For details, refer to the relevant description in the "Quad Serial Peripheral Interface (QSPI)" chapter.
2. The DMC data address space 0x80000000-0x87FFFFFFF does not have bufferable functionality. To improve DMC write efficiency, use the ARM Cortex-M4 built-in MPU to set this address range as bufferable.

### SMC Address Space

The SMC address space can be divided into 4 programmable-sized Chips. The space size of each Chip can be divided by setting the chip select control register (SMC\_CSCR0-1). The external memory objects corresponding to each Chip can be independently configured and correspond to different CS (Chip Select) signals output to external ports. The figure below shows the SMC Chip Select division. When the CPU accesses the address space within an SMC CS space, the corresponding SMC\_CS signal output becomes active.



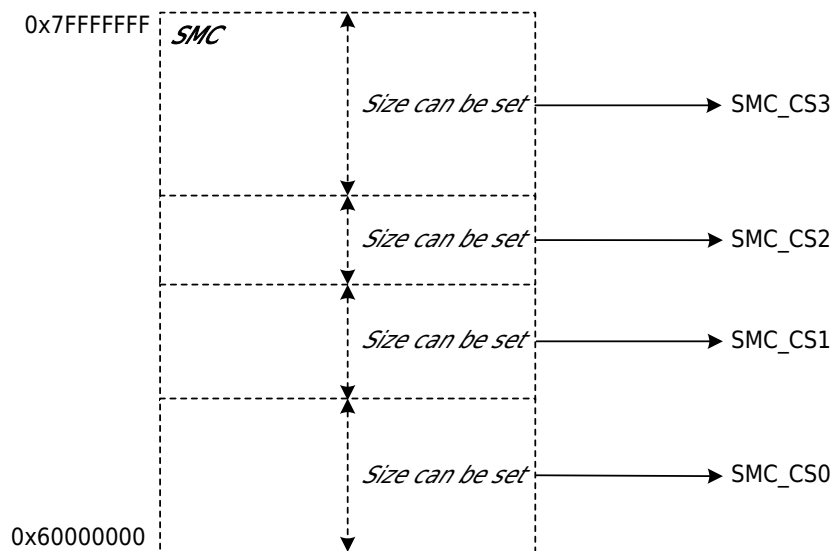


Figure 40-3 SMC Address Space Partitioning

**Note**

For the SMC address setting method, refer to "[Chip Select Control Register \(SMC\\_CSCR1\)](#)".

**DMC Address Space**

The DMC address space can be divided into 4 programmable-sized Chips. The space size of each Chip can be divided by setting the chip select control register (DMC\_CSCR0-3). The external memory objects corresponding to each Chip can be independently configured and correspond to different CS (Chip Select) signals output to external ports. The figure below shows the DMC Chip Select division. When the CPU accesses the address space within a DMC CS space, the corresponding DMC\_CS signal output becomes active.

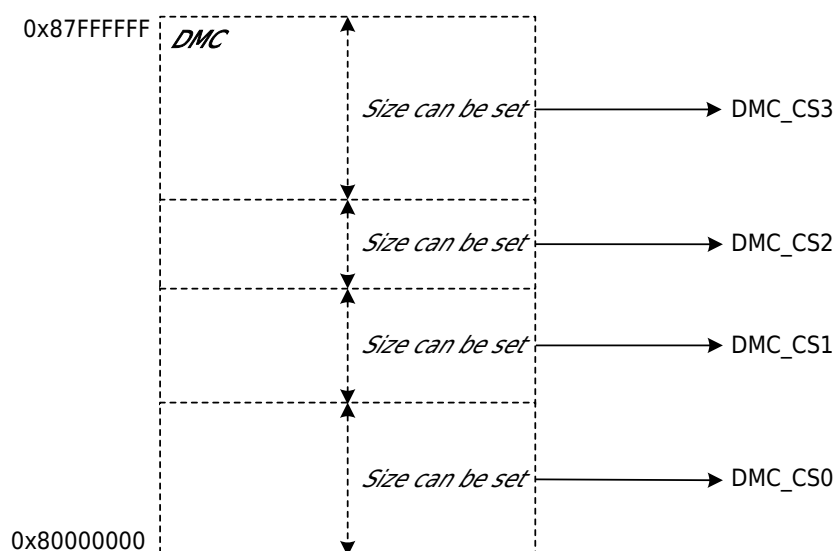


Figure 40-4 DMC Address Space Division

**Note**

For the DMC address setting method, refer to "[Chip Select Control Register \(DMC\\_CSCR0-3\)](#)".

## NFC Address Space

The NFC address space is divided into data register space and control register space. The data register occupies 32KB space, and the control register occupies 32KB space, totaling 64KB. The remaining addresses are reserved for NFC. The figure below shows the NFC address space division.

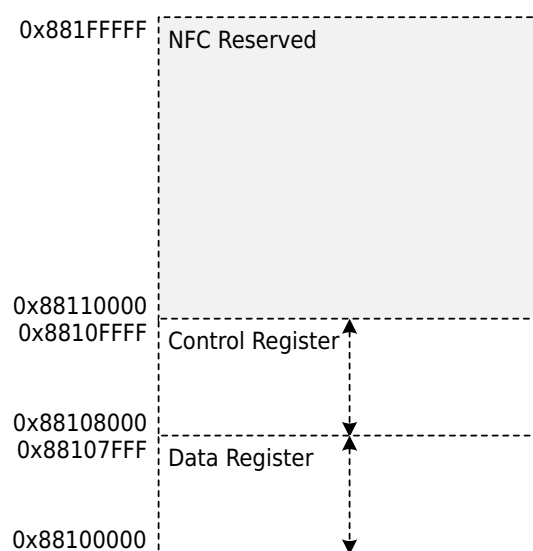


Figure 40-5 NFC Address Space Division

## 40.3.4 Protocol Interface

SMC, DMC, and NFC correspond to different types of storage, each with different types of protocol interface signals. EXMC achieves data exchange with external storage through this interface protocol. The following describes the protocol interfaces required for various types of storage.

### SMC Protocol Interface

The table below lists the protocol interfaces required for SRAM/PSRAM/NOR Flash control (SMC) access.

Table 40-2 SMC Protocol Interface

| Protocol Interface Name | Direction    | Active level.        | Description                        |
|-------------------------|--------------|----------------------|------------------------------------|
| SMC_CLK                 | Output       | High level/low level | SMC Clock Output                   |
| SMC_ADD[28:0]           | Output       | High level/low level | SMC address output                 |
| SMC_DATA[31:0]          | Input/Output | High level/low level | SMC Data                           |
| SMC_WE                  | Output       | Low level            | SMC write enable                   |
| SMC_OE                  | Output       | Low level            | SMC output enable                  |
| SMC_CS[3:0]             | Output       | Low level            | SMC chip select signal             |
| SMC_BLS[3:0]            | Output       | Low level            | SMC byte strobe signal             |
| SMC_ADV                 | Output       | Low level            | SMC address latch signal           |
| SMC_CRE                 | Output       | High level           | SMC's MdRegConfig indicates signal |

| Protocol Interface Name | Direction | Active level. | Description           |
|-------------------------|-----------|---------------|-----------------------|
| SMC_WAIT                | Input     | Low level     | SMC input wait signal |

### DMC Protocol Interface

The table below lists the protocol interfaces required for SDRAM control (DMC) access.

Table 40-3 DMC Protocol Interface

| Protocol Interface Name | Direction    | Active level. | Description                                                |
|-------------------------|--------------|---------------|------------------------------------------------------------|
| DMC_CLK                 | Output       | -             | DMC's clock output.                                        |
| DMC_ADD[13:0]           | Output       | -             | DMC's address output.                                      |
| DMC_DATA[31:0]          | Input/Output | -             | DMC's data.                                                |
| DMC_CKE                 | Output       | High level    | DMC's CLK output enable.                                   |
| DMC_AP                  | Output       | High level    | DMC's Auto Precharge command indication signal (optional). |
| DMC_WE                  | Output       | Low level     | DMC's write enable.                                        |
| DMC_RAS                 | Output       | Low level     | DMC's row address strobe signal.                           |
| DMC_CAS                 | Output       | Low level     | DMC's column address strobe signal.                        |
| DMC_BA[1:0]             | Output       | -             | DMC's Bank address signal.                                 |
| DMC_CS[3:0]             | Output       | Low level     | DMC's chip select signal.                                  |
| DMC_DQM[3:0]            | Output       | Low level     | DMC's byte strobe signal.                                  |

### NFC Protocol Interface

The table below lists the protocol interfaces required for NAND Flash control (NFC) access.

Table 40-4 NFC Protocol Interface

| Protocol Interface Name | Direction    | Active level. | Description                 |
|-------------------------|--------------|---------------|-----------------------------|
| NFC_CLE                 | Output       | High level    | NFC Command Lock Signal     |
| NFC_ALE                 | Output       | High level    | NFC Address Lock Signal     |
| NFC_DATA[15:0]          | Input/Output | -             | NFC Address and Data        |
| NFC_CE[7:0]             | Output       | Low level     | NFC's Chip Select Signal    |
| NFC_WE                  | Output       | Low level     | NFC's Write Enable          |
| NFC_RE                  | Output       | Low level     | NFC Read Enable             |
| NFC_WP                  | Output       | Low level     | NFC Write Protection Signal |
| NFC_RB[7:0]             | Input        | Low level     | NFC Input Busy Signal       |

### EXMC Port Function Configuration

Based on the interface signal tables of the above three types of external storage, the EXMC port function configuration is divided into two types: accessing one external device at a time and accessing multiple external devices simultaneously.

For port function configuration when accessing one external device at a time, refer to the table below.

**Table 40-5 EXMC Port Function Mapping 1**

| Pin Name                   | Direction        | Active Level.        | Description     |                 |                |
|----------------------------|------------------|----------------------|-----------------|-----------------|----------------|
|                            |                  |                      | SMC             | DMC             | NFC            |
| EXMC_CLK                   | Output           | High level/low level | SMC_CLK         | DMC_CLK         | -              |
| EXMC_ADD28-18              | Output           | High level/low level | SMC_ADD[28:18]  | -               | -              |
| EXMC_ADD17_DMC_BA1_NFC_ALE | Output           | High level/low level | SMC_ADD[17]     | DMC_BA[1]       | -              |
| EXMC_ADD16_DMC_BA0_NFC_CLE | Output           | High level/low level | SMC_ADD[16]     | DMC_BA[0]       | -              |
| EXMC_ADD15                 | Output           | High level/low level | SMC_ADD[15]     | -               | -              |
| EXMC_ADD14                 | Output           | High level/low level | SMC_ADD[14]     | -               | -              |
| EXMC_ADD13-0               | Output           | High level/low level | SMC_ADD[15:0]   | DMC_ADD[13:0]   | -              |
| EXMC_DATA31-0              | Input/<br>Output | High level/low level | SMC_DATA[31:16] | DMC_DATA[31:16] | -              |
|                            | Input/<br>Output | High level/low level | SMC_DATA[15:0]  | DMC_DATA[15:0]  | NFC_DATA[15:0] |
| EXMC_WE                    | Output           | Low level            | SMC_WE          | DMC_WE          | NFC_WE         |
| EXMC_CE7-4                 | Output           | Low level            | SMC_BLS[3:0]    | DMC_DQM[3:0]    | NFC_CE[7:4]    |
| EXMC_CE3-0                 | Output           | Low level            | SMC_CS[3:0]     | DMC_CS[3:0]     | NFC_CE[3:0]    |
| EXMC_OE                    | Output           | Low level            | SMC_OE          | DMC_RAS         | NFC_RE         |
| EXMC_BAA                   | Output           | Low level            | -               | DMC_CAS         | NFC_WP         |
| EXMC_ADV                   | Output           | Low level            | SMC_ADV         | -               | -              |
| EXMC_ALE                   | Output           | High level           | SMC_CRE         | DMC_CKE         | NFC_ALE        |
| EXMC_CLE                   | Output           | High level           | -               | DMC_AP          | NFC_CLE        |
| EXMC_RB7-1                 | Input            | Low level            | -               | -               | NFC_RB[7:1]    |
| EXMC_RB0                   | Input            | Low level            | SMC_WAIT        | -               | NFC_RB[0]      |

The function configuration for accessing multiple external devices simultaneously is shown in the table below. This configuration can also be used for accessing one external device at a time.



**Note**

1. NFC does not support simultaneous use with either or both SMC/DMC.
2. Direct data transfer between DMC and SMC via the CPU is disabled. If data transfer between SMC and DMC is required, use DMA transfer or relay transfer through system RAM.

**Table 40-6 EXMC Port Function Mapping 2**

| Pin Name           | Direction | Active Level.        | Description    |           |         |
|--------------------|-----------|----------------------|----------------|-----------|---------|
|                    |           |                      | SMC            | DMC       | NFC     |
| EXMC_DMC_CLK       | Output    | High level/low level | -              | DMC_CLK   | -       |
| EXMC_SMC_CLK       | Output    | High level/low level | SMC_CLK        | -         | -       |
| EXMC_ADD28-18      | Output    | High level/low level | SMC_ADD[28:18] | -         | -       |
| EXMC_ADD17_NFC_ALE | Output    | High level/low level | SMC_ADD[17]    | -         | NFC_ALE |
| EXMC_ADD16_NFC_CLE | Output    | High level/low level | SMC_ADD[16]    | -         | NFC_CLE |
| EXMC_ADD15_DMC_BA1 | Output    | High level/low level | SMC_ADD[15]    | DMC_BA[1] | -       |
| EXMC_ADD14_DMC_BA0 | Output    | High level/low level | SMC_ADD[14]    | DMC_BA[0] | -       |

| Pin Name            | Direction        | Active Level.        | Description     |                 |                |
|---------------------|------------------|----------------------|-----------------|-----------------|----------------|
|                     |                  |                      | SMC             | DMC             | NFC            |
| EXMC_ADD13-0        | Output           | High level/low level | SMC_ADD[13:0]   | DMC_ADD[13:0]   | -              |
| EXMC_DATA31-0       | Input/<br>Output | High level/low level | SMC_DATA[31:16] | DMC_DATA[31:16] | -              |
|                     | Input/<br>Output | High level/low level | SMC_DATA[15:0]  | DMC_DATA[15:0]  | NFC_DATA[15:0] |
| EXMC_DMC_WE         | Output           | Low level            | -               | DMC_WE          | -              |
| EXMC_SMC_NFC_WE     | Output           | Low level            | SMC_WE          | -               | NFC_WE         |
| EXMC_SMC_OE_NFC_RE  | Output           | Low level            | SMC_OE          | -               | NFC_RE         |
| EXMC_SMC_DMC_BLS3-0 | Output           | Low level            | SMC_BLS[3:0]    | DMC_DQM[3:0]    | -              |
| EXMC_SMC_NFC_CS3-0  | Output           | Low level            | SMC_CS[3:0]     | -               | NFC_CE[3:0]    |
| EXMC_DMC_CS3-0      | Output           | Low level            | -               | DMC_CS[3:0]     | -              |
| EXMC_NFC_CE7-4      | Output           | Low level            | -               | -               | NFC_CE[7:4]    |
| EXMC_DMC_RAS        | Output           | Low level            | -               | DMC_RAS         | -              |
| EXMC_SMC_BAA_NFC_WP | Output           | Low level            | -               | -               | NFC_WP         |
| EXMC_DMC_CAS        | Output           | Low level            | -               | DMC_CAS         | -              |
| EXMC_SMC_ADV        | Output           | Low level            | SMC_ADV         | -               | -              |
| EXMC_DMC_CKE        | Output           | High level           | -               | DMC_CKE         | -              |
| EXMC_SMC_CRE        | Output           | High level           | SMC_CRE         | -               | -              |
| EXMC_DMC_AP         | Output           | High level           | -               | DMC_AP          | -              |
| EXMC_RB7-1          | Input            | Low level            | -               | -               | NFC_RB[7:1]    |
| EXMC_RB0            | Input            | Low level            | SMC_WAIT        | -               | NFC_RB[0]      |

### 40.3.5 SMC-SRAM/PSRAM/NOR Flash Controller

#### 40.3.5.1 SRAM/PSRAM/NOR Flash Introduction

SRAM/NOR Flash are storage with static access functionality that can retain internally stored data without a refresh circuit. The width of the address lines determines the number of storage units. Each storage unit corresponds to a space of the memory storage width for storing data. Therefore, the capacity of SRAM/PSRAM/NOR Flash storage is memory data width x  $2^{\text{address line width}}$ .

PSRAM is pseudo-static RAM. Because it has an SRAM interface, it can exchange data with the system internally through the SMC block of EXMC, just like SRAM.

#### 40.3.5.2 SMC Basic Functions

The basic functions and features of SMC are as follows:

- Supports 8-bit, 16-bit and 32-bit external memory data bandwidth.
- AHB word, half-word, and byte access.
- Provides independent chip select control for each memory Chip.
- Byte selection signal output.
- Address line and data line multiplexing.
- Programmable protocol timing parameters.
- Programmable-rate automatic refresh action (used for PSRAM).
- 2 command FIFOs.

- Equipped with four 32-bit write data FIFOs
- Equipped with four 32-bit read data FIFOs.
- Low power management.
- Supports LCD parallel interface, compatible with 8080/6800 modes.

#### 40.3.5.3 SMC Initial Setting

The state diagram and transitions between states of SMC after power on reset are shown below.

During initial reset, SMC is in the Low Power state. By setting the state control register (SMC\_STCR0 and SMC\_STCR1), SMC can switch between the Ready state and the Low Power state. For details, refer to "[SMC Low Power Management](#)".

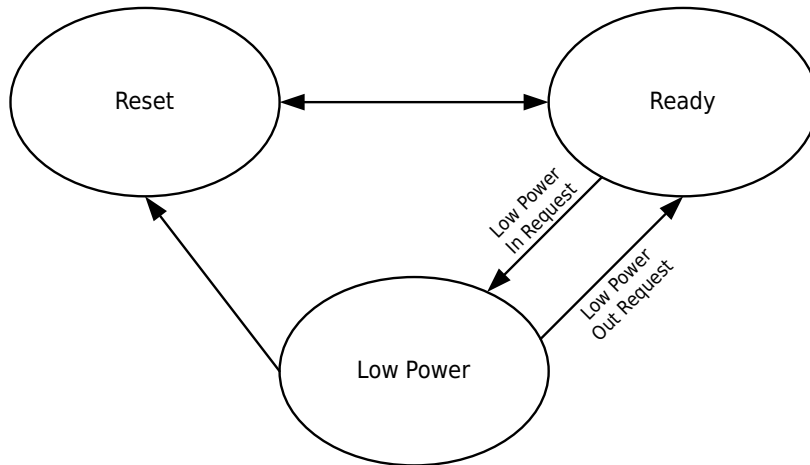


Figure 40-6 SMC State Diagram

Before communicating with external SRAM, PSRAM, NOR Flash, etc., SMC must be initially set up and the relevant parameters configured to ensure correct data transmission. For the specific setup sequence, refer to the figure below.

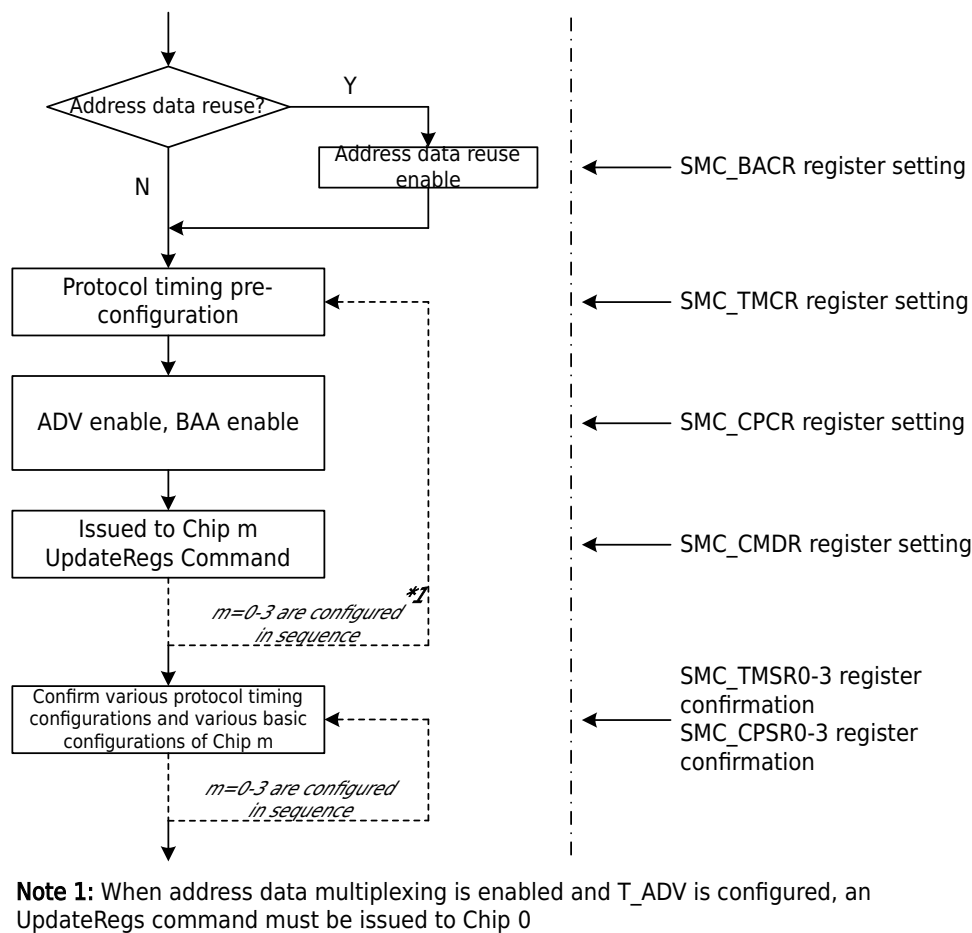


Figure 40-7 SMC Initial Setup Process

#### 40.3.5.4 SMC Access Actions

##### 40.3.5.4.1 Access Method

For SMC's read and write operations on external storage, either synchronous or asynchronous access modes can be selected.

Synchronous access mode corresponds to synchronous-access-type external storage. When SMC outputs read and write control signals, the clock signal is also output simultaneously. The external memory performs data transmission with SMC based on the clock output by SMC, i.e., SMC\_CLK is active. Asynchronous access mode corresponds to asynchronous-access-type external storage. When SMC outputs read and write control signals, the clock signal is not output (output remains low). The external memory achieves data access based on other control signals.

In synchronous access mode, the read and write action between SMC and external memory is controlled by the SMC\_WAIT signal. When the SMC\_WAIT signal input is at a low level, SMC is in a wait state and does not perform any operations. SMC resumes normal read and write access only after SMC\_WAIT becomes high level.

By setting the SMC\_CPCR.RSYN and SMC\_CPCR.WSYN bits, synchronous or asynchronous read and write modes can be selected.



#### Note

If no pin is configured as the EXMC\_RB0 function (see "Pin Definitions - Pin Function Description" in the "Datasheet"), the read and write action of SMC is not controlled by the SMC\_WAIT signal.

#### 40.3.5.4.2 Access Timing

SMC supports access to multiple types of storage and can convert AHB single or burst read and write operations into memory read and write operations. Below are examples of timing sequences for data read and write operations between the controller and external memory:

1. [Single Read Action](#)
2. [Single Read Action with Address and Data Line Multiplexing](#)
3. [Single Write Operation](#)
4. [Single Write Action with Address and Data Line Multiplexing in Asynchronous Mode](#)
5. [Single Write Action with Address and Data Line Multiplexing in Synchronous Mode](#)

#### Single Read Action

The basic timing diagram and setup example for a single read action are as follows:

**Table 40-7 Basic Setup Example for a Single Read Action**

|                  |        |       |        |      |       |       |                                                                   |
|------------------|--------|-------|--------|------|-------|-------|-------------------------------------------------------------------|
| Basic set-tings  | MW     | RSYN  | WSYN   | -    | ADVS  | BLSS  | - indicates not appli-cable.<br><set> indi-cates user-set values. |
|                  | <set>  | <set> | -      | -    | 0b0   | 0b0   |                                                                   |
| Timing set-tings | T_RC   | T_WC  | T_CEOE | T_WP | T_TR  | T_ADV |                                                                   |
|                  | 0b0011 | -     | 0b001  | -    | 0b001 | -     |                                                                   |



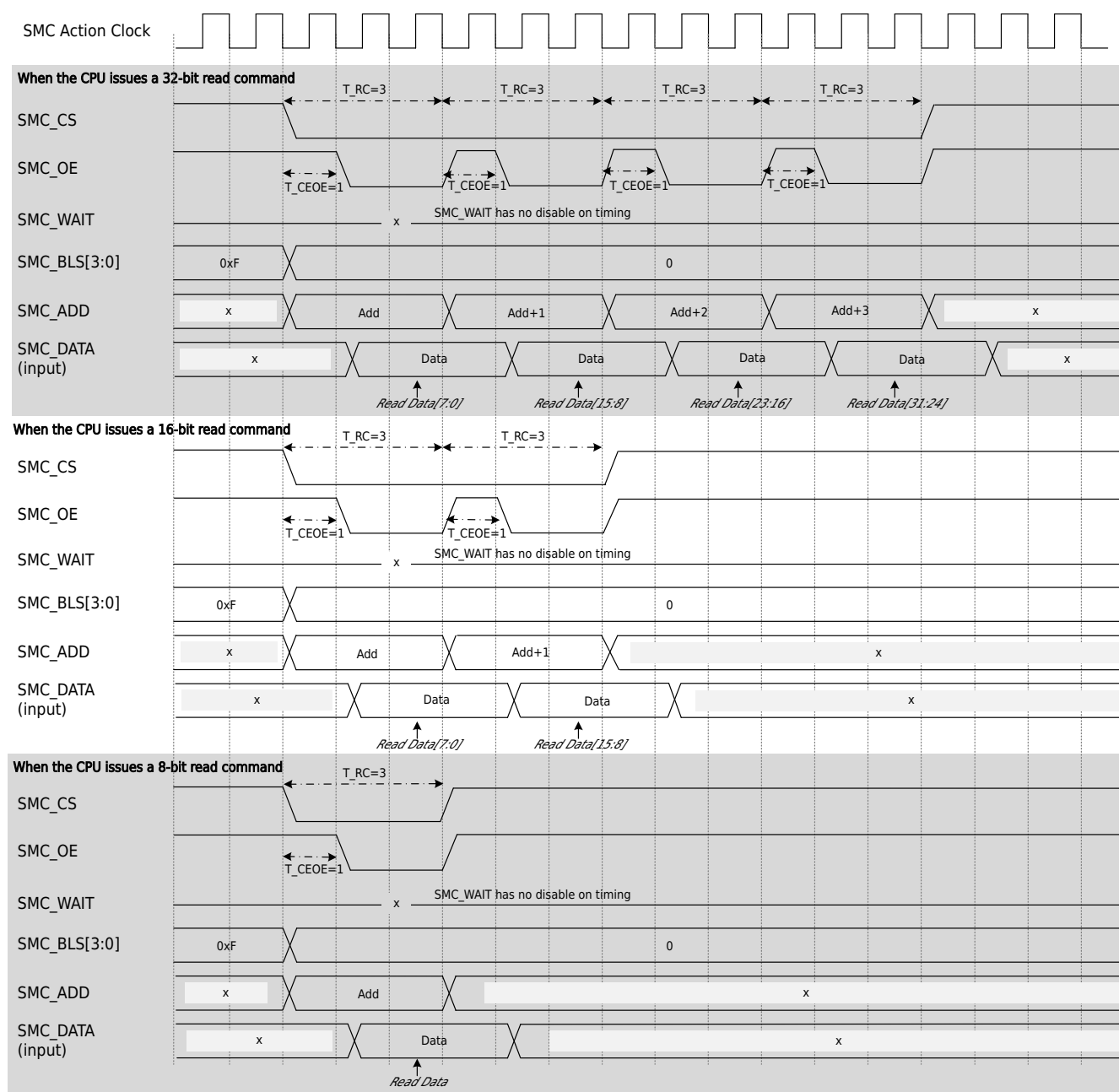


Figure 40-8 Single Read Operation Basic Timing (Asynchronous Mode RSYN=0b0 &amp; 8-bit Data Width MW=0b00)

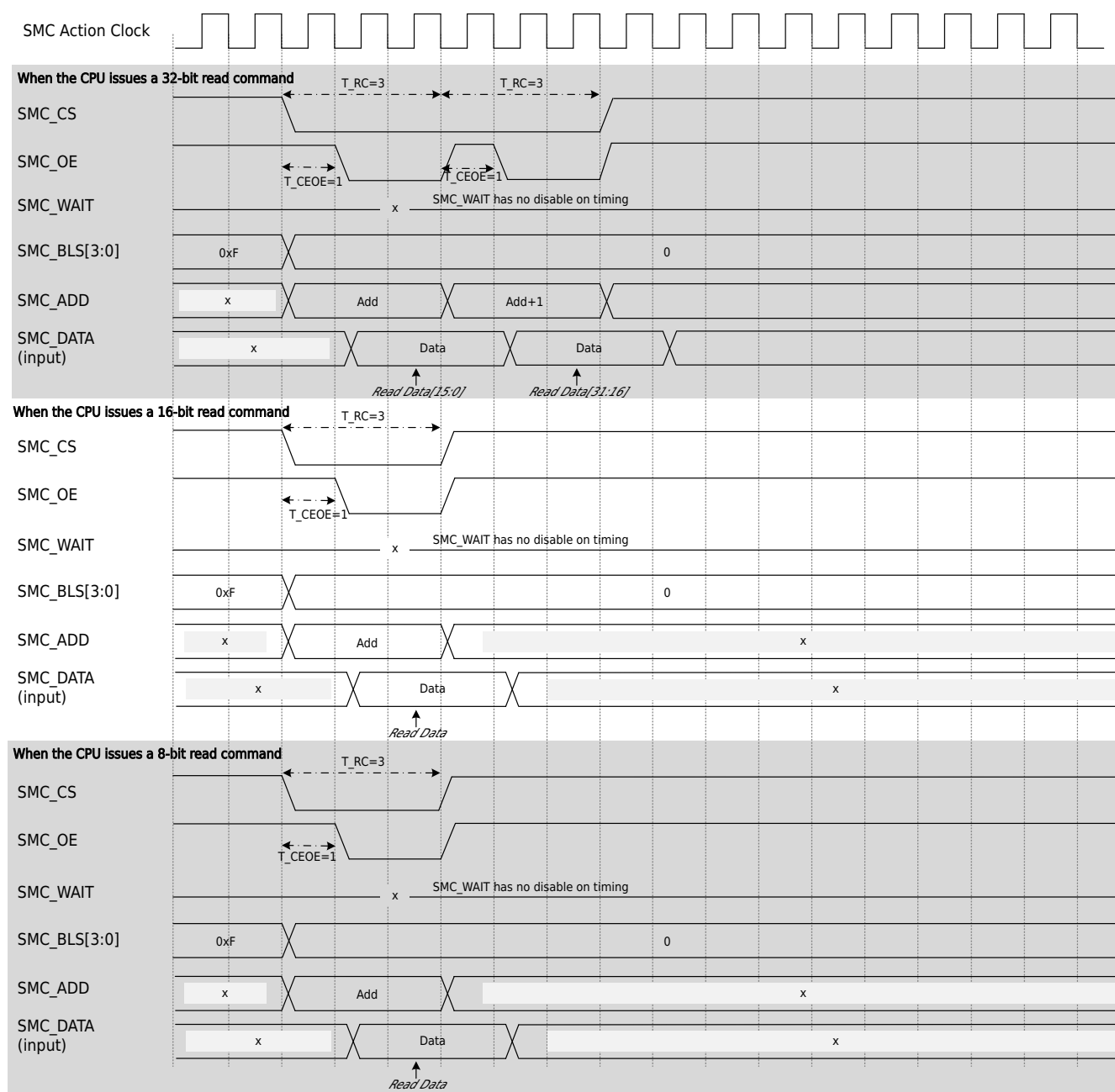


Figure 40-9 Single Read Operation Basic Timing (Asynchronous Mode RSYN=0b0 &amp; 16-bit Data Width MW=0b01)

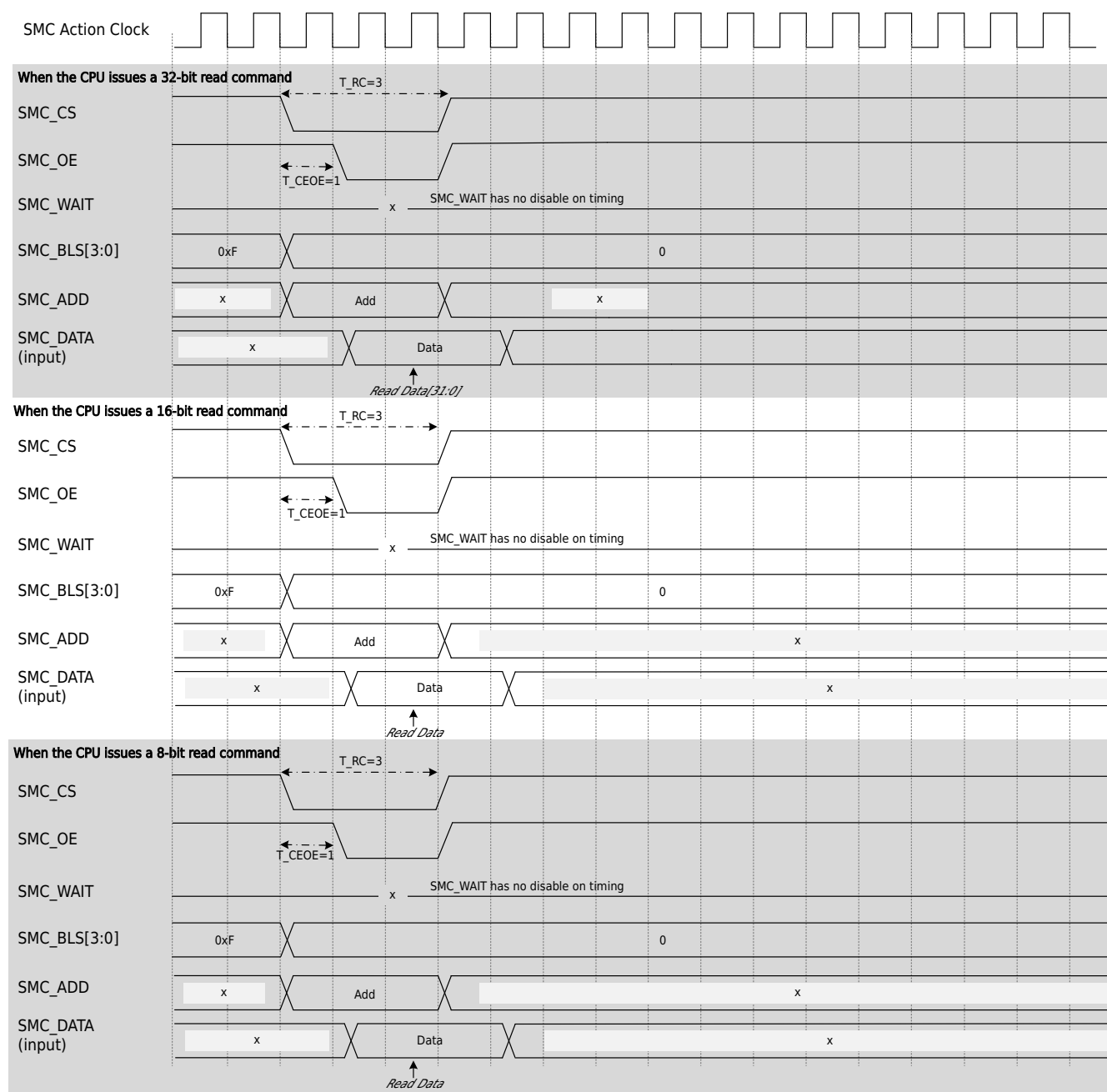


Figure 40-10 Single Read Operation Basic Timing (Asynchronous Mode RSYN=0b0 &amp; 32-bit Data Width MW=0b10)

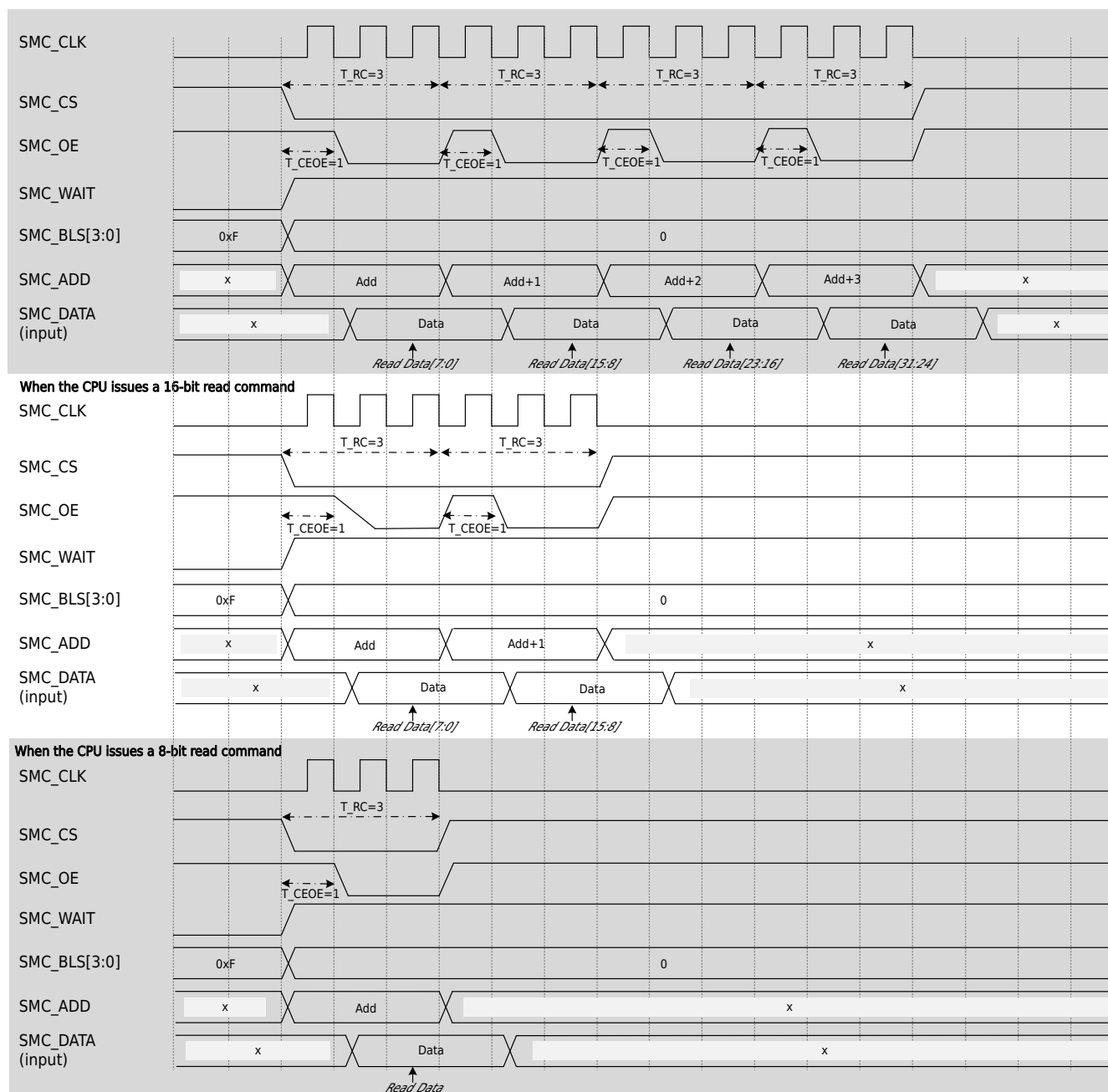


Figure 40-11 Basic Timing for Single Read Action (Synchronous Mode (RSYN=0b1) &amp; 8-bit Width (MW=0b00) )

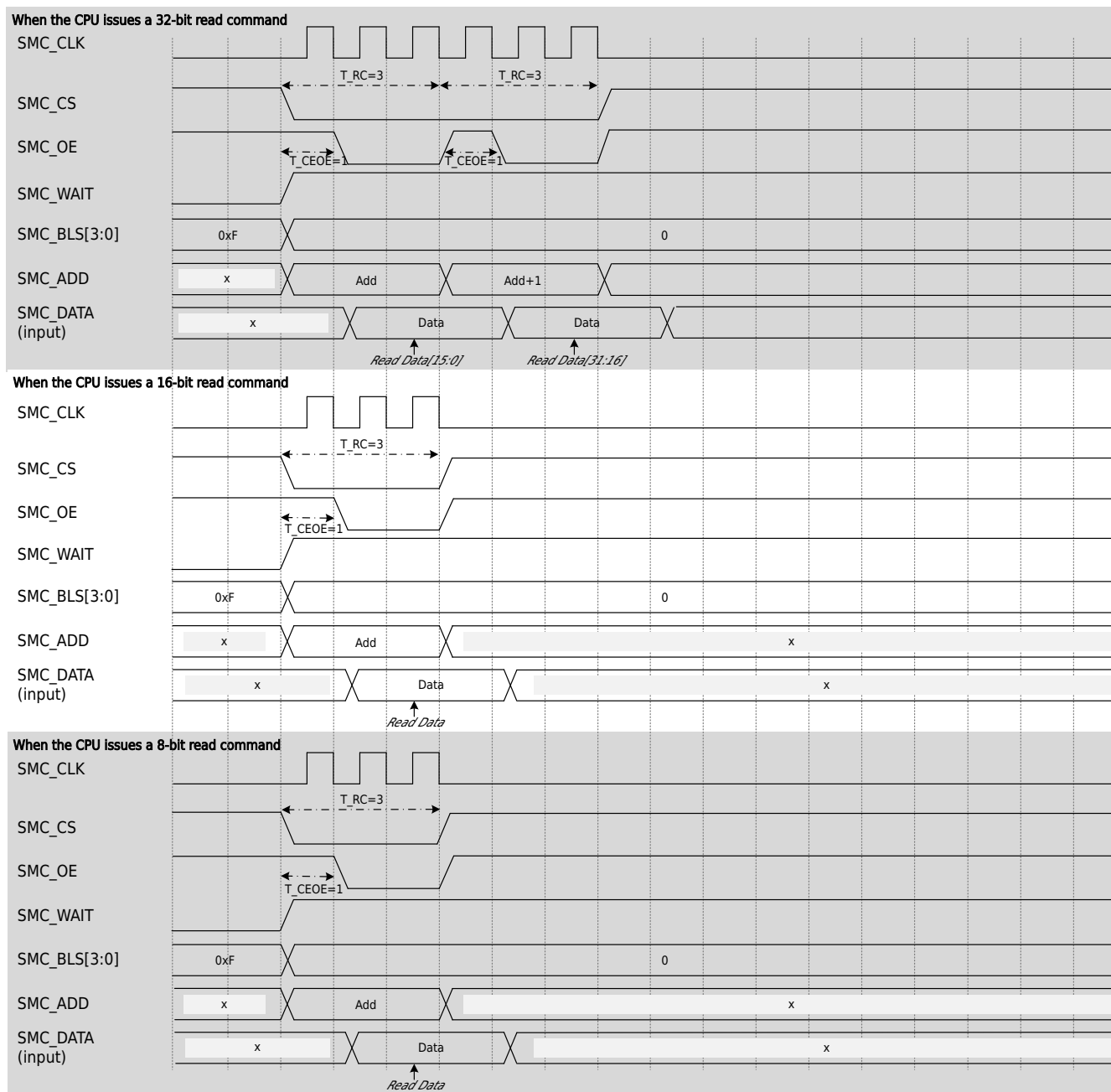


Figure 40-12 Basic Timing for Single Read Action (Synchronous Mode (RSYN=0b1) &amp; 16-bit Width (MW=0b01) )

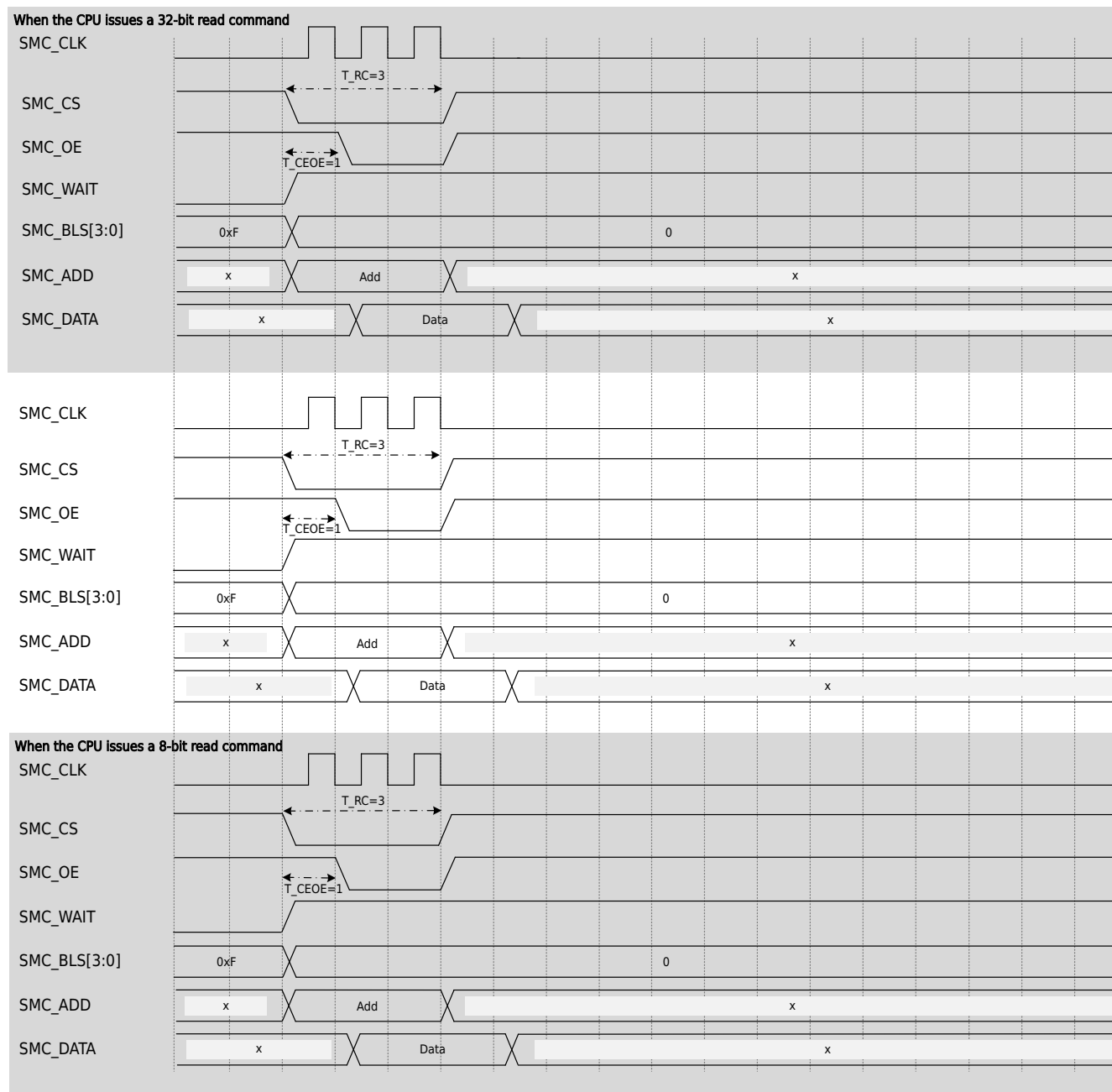


Figure 40-13 Basic Timing for Single Read Action (Synchronous Mode (RSYN=0b1) &amp; 32-bit Width (MW=0b10) )

## Single Read Action with Address and Data Line Multiplexing

The basic timing diagram and setup example for a single read action with address and data line multiplexing are as follows:

**Table 40-8 Basic Setup Example for a Single Read Action with Address and Data Line Multiplexing**

|                 |        |       |        |      |       |       |                                                                 |
|-----------------|--------|-------|--------|------|-------|-------|-----------------------------------------------------------------|
| Basic settings  | MW     | RSYN  | WSYN   | -    | ADVS  | BLSS  | - indicates not applicable.<br><set> indicates user-set values. |
|                 | <set>  | <set> | -      | -    | 0b1   | 0b0   |                                                                 |
| Timing settings | T_RC   | T_WC  | T_CEOE | T_WP | T_TR  | T_ADV |                                                                 |
|                 | 0b0110 | -     | 0b011  | -    | 0b010 | 0b001 |                                                                 |

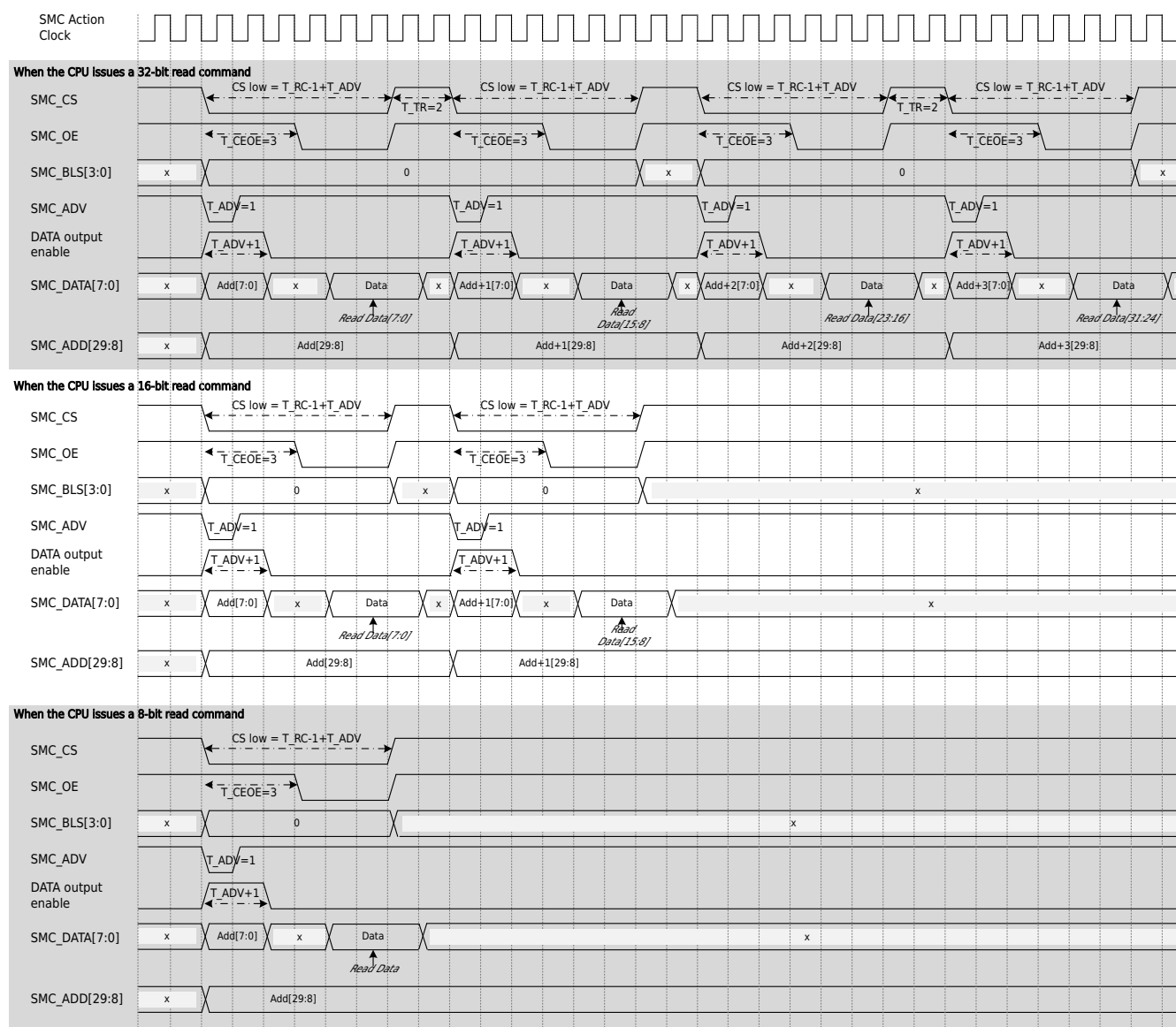


**Figure 40-14 Basic Timing for Single Read Action with Address and Data Line Multiplexing (8-bit Width (MW=0b00) )**

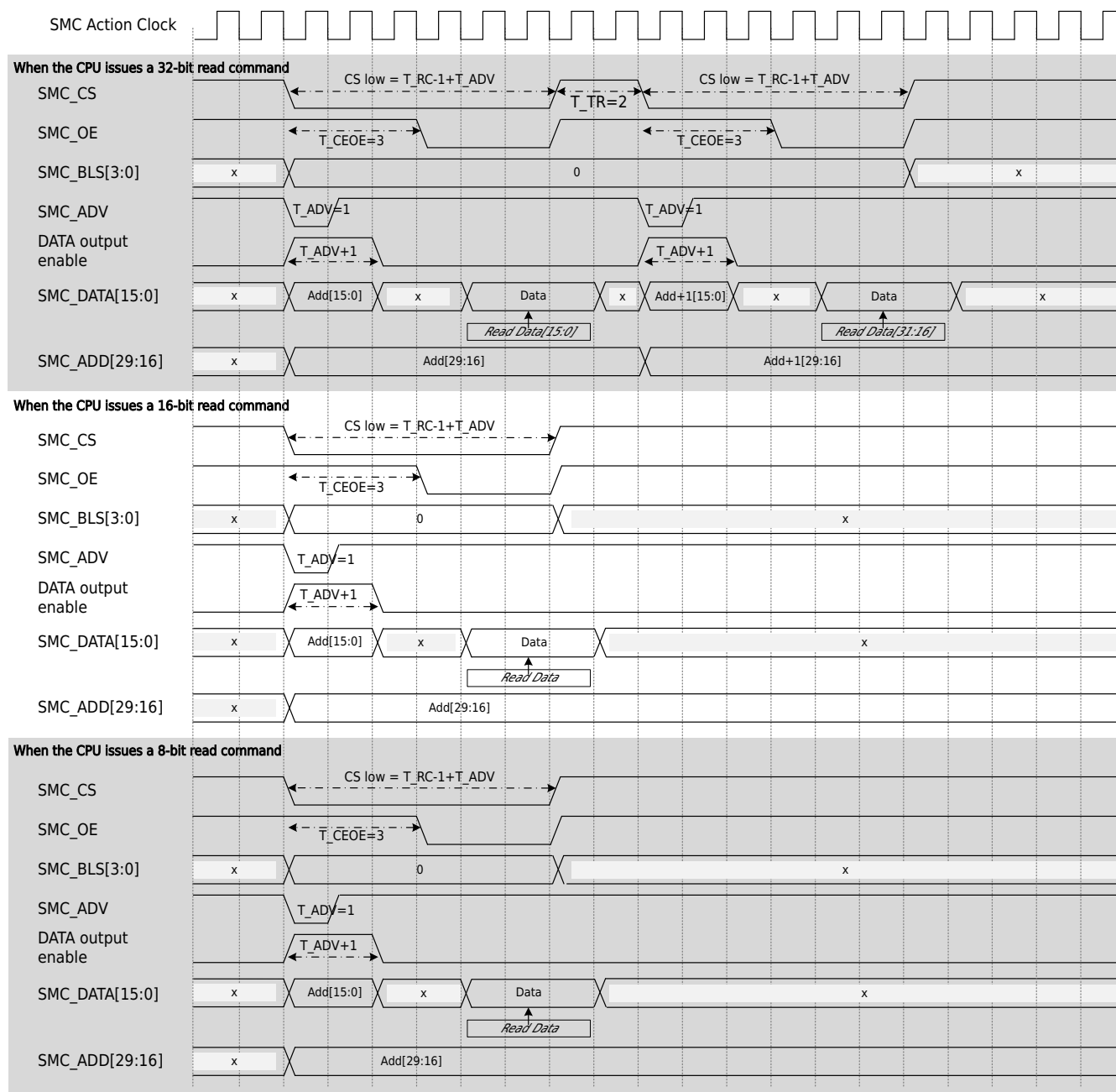


Figure 40-15 Basic Timing for Single Read Action with Address and Data Line Multiplexing (16-bit Width (MW=0b01) )



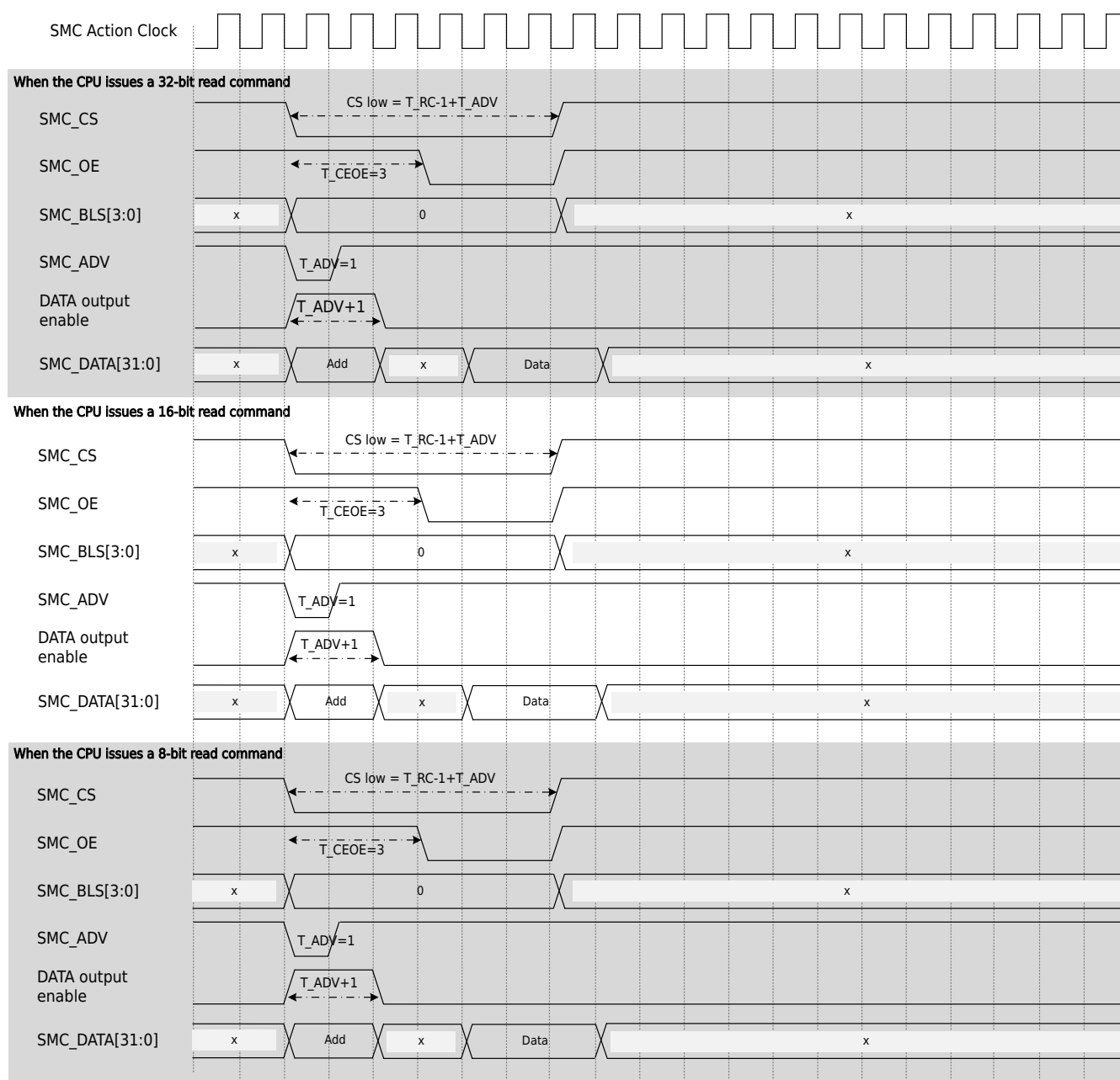


Figure 40-16 Basic Timing for Single Read Action with Address and Data Line Multiplexing (32-bit Width (MW=0b10) )

## Single Write Operation

The basic timing diagram and setup example for a single write action are as follows:

Table 40-9 Basic Setup Example for a Single Write Action

|                  |       |        |        |       |       |       |                                                                               |
|------------------|-------|--------|--------|-------|-------|-------|-------------------------------------------------------------------------------|
| Basic set-tings  | MW    | RSYN   | WSYN   | -     | ADVS  | BLSS  | - indicates not appli-<br>cable.<br><set> indi-<br>cates user-<br>set values. |
|                  | <set> | -      | <set>  | -     | 0b0   | <set> |                                                                               |
| Timing set-tings | T_RC  | T_WC   | T_CEOE | T_WP  | T_TR  | T_ADV |                                                                               |
|                  | -     | 0b0100 | -      | 0b010 | 0b001 | -     |                                                                               |

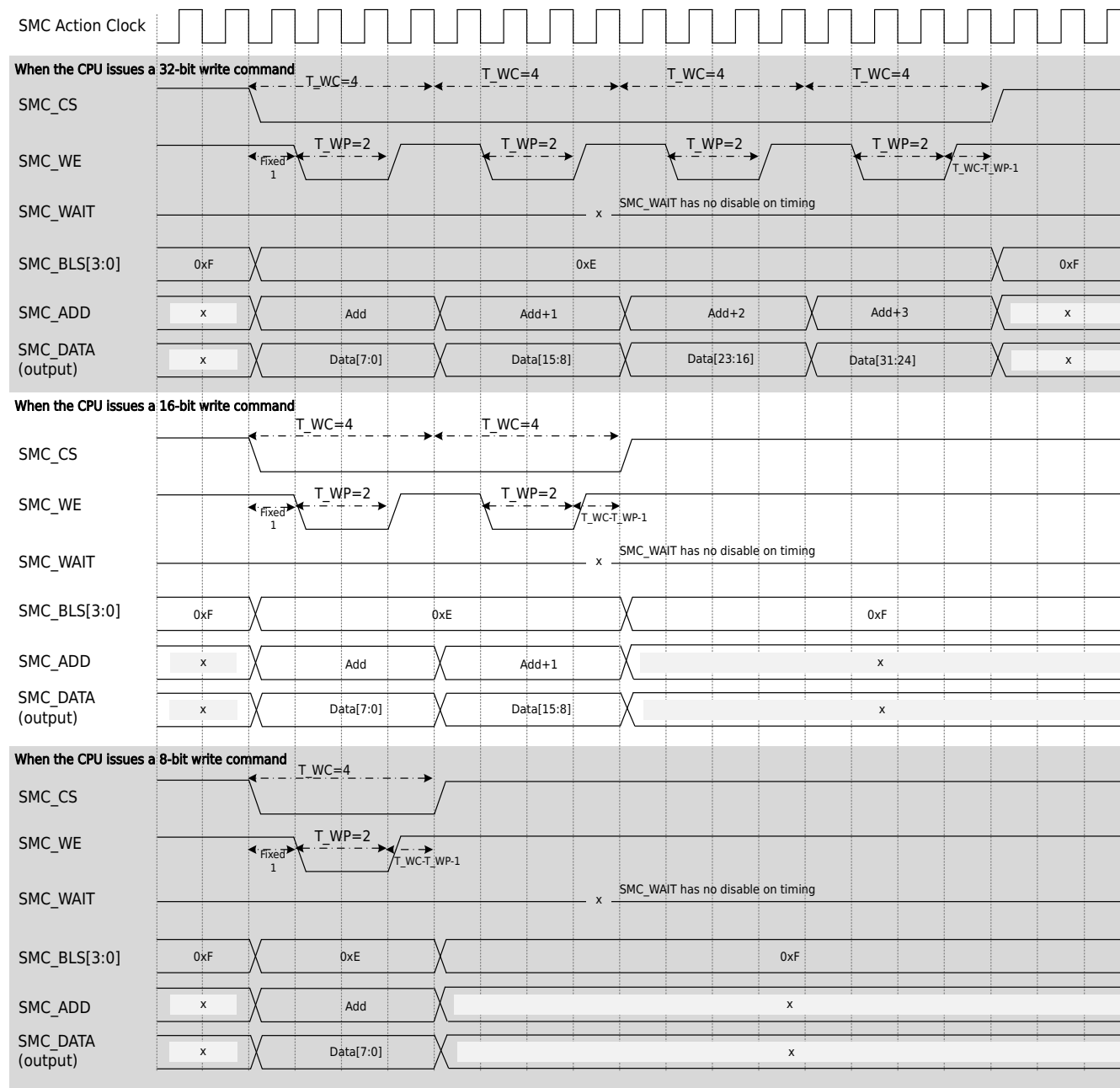


Figure 40-17 Single Write Operation Basic Timing (Asynchronous Mode WSYN=0b0, 8-bit Data Width MW=0b00, BLSS=0b0)

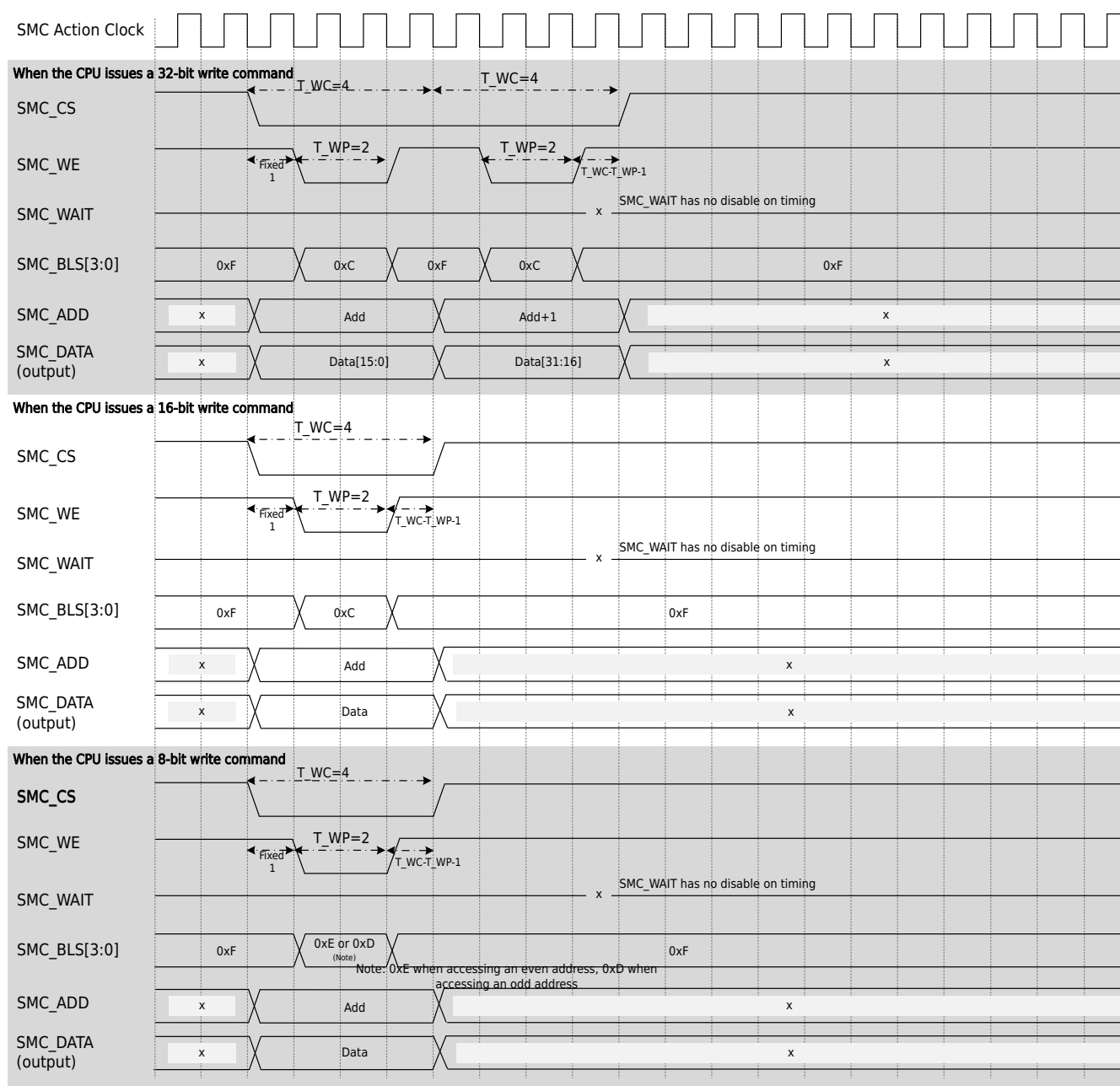


Figure 40-18 Single Write Operation Basic Timing (Asynchronous Mode WSYN=0b0, 16-bit Data Width MW=0b01, BLSS=0b1)

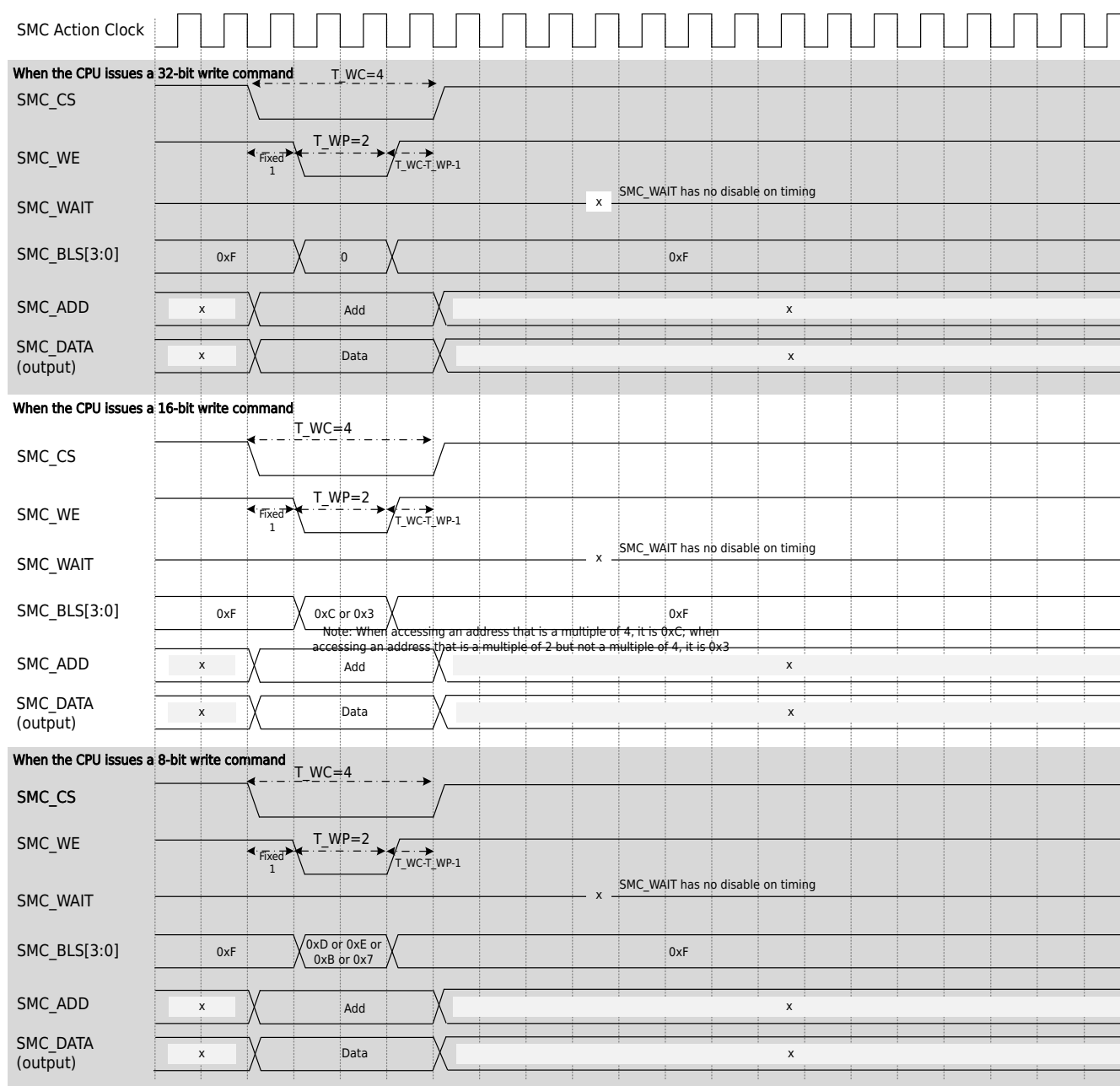


Figure 40-19 Single Write Operation Basic Timing (Asynchronous Mode WSYN=0b0, 32-bit Data Width MW=0b10, BLSS=0b1)

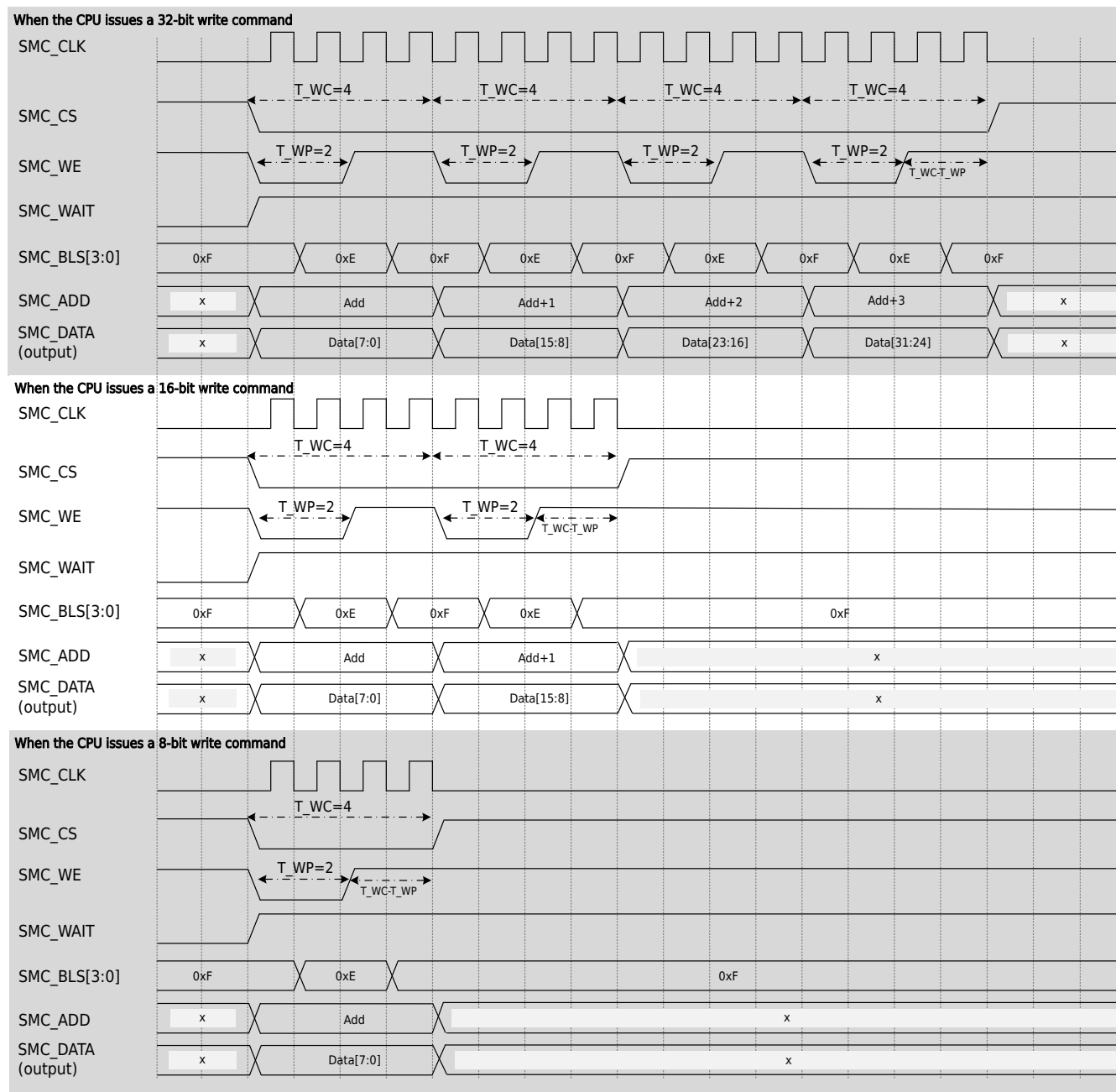


Figure 40-20 Basic Timing for Single Write Action (WSYN=0b1) & 8-bit Width (MW=0b00) & BLSS=0b1)

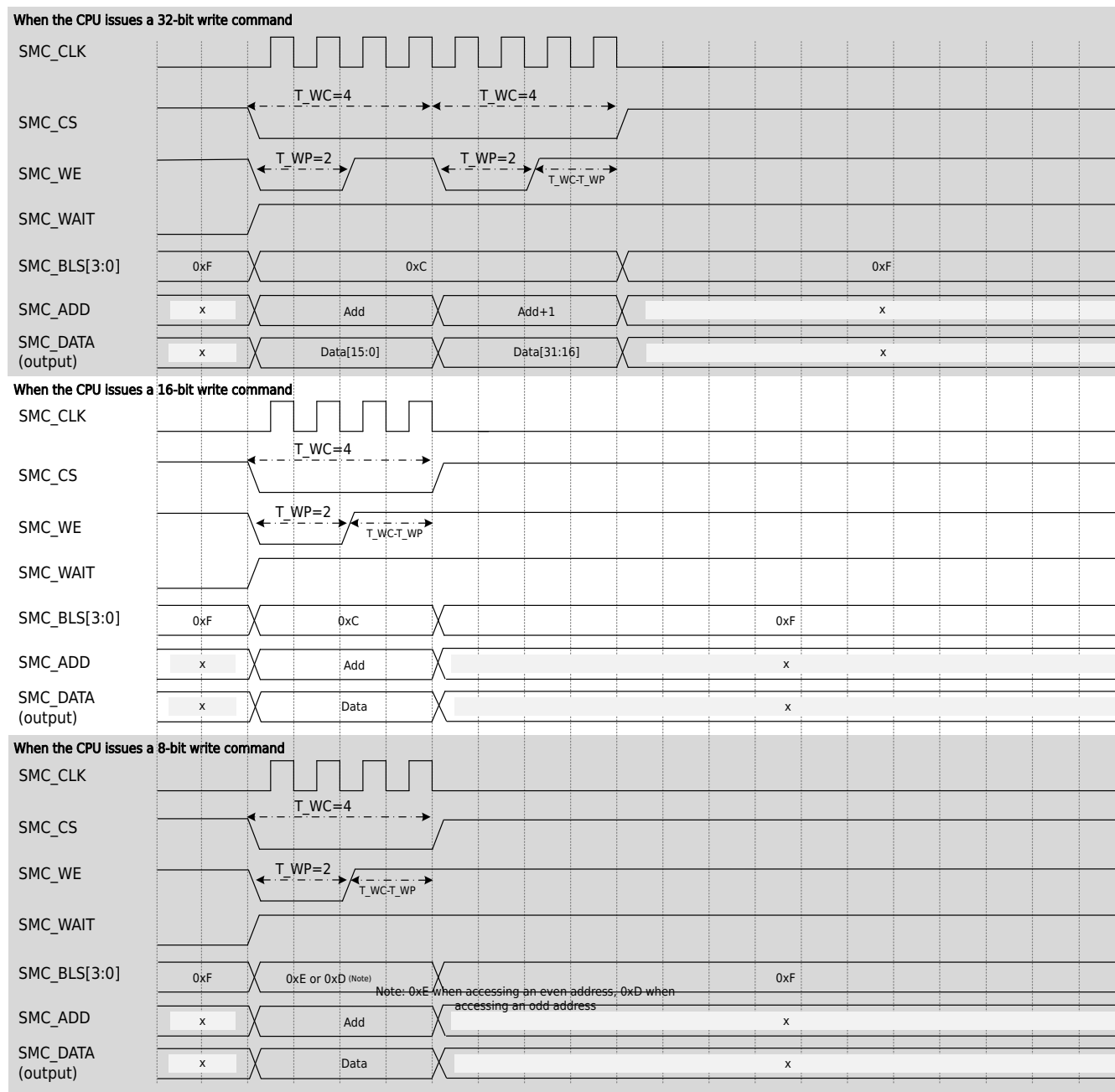


Figure 40-21 Basic Timing for Single Write Action (WSYN=0b1) & 16-bit Width (MW=0b01) & BLSS=0b0)

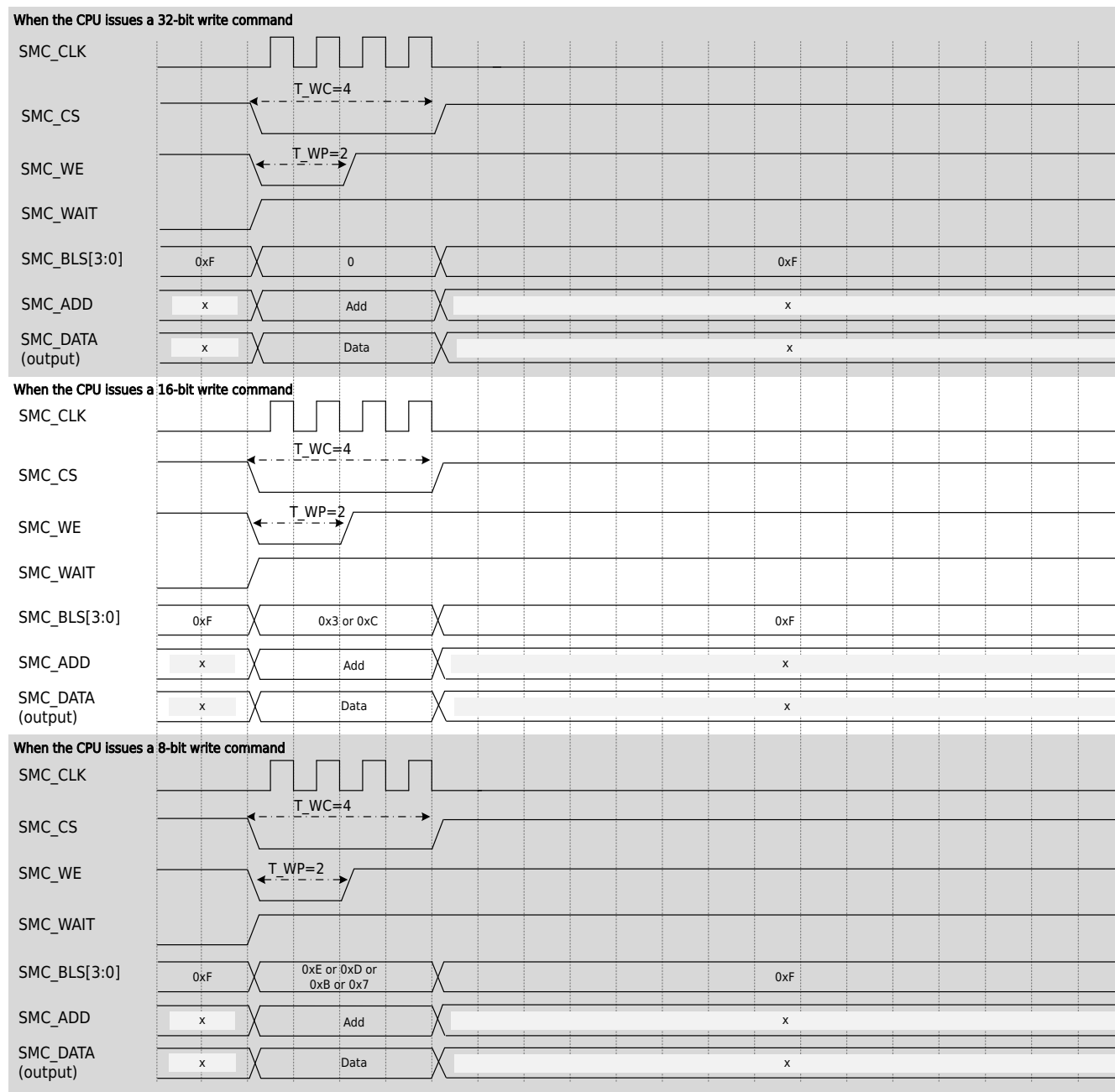


Figure 40-22 Basic Timing for Single Write Action (Synchronous Mode (WSYN=0b1) & 32-bit Width (MW=0b10) & BLSS=0b0)

## Single Write Action with Address and Data Line Multiplexing in Asynchronous Mode

The basic timing diagram and setup example for a single write action with address and data line multiplexing in asynchronous mode are as follows:

**Table 40-10 Basic Setup Example for a Single Write Action with Address and Data Line Multiplexing in Asynchronous Mode**

|                      |       |        |        |       |       |       |                                                                                  |
|----------------------|-------|--------|--------|-------|-------|-------|----------------------------------------------------------------------------------|
| Basic set-<br>tings  | MW    | RSYN   | WSYN   | -     | ADVS  | BLSS  | - indicates<br>not appli-<br>cable.<br><set> indi-<br>cates user-<br>set values. |
|                      | <set> | -      | 0      | -     | 0b1   | 0b1   |                                                                                  |
| Timing set-<br>tings | T_RC  | T_WC   | T_CEOE | T_WP  | T_TR  | T_ADV |                                                                                  |
|                      | -     | 0b0101 | -      | 0b010 | 0b010 | 0b001 |                                                                                  |

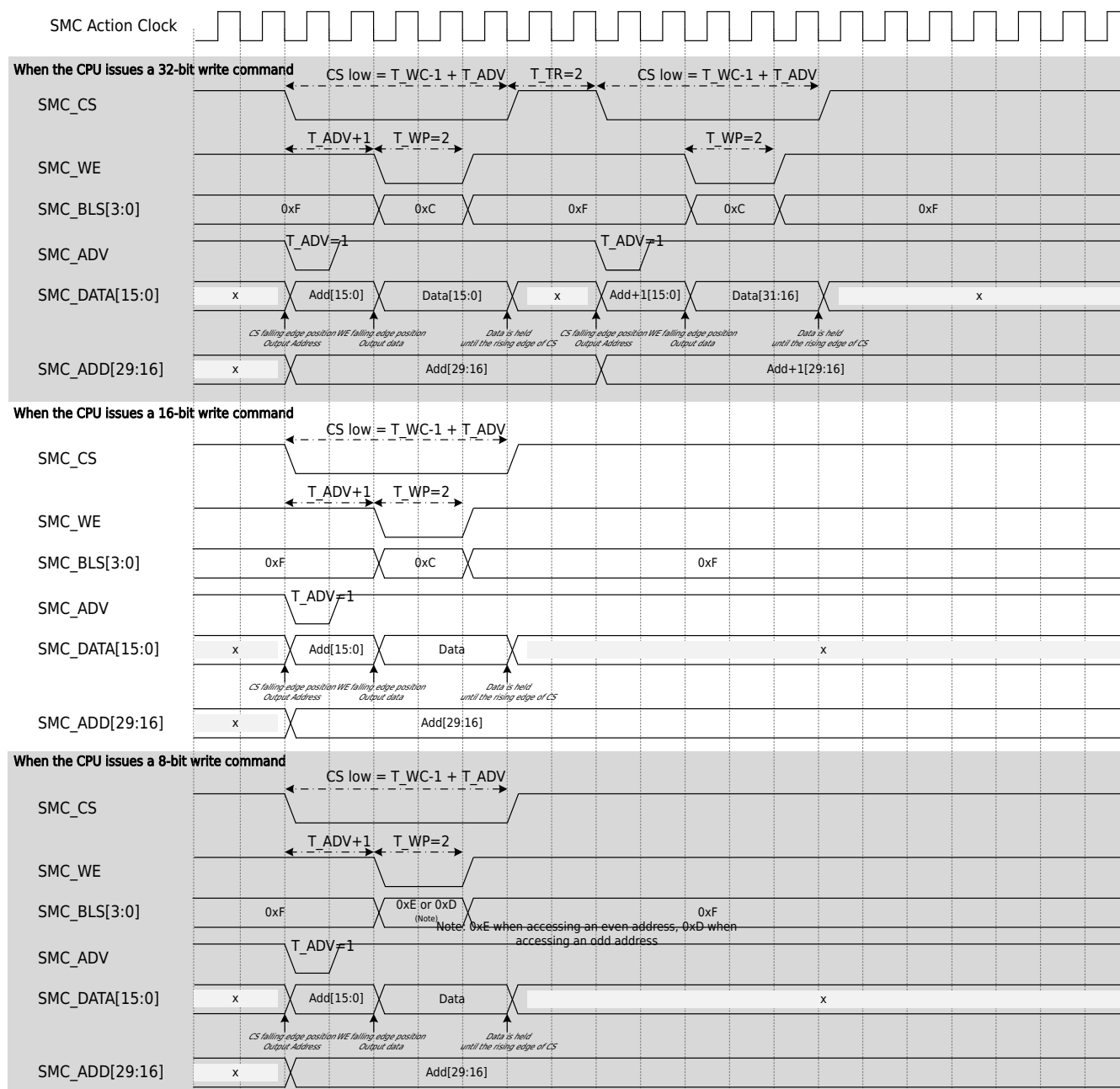


**Figure 40-23 Basic Timing for Single Write Action with Address and Data Line Multiplexing (Asynchronous mMode (WSYN=0b0) & 16-bit Width (MW=0b01) )**



### Single Write Action with Address and Data Line Multiplexing in Synchronous Mode

The basic timing diagram and setup example for a single write action with address and data line multiplexing in synchronous mode are as follows:

**Table 40-11 Basic Setup Example for a Single Write Action with Address and Data Line Multiplexing in Synchronous Mode**

|                      |       |        |        |       |       |       |                                                                         |
|----------------------|-------|--------|--------|-------|-------|-------|-------------------------------------------------------------------------|
| Basic set-<br>tings  | MW    | RSYN   | WSYN   | -     | ADVS  | BLSS  | - indicates<br>not appli-<br>cable.<br>indicates<br>user-set<br>values. |
|                      | <set> | -      | 1      | -     | 0b1   | 0b1   |                                                                         |
| Timing set-<br>tings | T_RC  | T_WC   | T_CEOE | T_WP  | T_TR  | T_ADV |                                                                         |
|                      | -     | 0b0101 | -      | 0b011 | 0b010 | 0b001 |                                                                         |

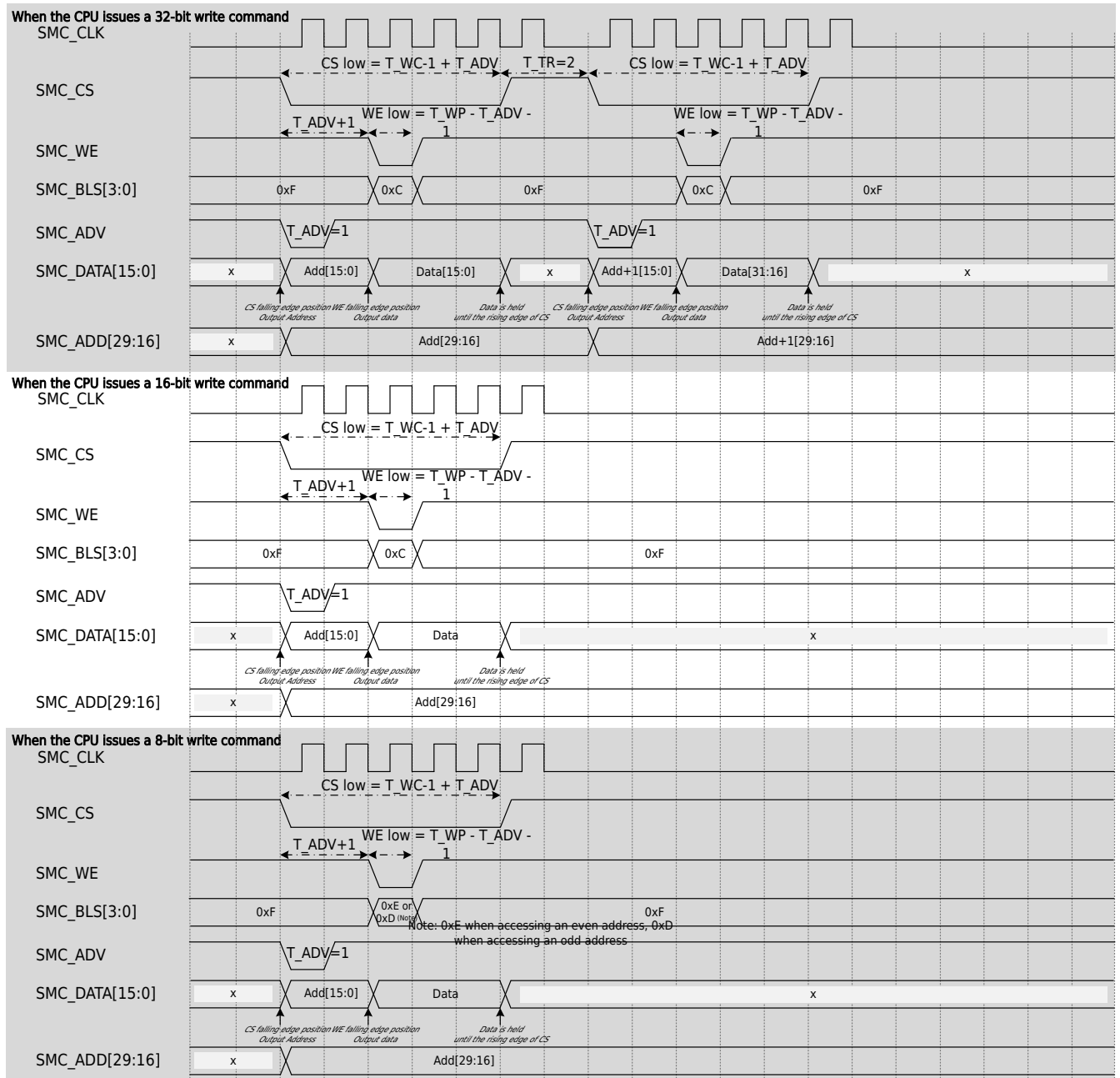


**Figure 40-24 Basic Timing for Single Write Action with Address and Data Line Multiplexing (Synchronous Mode (WSYN=0b1) & 16-bit Width (MW=0b01) )**

#### 40.3.5.4.3 FIFO Management

SMC contains 3 types of FIFOs. These are command FIFOs, write data FIFOs, and read data FIFOs.

- Command FIFO: 2 command FIFOs for buffering 2 SMC commands and their corresponding command addresses, CS, etc.
- Write data FIFO: 4 write data FIFOs for buffering AHB write data.
- Read data FIFO: 4 read data FIFOs for buffering data read from SRAM/PSRAM/NOR Flash.

#### 40.3.5.5 SMC Low Power Management

When SMC is in Low Power state, if the LPWOR bit of the state control register (SMC\_STCR1) is set to 1, SMC transitions from the Low Power state to the Ready state and starts normal operation. If there is no operation on SMC for a long time, the LPWIR bit of the state control register (SMC\_STCR0) can be set to 1 to put SMC into the Low Power state. After entering the Low Power state, SMC's internal actions stop, and power consumption decreases.

### 40.3.6 DMC-SDRAM Controller

#### 40.3.6.1 SDRAM Introduction

Synchronous Dynamic Random Access Memory (SDRAM) is a type of DRAM refreshed based on an external synchronous clock. Its synchronous clock is provided by the DMC\_CLK pin of DMC.

#### Bank Characteristics

SDRAM can be internally divided into multiple Banks (EXMC supports storage with up to 4 Banks), allowing devices to access them in an interleaved manner for greater concurrency and data transfer capacity. Each Bank is a memory matrix composed of rows and columns. The selection of rows and columns locates a storage unit, which corresponds to a space of the memory storage width. Therefore, the size of each Bank in SDRAM can be considered as memory data width x number of rows x number of columns. Users can match the parameters of DMC and external SDRAM by setting the relevant parameters (row address length, column address length, etc.) in the chip select control register (DMC\_CSCR), enabling communication with different SDRAMs.

#### Row Activation

Row activation enables the Bank where the row address is located. The complete row address consists of a 2-Bit Bank address (DMC\_BA[1:0]) and a 16-Bit row address (DMC\_ADD[15:0]). Row activation reads all bit information of the selected row into the read and write amplifier. Once row activation is complete, read and write operations can proceed smoothly.

Row activation requires a certain amount of time, called the row-to-column delay, which is the minimum time interval between row addressing and column addressing. The row-to-column delay ( $T_{RCD}$ ) of DMC is the minimum number of clock periods that includes the SDRAM row-to-column delay. It represents the minimum wait time from row enable to SDRAM read and write. This time can be set via the timing configuration register (DMC\_TMCR\_T\_RCD). During this time, since the SDRAM controller operates on Banks independently, users can issue control commands to other Bank addresses. The figure below shows the timing diagram from row activation to read and write operations.

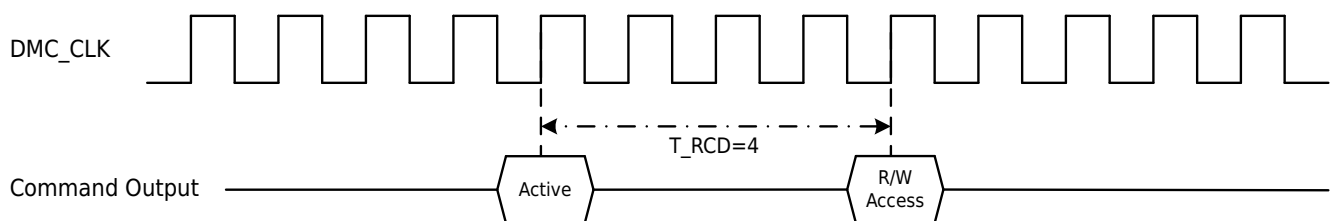


Figure 40-25 Timing from Row Activation to Read and Write Operations

**Note**

Row activation is represented as "Active" in the timing diagrams of the EXMC chapter.

**Precharge**

If DMC needs to perform row switching when accessing external data, the originally active (working) row of the Bank must be closed, and the row and column addresses must be resent. The operation of closing the current working row of a Bank and preparing to open a new row is called precharging. The precharge action can be triggered by the precharge command (PrechargeAll) in the command register (DMC\_CMDR) or during read and write row switching.

In burst read or burst write commands, the A10/A8 bit (set by DMC\_CPCR.APBS) becomes "H", and a precharge action is automatically appended after the read and write operation is completed, achieving auto-precharge.

The row precharge delay ( $T_{RP}$ ) represents the minimum time for SDRAM row switching. It is the minimum time interval from precharge completion to the next execution of enable command. This time can be set via the timing configuration register (DMC\_TMCR\_T\_RP). Below are timing diagrams for two different precharge methods.

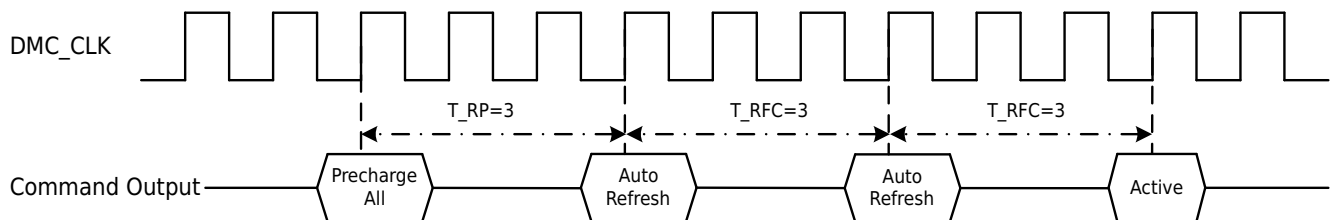


Figure 40-26 Command-based Precharge

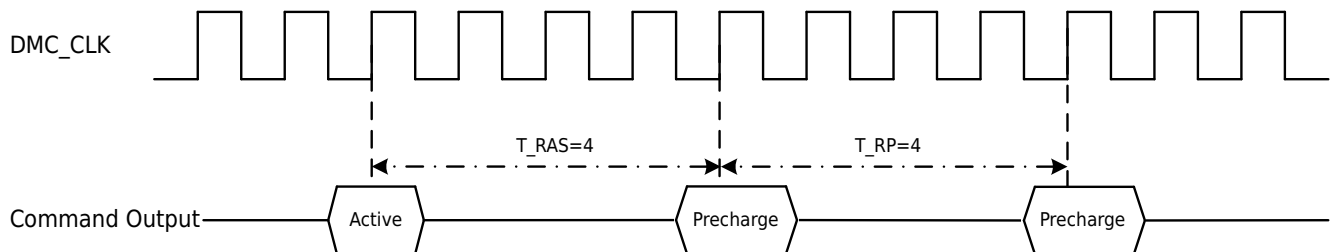


Figure 40-27 Auto-precharge after Read and Write

**Refresh Action**

Due to leakage in the storage cells of dynamic memory, periodic refreshing of all storage cells is necessary to maintain data correctness. EXMC supports two refresh methods: Auto Refresh (AR) and Self Refresh (SR).

Auto Refresh (AR) involves EXMC periodically issuing refresh commands. The refresh interval is determined by the refresh time register (DMC\_RFTR). The refresh targets all Banks (corresponding rows of all Banks are refreshed simultaneously), and auto-refresh operations are initiated only when all Banks are idle and not in a low power state. During auto-refresh execution, only no-operation commands can be input. Other commands must wait and cannot be executed. After auto-refresh execution completes, all Banks enter the idle state.

Self Refresh (SR) is mainly used for data storage in low power states during Sleep mode. When DMC\_CKE is inactive during an AR command, the self-refresh mode is entered. During SR, SDRAM no longer relies on the DMC\_CLK clock but performs refresh operations based on its internal clock. During SR, all

external signals except DMC\_CKE are inactive (no external refresh commands are required). When DMC transitions from the Pause state to the Low Power state (by issuing a Sleep command), it automatically issues a self-refresh command. The figure below shows the timing diagram for entering and exiting the self-refresh action.

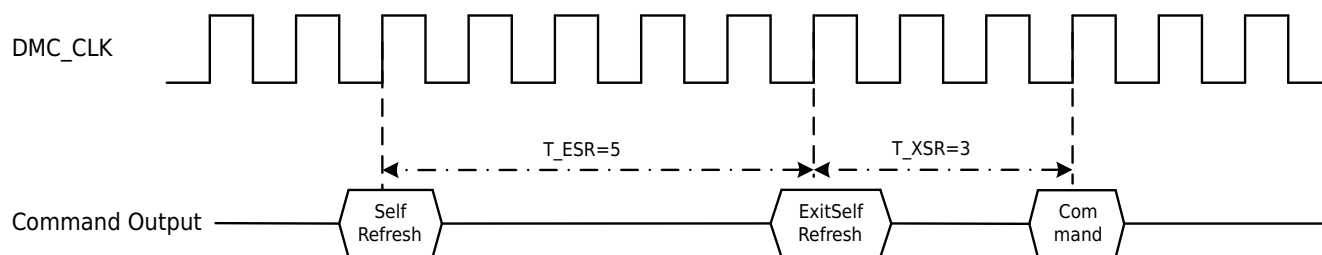


Figure 40-28 Timing for Entering and Exiting the Self-refresh Action

### Command Truth Table

Various commands for controlling and accessing SDRAM are issued by the SDRAM protocol interface of DMC. The specific command list is shown in the table below.

Table 40-12 DMC Command Truth Table

| DMC_CS | DMC_RAS | DMC_CAS | DMC_WE | DMC_BA | DMC_ADD |             | Command                                |
|--------|---------|---------|--------|--------|---------|-------------|----------------------------------------|
|        |         |         |        |        | ADD[xx] | ADD[10]/[8] |                                        |
| 0      | 0       | 0       | 0      | 0      | Config  |             | MdRegConfig Command                    |
| 0      | 0       | 0       | 1      | -      | -       | -           | SelfRefresh command (DMC_CKE=0).       |
|        |         |         |        |        |         |             | AutoRefresh Command (DMC_CKE=1)        |
| 0      | 0       | 1       | 0      | -      | -       | 1           | PrechargeAll Command                   |
|        |         |         |        | Bank   | -       | 0           | Precharge command for a specified Bank |
| 0      | 0       | 1       | 1      | Bank   | Row     | Row         | Row activation command                 |
| 0      | 1       | 0       | 0      | Bank   | Col     | 1           | Precharge command during write action  |
|        |         |         |        | Bank   | Col     | 0           | Write operation command                |
| 0      | 1       | 0       | 1      | Bank   | Col     | 1           | Precharge command during read action   |
|        |         |         |        | Bank   | Col     | 0           | Read operation command                 |
| 0      | 1       | 1       | 0      | -      | -       | -           | Abort burst transfer                   |
| 0      | 1       | 1       | 1      | -      | -       | -           | No operation                           |
| 1      | -       | -       | -      | -      | -       | -           | Command Inhibit                        |

### 40.3.6.2 DMC Basic Functions

The basic functions and features of DMC are as follows:

- Supports 16-bit and 32-bit external memory data bandwidth.
- Supports up to 16-bit row addresses, 12-bit column addresses, and 2-bit internal Bank addresses.
- AHB word, half-word, and byte access.
- Provides independent chip select control for each memory Chip.
- The size of each memory Chip can be independently configured.
- Byte selection signal output.
- Automatic row and Bank boundary management.

- Programmable protocol timing parameters.
- Programmable-rate auto-refresh action.
- With 2 41-bit command FIFOs.
- With 10 36-bit write data FIFOs
- With 10 32-bit read data FIFOs.
- Low power management.

#### 40.3.6.3 DMC Initial Setting

The state diagram and transitions between states of DMC after Power On Reset are shown below.

During initial reset, DMC enters the Config state. If a Go command is issued via the state control register (DMC\_STCR) and no operations follow, DMC automatically enters the Low Power state. During normal action, state transitions of DMC can be controlled by settings in the state control register (DMC\_STCR). For details, refer to "[DMC Low Power Management](#)".

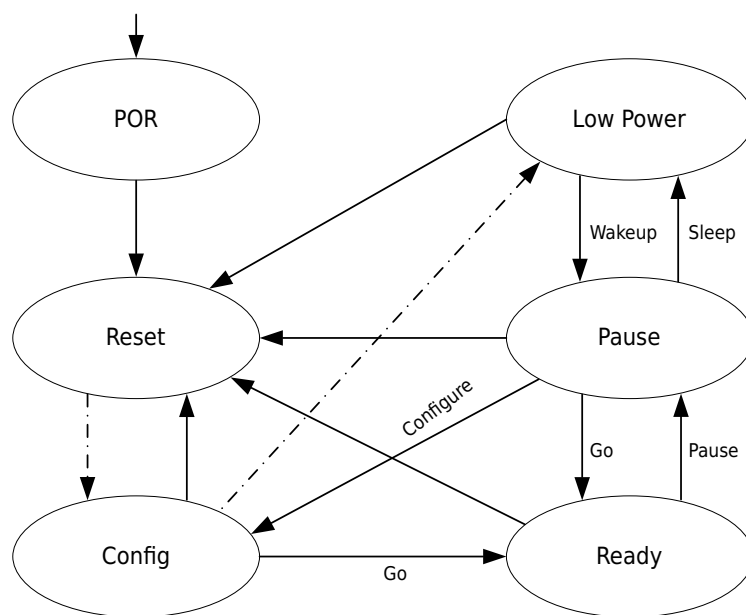


Figure 40-29 DMC State Diagram

Before communicating with external SDRAM, DMC must be initially set up and the relevant parameters configured to ensure correct data exchange. For the specific setup sequence, refer to the figure below.

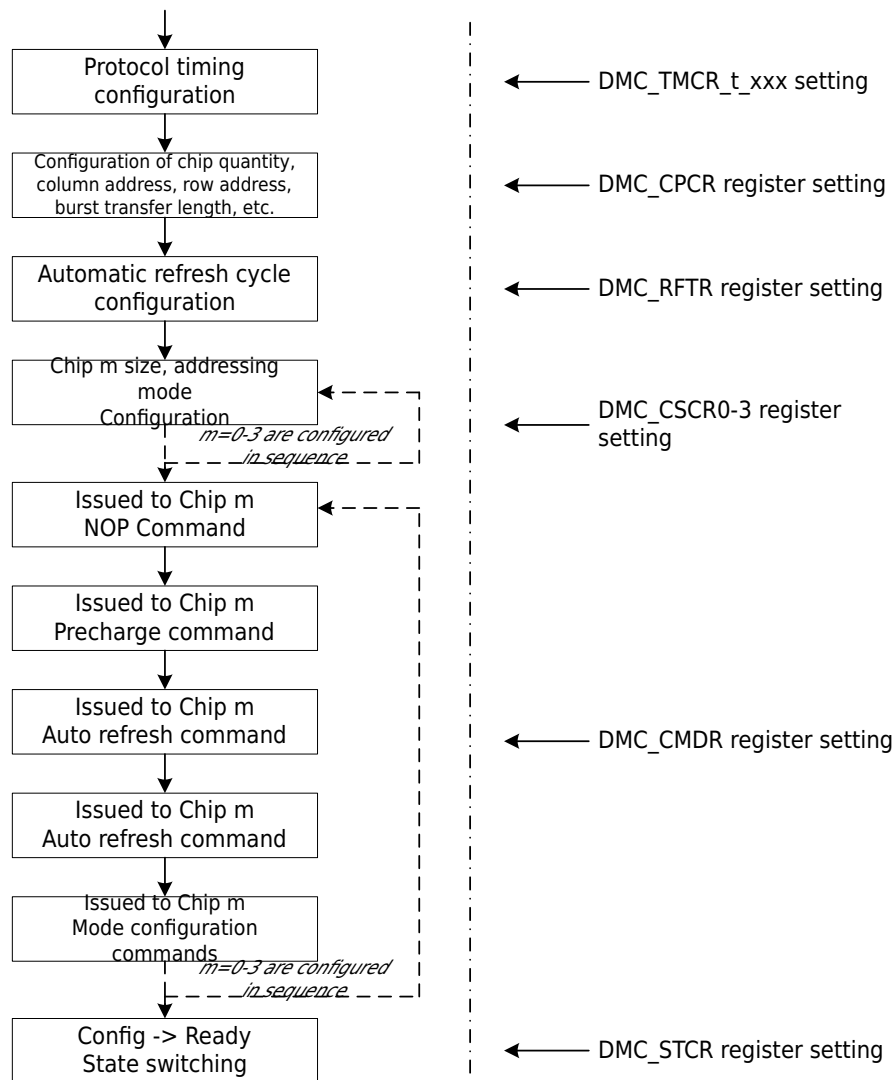


Figure 40-30 DMC Initial Setup Process

#### 40.3.6.4 DMC Access Actions

##### 40.3.6.4.1 Access Timing

DMC can convert AHB single or burst read and write operations into memory read and write operations. The following are methods for data read and write operations with external SDRAM:

1. [Single Read Action](#)
2. [Single Write Action](#)
3. [Burst Read Action](#)
4. [Burst Write Action](#)

##### Single Read Action

The figure below shows the basic timing diagram for a single read action.

In the figure, `cas_latency` is the delay from the read command to data readout (the DMC block reads data on the rising edge of `DMC_CLK`). This can be set via the timing configuration register (`DMC_TMCR_T_CASL`).

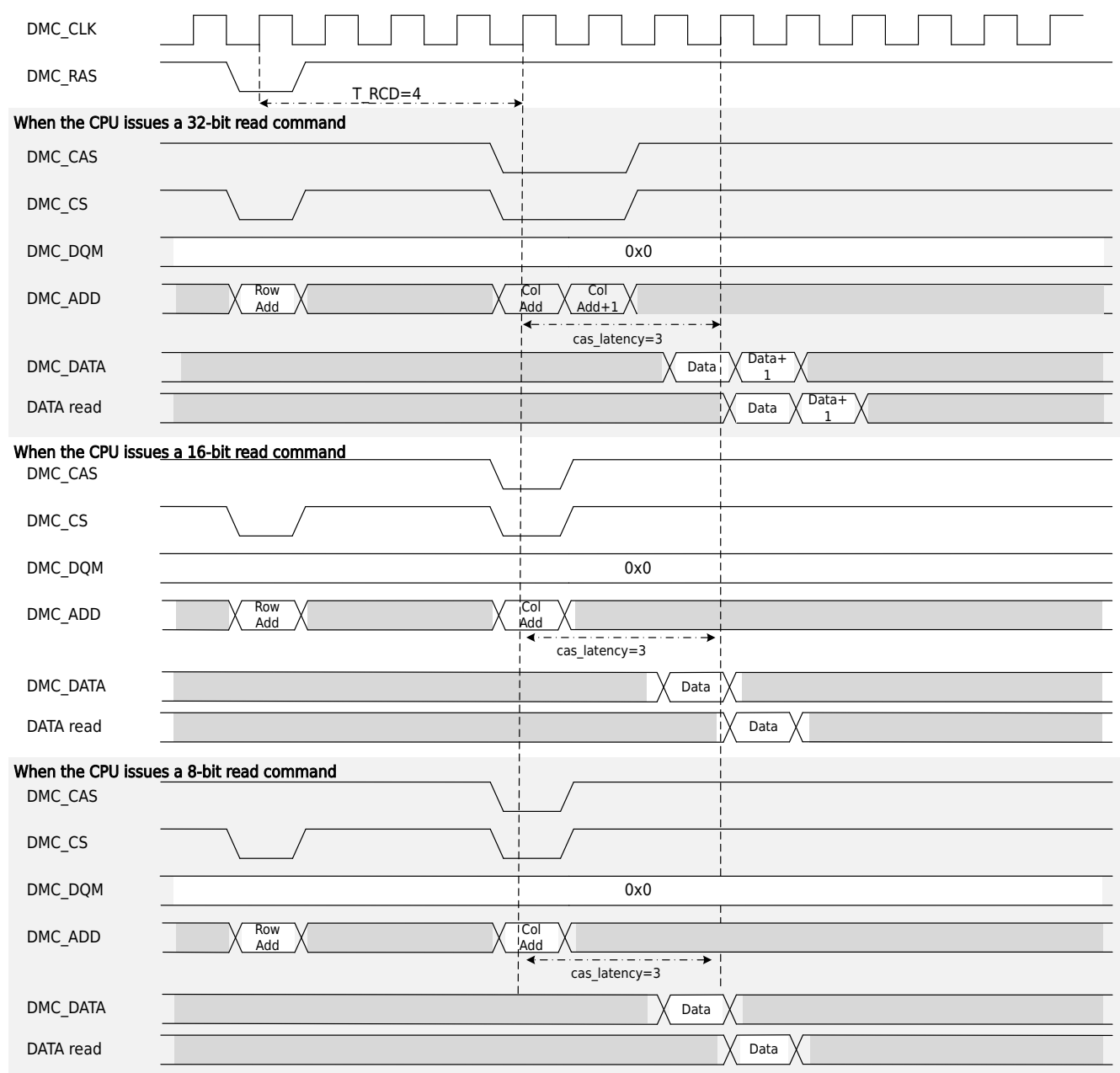


Figure 40-31 Basic Timing for Single Read Action (16-bit Width (DMCMW=0b00) )

## Single Write Action

The figure below shows the basic timing diagram for a single write action.

In the figure,  $t_{dqss}$  is the delay between write enable and write data. This can be configured via the timing configuration register (DMC\_TMCR\_T\_DQSS).

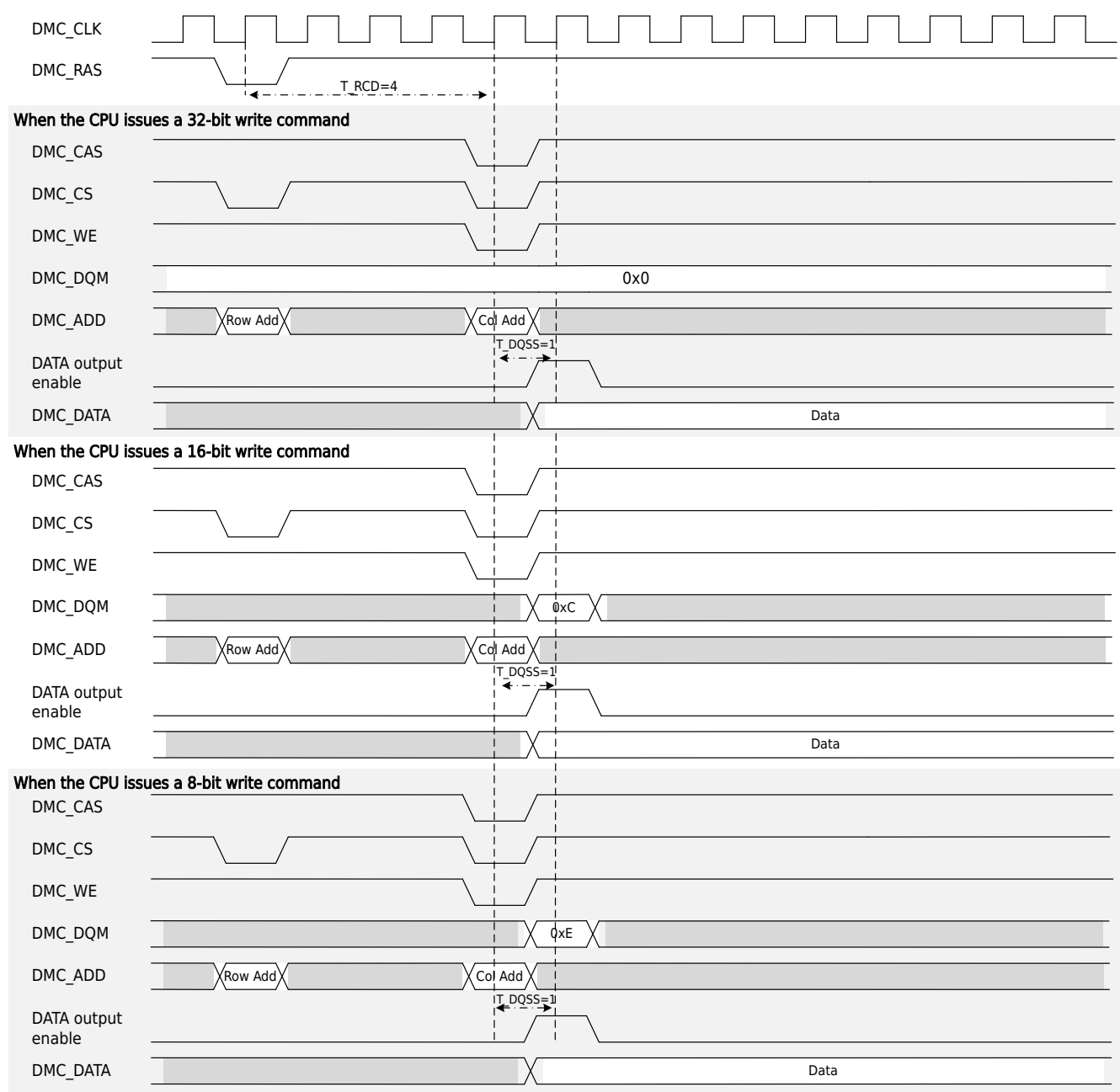


Figure 40-32 Basic Timing for Single Write Action (32-bit Width (DMCMW=0b01) )



### Burst Read Action

The figure below shows the basic timing diagram for a burst read action.

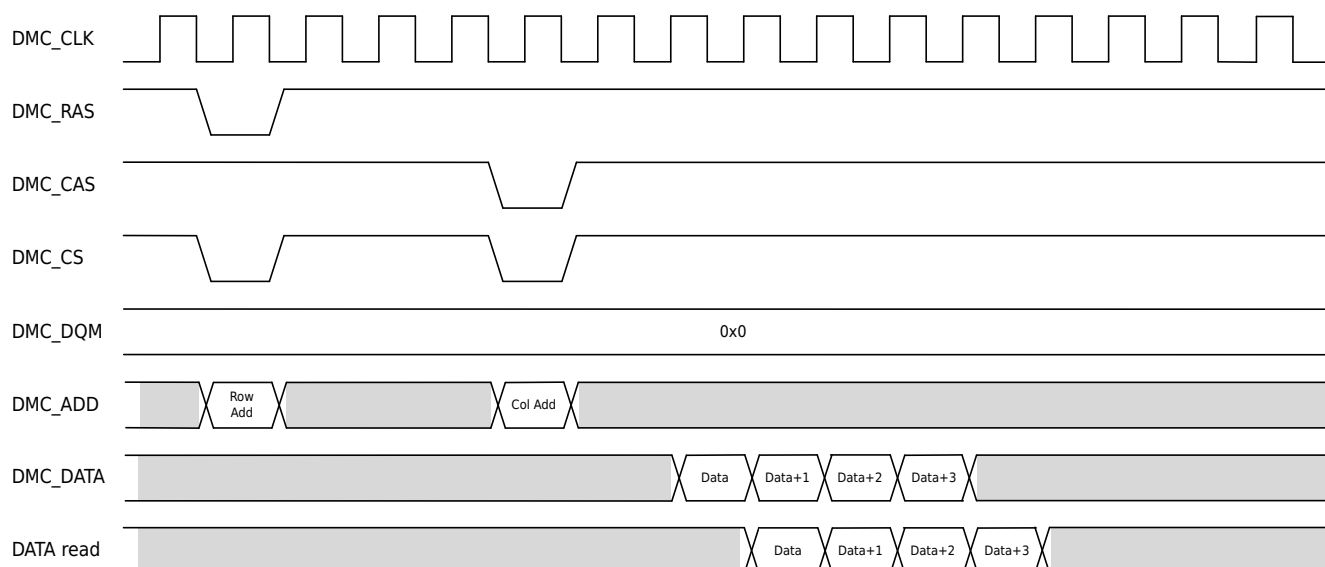


Figure 40-33 Basic Timing for Burst Read Action (32-bit Width (DMCMW=0b10) & BURST=0b010)

### Burst Write Action

The figure below shows the basic timing diagram for a burst write action.

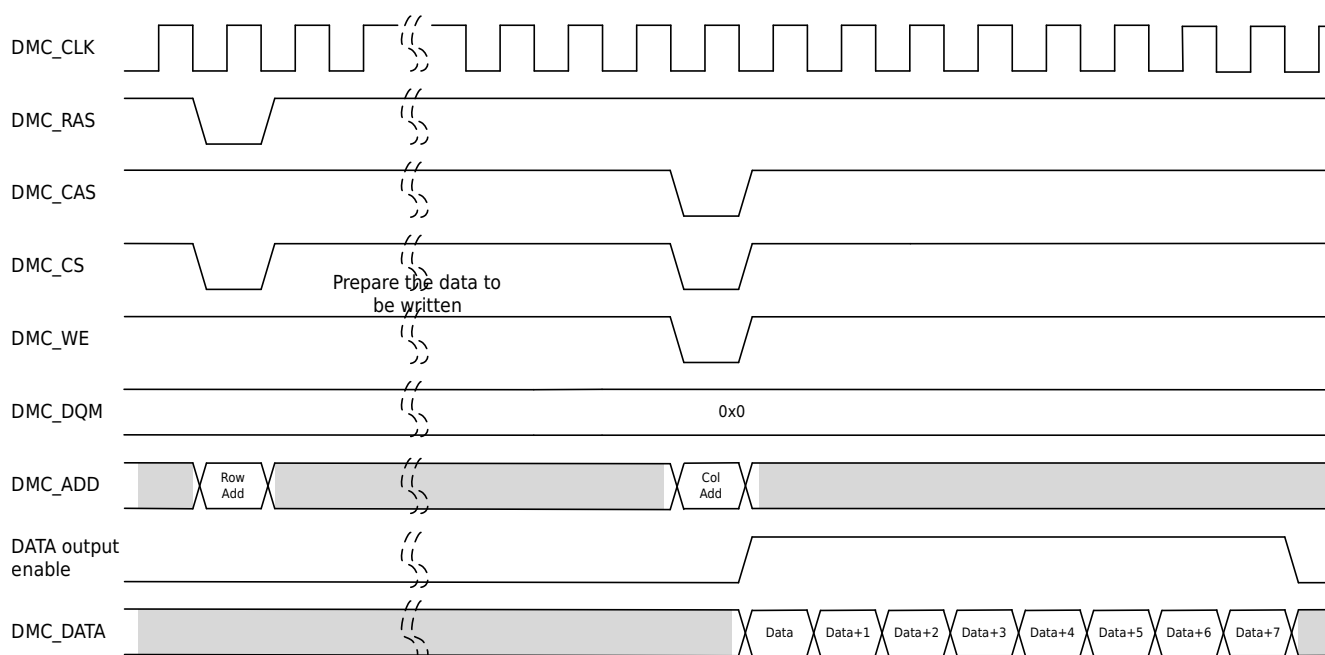


Figure 40-34 Basic Timing for Burst Write Action (32-bit Width (DMCMW=0b10) & BURST=0b011)

#### 40.3.6.4.2 FIFO Management

DMC contains 3 types of FIFOs. These are command FIFOs, write data FIFOs, and read data FIFOs.

**Command FIFO:** 2 command FIFOs for buffering 2 DMC commands and their corresponding command addresses, CS, etc.

**Write data FIFO:** 10 write data FIFOs for buffering AHB write data.

**Read data FIFO:** 10 read data FIFOs for buffering data read from SDRAM.

### 40.3.6.5 DMC Low Power Management

DMC can switch between the Ready state and the Low Power state internally to manage power consumption. It can also control the DMC output clock or clock enable during idle states to manage the power consumption of external SDRAM.

#### Low Power State Control

When DMC is in the Config state, issuing Go, Pause, and Sleep commands in sequence via the state control register (DMC\_STCR) puts DMC into the Low Power state. After entering the Low Power state, DMC's internal actions stop, and power consumption decreases. In the Low Power state, issuing Wakeup and Go commands in sequence via the state control register (DMC\_STCR) causes DMC to exit the Low Power state and enter the Ready state, starting normal action.

#### DMC\_CLK Output Control

When the CKSTOP bit of the CHIP configuration register (DMC\_CPCR) is set to 1, the clock output of the DMC\_CLK port automatically stops when DMC is not operating on external SDRAM. When a new operation occurs, the DMC\_CLK clock restarts. After the action completes, DMC\_CLK stops again automatically, achieving control over the DMC\_CLK clock output during idle states and thereby controlling the power consumption of DMC and SDRAM.

#### DMC\_CKE Output Control

When the CKEDIS bit of the CHIP configuration register (DMC\_CPCR) is set to 1, after DMC performs an operation on external SDRAM and the period set by DMC\_CPCR.CKEDISPRD[5:0] elapses, DMC becomes idle, and the port output of DMC\_CKE becomes inactive. When a new operation occurs, DMC\_CKE becomes active again. After the action completes, DMC\_CKE becomes inactive again, achieving control over the clock output enable DMC\_CKE during idle states and thereby controlling the power consumption of SDRAM. The following figure shows the timing diagram for DMC\_CKE port output control.

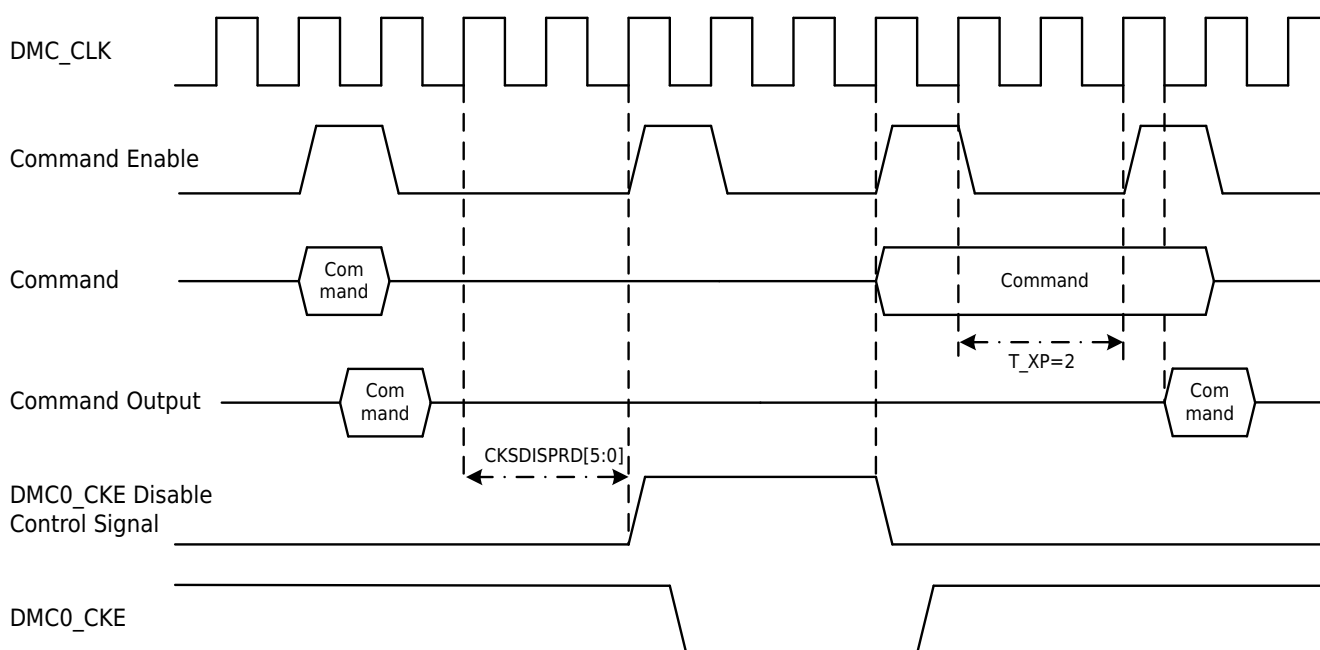


Figure 40-35 DMC\_CKE Port Output Control Timing

## 40.3.7 NFC-NAND Flash Controller

### 40.3.7.1 Introduction to NAND Flash

NAND Flash memory is a type of Flash memory that uses a nonlinear macrocell mode internally, providing a cost-active solution for solid-state high-capacity memory. NAND Flash memory has advantages such as large capacity and fast rewrite speeds, making it suitable for storing large amounts of data. It is increasingly widely used in industries, such as in embedded products like digital cameras, MP3 player memory cards, and compact USB drives.

### 40.3.7.2 NFC Basic Functions

The basic functions and features of NFC are as follows:

- Supports the ONFI protocol, enabling simple access to SLC and MLC NAND Flash devices.
- Supports devices with page sizes of 2KB/4KB/8KB.
- Supports devices with 8-bit and 16-bit bit width.
- Supports 1-bit Hamming code ECC correction per 512Byte unit.
- Can calculate 4-bit BCH code per 512Byte unit and provide the syndrome when an ECC error occurs.
- Access timing can be configured via registers

### 40.3.7.3 NFC Access Commands

The table below lists the ONFI access commands supported by NFC. "Mandatory" indicates commands that all ONFI devices must support. "Optional" indicates commands that some ONFI devices may support.

Table 40-13 ONFI Access Commands

| Command                      | Mandatory (M) / Optional (O). | First Period (Command). | Second Period (Confirm). |
|------------------------------|-------------------------------|-------------------------|--------------------------|
| Read                         | M                             | 0x00                    | 0x30                     |
| Copyback read                | O                             | 0x00                    | 0x35                     |
| Change read column           | M                             | 0x05                    | 0xE0                     |
| Change read column enhanced  | O                             | 0x06                    | 0xE0                     |
| Read cache random            | O                             | 0x00                    | 0x31                     |
| Read cache sequential        | O                             | 0x31                    | -                        |
| Read cache end               | O                             | 0x3F                    | -                        |
| Block erase                  | M                             | 0x60                    | 0xD0                     |
| Block erase interleaved      | O                             | 0x60                    | 0xD1                     |
| Read status                  | M                             | 0x70                    | -                        |
| Read status enhanced         | O                             | 0x78                    | -                        |
| Page program                 | M                             | 0x80                    | 0x10                     |
| Page program interleaved     | O                             | 0x80                    | 0x11                     |
| Page cache program           | O                             | 0x80                    | 0x15                     |
| Copyback program             | O                             | 0x85                    | 0x10                     |
| Copyback program interleaved | O                             | 0x85                    | 0x11                     |

| Command             | Mandatory (M) / Optional (O). | First Period (Command). | Second Period (Confirm). |
|---------------------|-------------------------------|-------------------------|--------------------------|
| Change write column | M                             | 0x85                    | -                        |
| Change row address  | O                             | 0x85                    | -                        |
| Read ID             | M                             | 0x90                    | -                        |
| Read parameter page | M                             | 0xEC                    | -                        |
| Read Unique ID      | O                             | 0xED                    | -                        |
| Get Features        | O                             | 0xEE                    | -                        |
| Set Features        | O                             | 0xEF                    | -                        |
| Reset LUN           | O                             | 0xFA                    | -                        |

#### 40.3.7.4 NFC Access Actions

##### Page Read

- Step1.** Write 0x00000000 to the command register.
- Step2.** Write the NAND Flash address to the index register.
- Step3.** Write 0x000000E0 to the command register or perform a read operation on the data register. At this time, the NAND Flash device pulls the RB signal low, indicating that the device is preparing data. When the RB signal returns to high, it indicates that the data is ready. Therefore, if the method of writing 0x000000E0 to the command register is used, wait until the RB signal is high before reading the data register. If a read operation is performed on the data register, the controller inserts a wait on the bus until RB is high and return the first data.
- Step4.** Read the entire page of data by reading the data register.
- Step5.** The user does not need to read all the data of the page. After reading the required data, write 0x00000023 to the command register. The controller automatically reads the remaining data and starts calculating ECC. When the ECC calculation is completed, an interrupt is generated, and the flag bit is set. The ECC result is written to the ECC result register. If the user selects ECC function inactive, this step is omitted.
- Step6.** Write 0x000000FE to the command register to end the current read operation and set CE to inactive.

##### Page Write

- Step1.** Write 0x00000080 to the command register.
- Step2.** Write the NAND Flash address to the index register.
- Step3.** Write data to the data register in sequence. When ECC is active, the controller calculates ECC while writing data.
- Step4.** After writing all the data, write 0x00000010 to the command register, and the NAND Flash device will start programming while pulling RB low. When ECC is enabled, if the data written by the user is less than 1 page, the controller will automatically fill the remaining address with FF and calculate the ECC value. (When ECCM is in 1-bit ECC mode: the ECC value is automatically written to the spare area of the NAND Flash device after writing the data; when ECCM is in 4-bit ECC mode: an 8-byte ECC value is written after every 512 bytes);
- Step5.** After programming is completed, the NAND Flash device restores RB to high. Write 0x000000FE to the command register to end the write operation and set CE to inactive.

### Block Erase

- Step1.** Write 0x81000M60 to the command register, where M is the BANK number;
- Step2.** Write 0x40000MAB to the command register, where M is the BANK number and AB is the least significant byte of the NAND Flash row address.
- Step3.** Write 0x40000MWX to the command register, where M is the BANK number and WX is the second lowest byte of the NAND Flash row address.
- Step4.** Write 0x40000MYZ to the command register, where M is the BANK number and YZ is the highest byte of the NAND Flash row address.
- Step5.** Write 0x00000MD0 to the command register, where M is the BANK number;
- Step6.** Wait for the NAND Flash device to restore RB to high.

### Reset

- Step1.** Write 0x00000MFF to the command register, where M is the BANK number;
- Step2.** Wait for the NAND Flash device to restore RB to high.

### Read ID

- Step1.** Read the data register once;
- Step2.** Write 0x81000M90 to the command register, where M is the BANK number;
- Step3.** Write 0x40000MAB to the command register, where M is the BANK number and AB is the address of the ID.
- Step4.** Read the data register to get the ID;
- Step5.** Writing 0x000000FE to the command register deasserts CE.

### Read Status

- Step1.** Write 0x81000M70 to the command register, where M is the BANK number;
- Step2.** Read the data register, the lowest byte of the register value is the status value;
- Step3.** Writing 0x000000FE to the command register deasserts CE.

## 40.3.8 Interrupt Description

NFC has three types of interrupt events. The interrupt enable register (NFC\_IENR) controls whether the corresponding interrupt occurs. The user can determine whether an interrupt event occurs by querying the interrupt status register (NFC\_ISTR) and clear the interrupt event by writing the register. The three types of NFC interrupt events are as follows:

1. An interrupt occurs when the ECC calculation is completed
2. Interrupt when ECC check error occurs
3. Each RB signal generates a rising edge, that is, an interrupt occurs when the corresponding device is ready

## 40.4 Register Description

### 40.4.1 Register List

EXMC common register base address: 0x40055400

Table 40-14 EXMC Common Register Offset

| Offset | Register  | Description          | Width |
|--------|-----------|----------------------|-------|
| 0x08   | NFC_STCR  | NFC Control Register | 32    |
| 0x28   | NFC_STSR  | NFC Status Register  | 32    |
| 0x0C   | EXMC_ENAR | EXMC Enable Register | 32    |

SMC base address: 0x88000000

Table 40-15 External Memory Controller (EXMC) SMC Register Offset

| Offset        | Register  | Description                      | Width |
|---------------|-----------|----------------------------------|-------|
| 0x0000        | SMC_STSR  | Status Register                  | 32    |
| 0x0010        | SMC_CMDR  | Command Register                 | 32    |
| 0x0008        | SMC_STCR0 | Status Control Register 0        | 32    |
| 0x000C        | SMC_STCR1 | Status Control Register 1        | 32    |
| 0x0200        | SMC_BACR  | Basic Control Register           | 32    |
| 0x0208        | SMC_CSCR0 | Chip Select Control Register     | 32    |
| 0x020C        | SMC_CSCR1 | Chip Select Control Register     | 32    |
| 0x0018        | SMC_CPCR  | CHIP Configuration Register      | 32    |
| 0x0104+0x20*x | SMC_CPSRx | CHIP Status Register x (x=0-3)   | 32    |
| 0x0014        | SMC_TMCR  | Timing Configuration Register    | 32    |
| 0x0100+0x20*x | SMC_TMSRx | Timing Status Register x (x=0-3) | 32    |

DMC base address: 0x88000400

Table 40-16 External Memory Controller (EXMC) DMC Register Offset

| Offset        | Register  | Description                            | Width |
|---------------|-----------|----------------------------------------|-------|
| 0x0000        | DMC_STSR  | Status Register                        | 32    |
| 0x0008        | DMC_CMDR  | Command Register                       | 32    |
| 0x0004        | DMC_STCR  | Status Control Register                | 32    |
| 0x0010        | DMC_RFTR  | Refresh Time Register                  | 32    |
| 0x0300        | DMC_BACR  | Basic Control Register                 | 32    |
| 0x0200+0x04*x | DMC_CSCRx | Chip Select Control Register x (x=0-3) | 32    |

| Offset | Register                        | Description                   | Width |
|--------|---------------------------------|-------------------------------|-------|
| 0x000C | <a href="#">DMC_CPCR</a>        | CHIP Configuration Register   | 32    |
| 0x0014 | <a href="#">DMC_TMCR_T_CASL</a> | Timing Configuration Register | 32    |
| 0x0018 | <a href="#">DMC_TMCR_T_DQSS</a> | Timing Configuration Register | 32    |
| 0x001C | <a href="#">DMC_TMCR_T_MRD</a>  | Timing Configuration Register | 32    |
| 0x0020 | <a href="#">DMC_TMCR_T_RAS</a>  | Timing Configuration Register | 32    |
| 0x0024 | <a href="#">DMC_TMCR_T_RC</a>   | Timing Configuration Register | 32    |
| 0x0028 | <a href="#">DMC_TMCR_T_RCD</a>  | Timing Configuration Register | 32    |
| 0x002C | <a href="#">DMC_TMCR_T_RFC</a>  | Timing Configuration Register | 32    |
| 0x0030 | <a href="#">DMC_TMCR_T_RP</a>   | Timing Configuration Register | 32    |
| 0x0034 | <a href="#">DMC_TMCR_T_RRD</a>  | Timing Configuration Register | 32    |
| 0x0038 | <a href="#">DMC_TMCR_T_WR</a>   | Timing Configuration Register | 32    |
| 0x003C | <a href="#">DMC_TMCR_T_WTR</a>  | Timing Configuration Register | 32    |
| 0x0040 | <a href="#">DMC_TMCR_T_XP</a>   | Timing Configuration Register | 32    |
| 0x0044 | <a href="#">DMC_TMCR_T_XSR</a>  | Timing Configuration Register | 32    |
| 0x0048 | <a href="#">DMC_TMCR_T_ESR</a>  | Timing Configuration Register | 32    |

NFC base address: 0x88100000

Table 40-17 External Memory Controller (EXMC) NFC Register Offset

| Offset        | Register                  | Description                     | Width |
|---------------|---------------------------|---------------------------------|-------|
| 0x0000-0x7FFC | <a href="#">NFC_DATR</a>  | Data Register                   | 32    |
| 0x8000        | <a href="#">NFC_CMDR</a>  | Command Register                | 32    |
| 0x8004        | <a href="#">NFC_IDXR0</a> | Index Register 0                | 32    |
| 0x8048        | <a href="#">NFC_IDXR1</a> | Index Register 1                | 32    |
| 0x8054        | <a href="#">NFC_BACR</a>  | Basic Configuration Register    | 32    |
| 0x8030        | <a href="#">NFC_IENR</a>  | Interrupt Enable Register       | 32    |
| 0x8034        | <a href="#">NFC_ISTR</a>  | Interrupt Status Register       | 32    |
| 0x8038        | <a href="#">NFC_IRSR</a>  | Interrupt Result Register       | 32    |
| 0x804C        | <a href="#">NFC_TMCR0</a> | Timing Configuration Register 0 | 32    |
| 0x8050        | <a href="#">NFC_TMCR1</a> | Timing Configuration Register 1 | 32    |
| 0x805C        | <a href="#">NFC_TMCR2</a> | Timing Configuration Register 2 | 32    |
| 0x8060-0x817C | <a href="#">NFC_ECCR</a>  | ECC Check Register              | 32    |

## 40.4.2 EXMC Common Registers

### 40.4.2.1 EXMC Enable Register (EXMC\_ENAR)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |    |       |    |       |  |       |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|----|-------|----|-------|--|-------|
| 0x0C   | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |    |       |    |       |  |       |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |    |       |    |       |  |       |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |    | NFCEN |    | SMCEN |  | DMCEN |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   | RW | RW    | RW |       |  |       |

| Bit  | Name     | Function    | Description                                                                                                                                   | Access |
|------|----------|-------------|-----------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:3 | Reserved | Reserved    | -                                                                                                                                             | -      |
| 2    | NFCEN    | NFC Enabled | <ul style="list-style-type: none"> <li>0b0: Function port of NFC module disabled</li> <li>0b1: Function port of NFC module enabled</li> </ul> | RW     |
| 1    | SMCEN    | SMC Enabled | <ul style="list-style-type: none"> <li>0b0: Function port of SMC module disabled</li> <li>0b1: Function port of SMC module enabled</li> </ul> | RW     |
| 0    | DMCEN    | DMC Enabled | <ul style="list-style-type: none"> <li>0b0: Function port of DMC module disabled</li> <li>0b1: Function port of DMC module enabled</li> </ul> | RW     |



40.4.3 SMC-SRAM/PSRAM/NOR Flash Register

40.4.3.1 Status Register (SMC\_STSR)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |        |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|--------|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0x0000 | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14     | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000001   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |        |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | STATUS |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RO     |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit  | Name     | Function           | Description                                                                       | Access |
|------|----------|--------------------|-----------------------------------------------------------------------------------|--------|
| 31:1 | Reserved | Reserved           | -                                                                                 | -      |
| 0    | STATUS   | SMC current status | <ul style="list-style-type: none"><li>0b0: Ready</li><li>0b1: Low Power</li></ul> | RO     |

## 40.4.3.2 Command Register (SMC\_CMDR)

| Offset | Bit Position |    |    |    |    |         |    |    |    |     |    |    |    |      |    |    |    |        |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |  |  |  |
|--------|--------------|----|----|----|----|---------|----|----|----|-----|----|----|----|------|----|----|----|--------|----|----|----|----|---|---|---|---|---|---|---|---|---|---|--|--|--|--|
| 0x0010 | 31           | 30 | 29 | 28 | 27 | 26      | 25 | 24 | 23 | 22  | 21 | 20 | 19 | 18   | 17 | 16 | 15 | 14     | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |  |  |  |  |
| Reset  | 0xFFFFFFFF   |    |    |    |    |         |    |    |    |     |    |    |    |      |    |    |    |        |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |  |  |  |
| Name   | Reserved     |    |    |    |    | CMDCHIP |    |    |    | CMD |    |    |    | CRES |    |    |    | CMDADD |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |  |  |  |
| Access |              |    |    |    |    | WO      |    |    |    | WO  |    |    |    | WO   |    |    |    | WO     |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |  |  |  |

| Bit   | Name     | Function               | Description                                                                                                                                                                 | Access |
|-------|----------|------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:26 | Reserved | Reserved               | -                                                                                                                                                                           | -      |
| 25:23 | CMDCHIP  | CS object setting      | <ul style="list-style-type: none"> <li>0b000: Chip 0</li> <li>0b001: Chip 1</li> <li>0b010: Chip 2</li> <li>0b011: Chip 3</li> <li>Others: Setting prohibited</li> </ul>    | WO     |
| 22:21 | CMD      | Command input          | <ul style="list-style-type: none"> <li>0b00: Setting prohibited</li> <li>0b01: MdRegConfig</li> <li>0b10: UpdateRegs</li> <li>0b11: MdRegConfig &amp; UpdateRegs</li> </ul> | WO     |
| 20    | CRES     | CRE port settings      | <ul style="list-style-type: none"> <li>0b0: SMC_CRE outputs low level</li> <li>0b1: When issuing the MdRegConfig command, the SMC_CRE port outputs high level</li> </ul>    | WO     |
| 19:0  | CMDADD   | Address object setting | When CMD=0b01 or 0b11, CMDADD[19:0] indicates the external MEM address to be accessed, and the SMC_ADD[19:0] pin outputs CMDADD[19:0]                                       | WO     |

## 40.4.3.3 Status Control Register 0 (SMC\_STCR0)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |       |   |          |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|-------|---|----------|---|
| 0x0008 | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3     | 2 | 1        | 0 |
| Reset  | 0XXXXXXXXX   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |       |   |          |   |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   | LPWIR |   | Reserved |   |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   | WO    |   |          |   |

| Bit  | Name     | Function                         | Description                                             | Access |
|------|----------|----------------------------------|---------------------------------------------------------|--------|
| 31:3 | Reserved | Reserved                         | -                                                       | -      |
| 2    | LPWIR    | Request to enter Low Power state | Write 1 to enter Low Power state, write 0 has no effect | WO     |
| 1:0  | Reserved | Reserved                         | -                                                       | -      |

## 40.4.3.4 Status Control Register 1 (SMC\_STCR1)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |       |   |          |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|-------|---|----------|---|
| 0x000C | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3     | 2 | 1        | 0 |
| Reset  | 0xFFFFFFFF   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |       |   |          |   |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   | LPWOR |   | Reserved |   |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   | WO    |   |          |   |

| Bit  | Name     | Function                     | Description                                            | Access |
|------|----------|------------------------------|--------------------------------------------------------|--------|
| 31:3 | Reserved | Reserved                     | -                                                      | -      |
| 2    | LPWOR    | Low Power state exit request | Write 1 to exit Low Power state, write 0 has no effect | WO     |
| 1:0  | Reserved | Reserved                     | -                                                      | -      |

## 40.4.3.5 Basic Control Register (SMC\_BACR)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |       |          |    |    |    |    |   |   |   |       |          |   |   |   |   |   |  |  |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-------|----------|----|----|----|----|---|---|---|-------|----------|---|---|---|---|---|--|--|
| 0x0200 | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15    | 14       | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6     | 5        | 4 | 3 | 2 | 1 | 0 |  |  |
| Reset  | 0x00000300   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |       |          |    |    |    |    |   |   |   |       |          |   |   |   |   |   |  |  |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | CKSEL | Reserved |    |    |    |    |   |   |   | MUXMD | Reserved |   |   |   |   |   |  |  |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RW    |          |    |    |    |    |   |   |   | RW    |          |   |   |   |   |   |  |  |

| Bit   | Name     | Function                      | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | Access |
|-------|----------|-------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:16 | Reserved | Reserved                      | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | -      |
| 15:14 | CKSEL    | Sampling clock selection      | <ul style="list-style-type: none"> <li>0b00: Use internal EXCLK when reading latched SMC_DATA and SMC_WAIT port data (only for asynchronous mode)</li> <li>0b01: When reading the latched SMC_DATA and SMC_WAIT port data, use the internal EXCLK inverted as the clock (only for asynchronous mode)</li> <li>0b10: When reading the latched SMC_DATA and SMC_WAIT port data, use the EXMC_CLK port status feedback as the clock (only for synchronous mode)</li> <li>0b11: Setting prohibited</li> </ul> | RW     |
| 13:10 | Reserved | Reserved                      | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | -      |
| 9:8   | Reserved | Reserved                      | All bits read as 1, Write as 1                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | -      |
| 7:5   | Reserved | Reserved                      | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | -      |
| 4     | MUXMD    | SMC address data multiplexing | <ul style="list-style-type: none"> <li>0b0: SMC address and data are not multiplexed</li> <li>0b1: SMC address and data multiplexing</li> </ul>                                                                                                                                                                                                                                                                                                                                                           | RW     |
| 3:0   | Reserved | Reserved                      | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | -      |

## 40.4.3.6 Chip Select Control Register (SMC\_CSCR0)

| Offset | Bit Position |    |    |    |    |    |    |    |         |    |    |    |    |    |    |    |         |    |    |    |    |    |   |   |         |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|---------|----|----|----|----|----|----|----|---------|----|----|----|----|----|---|---|---------|---|---|---|---|---|---|---|
| 0x0208 | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23      | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15      | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7       | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0xFFFFFFFF   |    |    |    |    |    |    |    |         |    |    |    |    |    |    |    |         |    |    |    |    |    |   |   |         |   |   |   |   |   |   |   |
| Name   | ADDMSK3      |    |    |    |    |    |    |    | ADDMSK2 |    |    |    |    |    |    |    | ADDMSK1 |    |    |    |    |    |   |   | ADDMSK0 |   |   |   |   |   |   |   |
| Access | RW           |    |    |    |    |    |    |    | RW      |    |    |    |    |    |    |    | RW      |    |    |    |    |    |   |   | RW      |   |   |   |   |   |   |   |

| Bit   | Name    | Function               | Description                    | Access |
|-------|---------|------------------------|--------------------------------|--------|
| 31:24 | ADDMSK3 | CS3 space mask address | CS3 space mask address setting | RW     |
| 23:16 | ADDMSK2 | CS2 space mask address | CS2 space mask address setting | RW     |
| 15:8  | ADDMSK1 | CS1 space mask address | CS1 space mask address setting | RW     |
| 7:0   | ADDMSK0 | CS0 space mask address | CS0 space mask address setting | RW     |

## 40.4.3.7 Chip Select Control Register (SMC\_CSCR1)

| Offset | Bit Position |    |    |    |    |    |    |    |         |    |    |    |    |    |    |    |         |    |    |    |    |    |   |   |         |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|---------|----|----|----|----|----|----|----|---------|----|----|----|----|----|---|---|---------|---|---|---|---|---|---|---|
| 0x020C | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23      | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15      | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7       | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |         |    |    |    |    |    |    |    |         |    |    |    |    |    |   |   |         |   |   |   |   |   |   |   |
| Name   | ADDMAT3      |    |    |    |    |    |    |    | ADDMAT2 |    |    |    |    |    |    |    | ADDMAT1 |    |    |    |    |    |   |   | ADDMAT0 |   |   |   |   |   |   |   |
| Access | RW           |    |    |    |    |    |    |    | RW      |    |    |    |    |    |    |    | RW      |    |    |    |    |    |   |   | RW      |   |   |   |   |   |   |   |

| Bit   | Name    | Function                   | Description                        | Access |
|-------|---------|----------------------------|------------------------------------|--------|
| 31:24 | ADDMAT3 | CS3 space matching address | CS3 space matching address setting | RW     |
| 23:16 | ADDMAT2 | CS2 space matching address | CS2 space matching address setting | RW     |
| 15:8  | ADDMAT1 | CS1 space matching address | CS1 space matching address setting | RW     |
| 7:0   | ADDMAT0 | CS0 space matching address | CS0 space matching address setting | RW     |

For example, when ADDMAT0 and ADDMSK0 are set to different values, the corresponding way of SMC access space is shown in the following table.

Table 40-34 CS0 Setting and Access Address Correspondence Table

| ADDMAT0[7:0] | ADDMSK0[7:0]            | Corresponding External Address | CS0 Space Size |
|--------------|-------------------------|--------------------------------|----------------|
| 0x70         | 0xFF                    | 0x70000000-0x70FFFFFF          | 16MB           |
|              | 0xFE                    | 0x70000000-0x71FFFFFF          | 32MB           |
|              | 0xFC                    | 0x70000000-0x73FFFFFF          | 64MB           |
|              | 0xF8                    | 0x70000000-0x77FFFFFF          | 128MB          |
|              | 0xF0                    | 0x70000000-0x7FFFFFFF          | 256MB          |
|              | Other settings disabled | -                              | -              |
| 0x68         | 0xFF                    | 0x68000000-0x68FFFFFF          | 16MB           |
|              | 0xFE                    | 0x68000000-0x69FFFFFF          | 32MB           |
|              | 0xFC                    | 0x68000000-0x6BFFFFFF          | 64MB           |
|              | 0xF8                    | 0x68000000-0x6FFFFFFF          | 128MB          |
|              | Other settings disabled | -                              | -              |

| ADDMAT0[7:0] | ADDMSK0[7:0]            | Corresponding External Address | CS0 Space Size |
|--------------|-------------------------|--------------------------------|----------------|
| 0x66         | 0xFF                    | 0x66000000-0x66FFFFFF          | 16MB           |
|              | 0xFE                    | 0x66000000-0x67FFFFFF          | 32MB           |
|              | Other settings disabled | -                              | -              |
| 0x64         | 0xFF                    | 0x64000000-0x64FFFFFF          | 16MB           |
|              | 0xFE                    | 0x64000000-0x65FFFFFF          | 32MB           |
|              | 0xFC                    | 0x64000000-0x67FFFFFF          | 64MB           |
|              | Other settings disabled | -                              | -              |
| 0x62         | 0xFF                    | 0x62000000-0x62FFFFFF          | 16MB           |
|              | 0xFE                    | 0x62000000-0x63FFFFFF          | 32MB           |
|              | Other settings disabled | -                              | -              |
| 0x60         | 0xFF                    | 0x60000000-0x60FFFFFF          | 16MB           |
|              | 0xFE                    | 0x60000000-0x61FFFFFF          | 32MB           |
|              | 0xFC                    | 0x60000000-0x63FFFFFF          | 64MB           |
|              | 0xF8                    | 0x60000000-0x67FFFFFF          | 128MB          |
|              | 0xF0                    | 0x60000000-0x6FFFFFFF          | 256MB          |
|              | 0xE0                    | 0x60000000-0x7FFFFFFF          | 512MB          |
|              | Other settings disabled | -                              | -              |

## 40.4.3.8 CHIP Configuration Register (SMC\_CPCR)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |      |    |      |   |          |    |   |          |   |      |   |          |  |      |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|------|----|------|---|----------|----|---|----------|---|------|---|----------|--|------|
| 0x0018 | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11   | 10 | 9    | 8 | 7        | 6  | 5 | 4        | 3 | 2    | 1 | 0        |  |      |
| Reset  | 0xFFFFFFFF   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |      |    |      |   |          |    |   |          |   |      |   |          |  |      |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | BLSS |    | ADVS |   | Reserved | MW |   | Reserved |   | WSYN |   | Reserved |  | RSYN |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | WO   |    | WO   |   |          | WO |   |          |   | WO   |   |          |  | WO   |

| Bit   | Name     | Function               | Description                                                                                                                                                                                              | Access |
|-------|----------|------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:13 | Reserved | Reserved               | -                                                                                                                                                                                                        | -      |
| 12    | BLSS     | BLS port settings      | <ul style="list-style-type: none"> <li>0b0: The BLS port status changes synchronously with the CS port status</li> <li>0b1: The BLS port status changes synchronously with the WE port status</li> </ul> | WO     |
| 11    | ADVS     | ADV port settings      | <ul style="list-style-type: none"> <li>0b0: ADV port function disabled</li> <li>0b1: When the address data line multiplexing function is enabled, the ADV port function is enable</li> </ul>             | WO     |
| 10    | Reserved | Reserved               | -                                                                                                                                                                                                        | -      |
| 9:8   | MW       | Memory width selection | <ul style="list-style-type: none"> <li>0b00: 8-bit width</li> <li>0b01: 16-bit width</li> <li>0b10: 32-bit width</li> <li>Others: Setting prohibited</li> </ul>                                          | WO     |
| 7:5   | Reserved | Reserved               | -                                                                                                                                                                                                        | -      |
| 4     | WSYN     | Write synchronization  | <ul style="list-style-type: none"> <li>0b0: Asynchronous write enabled</li> <li>0b1: Synchronous write enabled</li> </ul>                                                                                | WO     |
| 3:1   | Reserved | Reserved               | -                                                                                                                                                                                                        | -      |
| 0     | RSYN     | Read synchronization   | <ul style="list-style-type: none"> <li>0b0: Asynchronous read enabled</li> <li>0b1: Synchronous read enabled</li> </ul>                                                                                  | WO     |



## 40.4.3.9 CHIP Status Register x (SMC\_CPSRx) (x=0-3)

Offset: 0x0104+0x20\*x

| Offset | Bit Position |    |    |    |    |    |    |    |        |    |    |    |    |    |    |    |          |    |      |    |      |          |    |          |    |      |          |    |      |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|--------|----|----|----|----|----|----|----|----------|----|------|----|------|----------|----|----------|----|------|----------|----|------|---|---|---|
| -      | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23     | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15       | 14 | 13   | 12 | 11   | 10       | 9  | 8        | 7  | 6    | 5        | 4  | 3    | 2 | 1 | 0 |
| Reset  | 0x00FF0A00   |    |    |    |    |    |    |    |        |    |    |    |    |    |    |    |          |    |      |    |      |          |    |          |    |      |          |    |      |   |   |   |
| Name   | ADDMAT       |    |    |    |    |    |    |    | ADDMSK |    |    |    |    |    |    |    | Reserved | RO | BLSS | RO | ADVS | Reserved | MW | Reserved | RO | WSYN | Reserved | RO | RSYN |   |   |   |
| Access |              |    |    |    |    |    |    |    |        |    |    |    |    |    |    |    |          | RO | RO   | RO | RO   | RO       |    |          | RO | RO   |          |    |      |   |   |   |
|        |              |    |    |    |    |    |    |    |        |    |    |    |    |    |    |    |          |    |      |    |      |          |    |          |    |      |          |    |      |   |   |   |
|        |              |    |    |    |    |    |    |    |        |    |    |    |    |    |    |    |          |    |      |    |      |          |    |          |    |      |          |    |      |   |   |   |

| Bit   | Name     | Function                  | Description                                                                                                                                                                                              | Access |
|-------|----------|---------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:24 | ADDMAT   | CS space matching address | CS space matching address information                                                                                                                                                                    | RO     |
| 23:16 | ADDMSK   | CS space masking address  | CS space masking address information                                                                                                                                                                     | RO     |
| 15:13 | Reserved | Reserved                  | -                                                                                                                                                                                                        | -      |
| 12    | BLSS     | SMC_BLS status            | <ul style="list-style-type: none"> <li>0b0: The BLS port status changes synchronously with the CS port status</li> <li>0b1: The BLS port status changes synchronously with the WE port status</li> </ul> | RO     |
| 11    | ADVS     | SMC_ADV status            | <ul style="list-style-type: none"> <li>0b0: ADV port function disabled</li> <li>0b1: When the address data line multiplexing function is enabled, the ADV port function is enable</li> </ul>             | RO     |
| 10    | Reserved | Reserved                  | -                                                                                                                                                                                                        | -      |
| 9:8   | MW       | Memory width status       | <ul style="list-style-type: none"> <li>0b00: 8-bit width</li> <li>0b01: 16-bit width</li> <li>0b10: 32-bit width</li> <li>Other: Reserved</li> </ul>                                                     | RO     |
| 7:5   | Reserved | Reserved                  | -                                                                                                                                                                                                        | -      |
| 4     | WSYN     | Write action mode status  | <ul style="list-style-type: none"> <li>0b0: Asynchronous write enabled</li> <li>0b1: Synchronous write enabled</li> </ul>                                                                                | RO     |
| 3:1   | Reserved | Reserved                  | -                                                                                                                                                                                                        | -      |
| 0     | RSYN     | Read action mode status   | <ul style="list-style-type: none"> <li>0b0: Asynchronous read enabled</li> <li>0b1: Synchronous read enabled</li> </ul>                                                                                  | RO     |

## 40.4.3.10 Timing Configuration Register (SMC\_TMCR)

| Offset | Bit Position |    |    |    |       |    |    |    |          |      |    |    |    |          |    |    |    |      |    |    |    |          |        |   |   |   |      |   |   |   |      |   |  |  |
|--------|--------------|----|----|----|-------|----|----|----|----------|------|----|----|----|----------|----|----|----|------|----|----|----|----------|--------|---|---|---|------|---|---|---|------|---|--|--|
| 0x0014 | 31           | 30 | 29 | 28 | 27    | 26 | 25 | 24 | 23       | 22   | 21 | 20 | 19 | 18       | 17 | 16 | 15 | 14   | 13 | 12 | 11 | 10       | 9      | 8 | 7 | 6 | 5    | 4 | 3 | 2 | 1    | 0 |  |  |
| Reset  | 0xFFFFFFFF   |    |    |    |       |    |    |    |          |      |    |    |    |          |    |    |    |      |    |    |    |          |        |   |   |   |      |   |   |   |      |   |  |  |
| Name   | Reserved     |    |    |    | T_ADV |    |    |    | Reserved | T_TR |    |    |    | Reserved |    |    |    | T_WP |    |    |    | Reserved | T_CEOE |   |   |   | T_WC |   |   |   | T_RC |   |  |  |
| Access |              |    |    |    | WO    |    |    |    |          | WO   |    |    |    |          |    |    |    | WO   |    |    |    |          | WO     |   |   |   | WO   |   |   |   | WO   |   |  |  |

| Bit   | Name     | Function              | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | Access |
|-------|----------|-----------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:27 | Reserved | Reserved              | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | -      |
| 26:24 | T_ADV    | T_ADV period setting  | <p>T_ADV setting, the setting value needs to be greater than or equal to 1, and each chip sets the same value</p> <div>  <b>Note</b><br/>           Only valid when address data multiplexing.<br/>           When address data multiplexing is enabled, pay attention to the following when configuring T_ADV:<br/>           1. The T_ADV setting value of each chip SMC_TMCR must be greater than or equal to 1 and the same.<br/>           2. To configure T_ADV of chip1-3 SMC_TMCR, the UpdateRegs command must be issued to chip 0.         </div> | WO     |
| 23    | Reserved | Reserved              | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | -      |
| 22:20 | T_TR     | T_TR period setting   | <p>T_TR setting</p> <div>  <b>Note</b><br/>           Only valid when address data is reused, and the valid value is 1-7; write 1 when address data is not reused.         </div>                                                                                                                                                                                                                                                                                                                                                                        | WO     |
| 19:15 | Reserved | Reserved              | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | -      |
| 14:12 | T_WP     | T_WP period setting   | T_WP setting                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | WO     |
| 11    | Reserved | Reserved              | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | -      |
| 10:8  | T_CEOE   | T_CEOE period setting | T_CEOE setting, the setting value needs to be greater than T_ADV                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | WO     |
| 7:4   | T_WC     | T_WC period setting   | T_WC setting                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | WO     |
| 3:0   | T_RC     | T_RC period setting   | T_RC setting                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | WO     |

## 40.4.3.11 Timing Status Register x (SMC\_TMSRx) (x=0-3)

Offset: 0x0100+0x20\*x

| Offset | Bit Position |    |    |    |       |    |    |    |          |      |    |    |    |          |    |    |    |      |    |    |    |          |        |   |   |   |      |   |   |   |      |   |  |  |
|--------|--------------|----|----|----|-------|----|----|----|----------|------|----|----|----|----------|----|----|----|------|----|----|----|----------|--------|---|---|---|------|---|---|---|------|---|--|--|
| -      | 31           | 30 | 29 | 28 | 27    | 26 | 25 | 24 | 23       | 22   | 21 | 20 | 19 | 18       | 17 | 16 | 15 | 14   | 13 | 12 | 11 | 10       | 9      | 8 | 7 | 6 | 5    | 4 | 3 | 2 | 1    | 0 |  |  |
| Reset  | 0x011263CC   |    |    |    |       |    |    |    |          |      |    |    |    |          |    |    |    |      |    |    |    |          |        |   |   |   |      |   |   |   |      |   |  |  |
| Name   | Reserved     |    |    |    | T_ADV |    |    |    | Reserved | T_TR |    |    |    | Reserved |    |    |    | T_WP |    |    |    | Reserved | T_CEOE |   |   |   | T_WC |   |   |   | T_RC |   |  |  |
| Access |              |    |    |    | RO    |    |    |    |          | RO   |    |    |    |          |    |    |    | RO   |    |    |    |          | RO     |   |   |   | RO   |   |   |   | RO   |   |  |  |

| Bit   | Name     | Function             | Description   | Access |
|-------|----------|----------------------|---------------|--------|
| 31:27 | Reserved | Reserved             | -             | -      |
| 26:24 | T_ADV    | T_ADV period status  | T_ADV period  | RO     |
| 23    | Reserved | Reserved             | -             | -      |
| 22:20 | T_TR     | T_TR period status   | T_TR period   | RO     |
| 19:15 | Reserved | Reserved             | -             | -      |
| 14:12 | T_WP     | T_WP period status   | T_WP period   | RO     |
| 11    | Reserved | Reserved             | -             | -      |
| 10:8  | T_CEOE   | T_CEOE period status | T_CEOE period | RO     |
| 7:4   | T_WC     | T_WC period status   | T_WC period   | RO     |
| 3:0   | T_RC     | T_RC period status   | T_RC period   | RO     |

## 40.4.4 DMC-SDRAM Registers

### 40.4.4.1 Status Register (DMC\_STSR)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |      |   |        |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|------|---|--------|---|
| 0x0000 | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3    | 2 | 1      | 0 |
| Reset  | 0x00000700   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |      |   |        |   |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   | MEMW |   | STATUS |   |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   | RO   |   | RO     |   |

| Bit  | Name     | Function           | Description                                                                                                                                                                                                                                                                                                                           | Access |
|------|----------|--------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:4 | Reserved | Reserved           | -                                                                                                                                                                                                                                                                                                                                     | -      |
| 3:2  | MEMW     | DMC MEM bit width  | <ul style="list-style-type: none"> <li>0b00: 16 bits</li> <li>0b01: 32 bits</li> <li>Other: Reserved</li> </ul> <div>  <b>Note</b><br/>           The read value of this bit is the same as the setting value of DMC_BACR.DMCMW[1:0].         </div> | RO     |
| 1:0  | STATUS   | DMC current status | <ul style="list-style-type: none"> <li>0b00: Config</li> <li>0b01: Ready</li> <li>0b10: Paused</li> <li>0b11: Low Power</li> </ul>                                                                                                                                                                                                    | RO     |

## 40.4.4.2 Command Register (DMC\_CMDR)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |         |    |     |    |       |    |          |    |        |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|---------|----|-----|----|-------|----|----------|----|--------|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0x0008 | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21      | 20 | 19  | 18 | 17    | 16 | 15       | 14 | 13     | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0xFFFFFFFF   |    |    |    |    |    |    |    |    |    |         |    |     |    |       |    |          |    |        |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    | CMDCHIP |    | CMD |    | CMDBA |    | Reserved |    | CMDADD |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access |              |    |    |    |    |    |    |    |    |    | WO      |    | WO  |    | WO    |    |          |    | WO     |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit   | Name     | Function               | Description                                                                                                                                                                         | Access |
|-------|----------|------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:22 | Reserved | Reserved               | -                                                                                                                                                                                   | -      |
| 21:20 | CMDCHIP  | CS object setting      | <ul style="list-style-type: none"> <li>0b00: Chip 0</li> <li>0b01: Chip 1</li> <li>0b10: Chip 2</li> <li>0b11: Chip 3</li> </ul>                                                    | WO     |
| 19:18 | CMD      | Command input          | <ul style="list-style-type: none"> <li>0b00: PrechargeAll</li> <li>0b01: AutoRefresh</li> <li>0b10: MdRegConfig</li> <li>0b11: NOP</li> </ul>                                       | WO     |
| 17:16 | CMDBA    | BANK object setting    | Bank object specified by CMD (MdRegConfig) command <ul style="list-style-type: none"> <li>0b00: Bank 0</li> <li>0b01: Bank 1</li> <li>0b10: Bank 2</li> <li>0b11: Bank 3</li> </ul> | WO     |
| 15:14 | Reserved | Reserved               | -                                                                                                                                                                                   | -      |
| 13:0  | CMDADD   | Address object setting | The Address object specified in the CMD (MdReg-Config) command, DMC_ADD[13:0] pin outputs CMDADD[13:0]                                                                              | WO     |

## 40.4.4.3 Status Control Register (DMC\_STCR)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |       |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|-------|---|---|---|
| 0x0004 | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3     | 2 | 1 | 0 |
| Reset  | 0XXXXXXXXX   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |       |   |   |   |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   | STCTL |   |   |   |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   | WO    |   |   |   |

| Bit  | Name     | Function                | Description                                                                                                                                                                                  | Access |
|------|----------|-------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:3 | Reserved | Reserved                | -                                                                                                                                                                                            | -      |
| 2:0  | STCTL    | State switching control | <ul style="list-style-type: none"> <li>0b000: Go</li> <li>0b001: Sleep</li> <li>0b010: Wakeup</li> <li>0b011: Pause</li> <li>0b100: Configure</li> <li>Others: Setting prohibited</li> </ul> | WO     |

## 40.4.4.4 Refresh Time Register (DMC\_RFTR)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |        |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|--------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0x0010 | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15     | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000A60   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |        |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | REFPRD |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RW     |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit   | Name     | Function       | Description                  | Access |
|-------|----------|----------------|------------------------------|--------|
| 31:15 | Reserved | Reserved       | -                            | -      |
| 14:0  | REFPRD   | Refresh period | Refresh period value setting | RW     |

## 40.4.4.5 Basic Control Register (DMC\_BACR)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |       |          |    |    |    |    |   |   |   |   |   |   |   |   |   |       |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-------|----------|----|----|----|----|---|---|---|---|---|---|---|---|---|-------|
| 0x0300 | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15    | 14       | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0     |
| Reset  | 0x00000300   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |       |          |    |    |    |    |   |   |   |   |   |   |   |   |   |       |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | CKSEL | Reserved |    |    |    |    |   |   |   |   |   |   |   |   |   | DMCMW |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RW    |          |    |    |    |    |   |   |   |   |   |   |   |   |   | RW    |

| Bit   | Name     | Function                 | Description                                                                                                                                                                                                                                                    | Access |
|-------|----------|--------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:16 | Reserved | Reserved                 | -                                                                                                                                                                                                                                                              | -      |
| 15:14 | CKSEL    | Sampling clock selection | <ul style="list-style-type: none"> <li>0b00: Setting prohibited</li> <li>0b01: Setting prohibited</li> <li>0b10: When reading the latched DMC_DATA port data, use the EXMC_CLK port status feed-back as the clock</li> <li>0b11: Setting prohibited</li> </ul> | RW     |
| 13:10 | Reserved | Reserved                 | -                                                                                                                                                                                                                                                              | -      |
| 9:8   | Reserved | Reserved                 | All bits read as 1, Write as 1                                                                                                                                                                                                                                 | -      |
| 7:2   | Reserved | Reserved                 | -                                                                                                                                                                                                                                                              | -      |
| 1:0   | DMCMW    | Memory width selection   | <ul style="list-style-type: none"> <li>0b00: 16-bit width</li> <li>0b01: 32-bit width</li> <li>Others: Setting prohibited</li> </ul>                                                                                                                           | RW     |

## 40.4.4.6 Chip Select Control Register x (DMC\_CSCRx) (x=0-3)

Offset: 0x0200+0x04\*x

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |      |         |    |    |    |    |    |   |   |         |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|------|---------|----|----|----|----|----|---|---|---------|---|---|---|---|---|---|---|
| '      | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16   | 15      | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7       | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x0000FF00   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |      |         |    |    |    |    |    |   |   |         |   |   |   |   |   |   |   |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    | BRCx | ADDMATx |    |    |    |    |    |   |   | ADDMSKx |   |   |   |   |   |   |   |
|        |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |      |         |    |    |    |    |    |   |   |         |   |   |   |   |   |   |   |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RW   | RW      |    |    |    |    |    |   |   | RW      |   |   |   |   |   |   |   |
|        |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |      |         |    |    |    |    |    |   |   |         |   |   |   |   |   |   |   |

| Bit   | Name     | Function                | Description                                                                                                            | Access |
|-------|----------|-------------------------|------------------------------------------------------------------------------------------------------------------------|--------|
| 31:17 | Reserved | Reserved                | -                                                                                                                      | -      |
| 16    | BRCx     | Address decoding method | <ul style="list-style-type: none"> <li>0b0: Row, Bank, Column method</li> <li>0b1: Bank, Row, Column method</li> </ul> | RW     |
| 15:8  | ADDMATx  | Address matching        | Address to match                                                                                                       | RW     |
| 7:0   | ADDMSKx  | Address masking         | Corresponding address to be masked                                                                                     | RW     |

Take ADDMAT0 and ADDMSK0 as examples to explain the corresponding method of DMC address space. The settings of other CS spaces are similar to CS0 settings.

Table 40-55 CS0 Setting and Access Address Correspondence Table

| ADDMAT0[7:0] | ADDMSK0[7:0]            | Corresponding External Address | CS0 Space Size |
|--------------|-------------------------|--------------------------------|----------------|
| 0x86         | 0xFF                    | 0x86000000-0x86FFFFFF          | 16MB           |
|              | 0xFE                    | 0x86000000-0x87FFFFFF          | 32MB           |
|              | Other settings disabled | -                              | -              |
| 0x84         | 0xFF                    | 0x84000000-0x84FFFFFF          | 16MB           |
|              | 0xFE                    | 0x84000000-0x85FFFFFF          | 32MB           |
|              | 0xFC                    | 0x84000000-0x87FFFFFF          | 64MB           |
|              | Other settings disabled | -                              | -              |
| 0x82         | 0xFF                    | 0x82000000-0x82FFFFFF          | 16MB           |
|              | 0xFE                    | 0x82000000-0x83FFFFFF          | 32MB           |
|              | Other settings disabled | -                              | -              |



| ADDMAT0[7:0] | ADDMSK0[7:0]            | Corresponding External Address | CS0 Space Size |
|--------------|-------------------------|--------------------------------|----------------|
| 0x80         | 0xFF                    | 0x80000000-0x80FFFFFF          | 16MB           |
|              | 0xFE                    | 0x80000000-0x81FFFFFF          | 32MB           |
|              | 0xFC                    | 0x80000000-0x83FFFFFF          | 64MB           |
|              | 0xF8                    | 0x80000000-0x87FFFFFF          | 128MB          |
|              | Other settings disabled | -                              | -              |

## 40.4.4.7 CHIP Configuration Register (DMC\_CPCR)

| Offset | Bit Position |    |    |    |       |    |    |    |          |    |    |    |       |    |    |    |           |    |    |    |        |    |   |   |        |   |   |   |      |   |   |   |       |  |  |  |          |  |  |  |       |  |  |  |
|--------|--------------|----|----|----|-------|----|----|----|----------|----|----|----|-------|----|----|----|-----------|----|----|----|--------|----|---|---|--------|---|---|---|------|---|---|---|-------|--|--|--|----------|--|--|--|-------|--|--|--|
| 0x000C | 31           | 30 | 29 | 28 | 27    | 26 | 25 | 24 | 23       | 22 | 21 | 20 | 19    | 18 | 17 | 16 | 15        | 14 | 13 | 12 | 11     | 10 | 9 | 8 | 7      | 6 | 5 | 4 | 3    | 2 | 1 | 0 |       |  |  |  |          |  |  |  |       |  |  |  |
| Reset  | 0x00020040   |    |    |    |       |    |    |    |          |    |    |    |       |    |    |    |           |    |    |    |        |    |   |   |        |   |   |   |      |   |   |   |       |  |  |  |          |  |  |  |       |  |  |  |
| Name   | Reserved     |    |    |    | ACTCP |    |    |    | Reserved |    |    |    | BURST |    |    |    | CKEDISPRD |    |    |    | CKSTOP |    |   |   | CKEDIS |   |   |   | APBS |   |   |   | ROWBS |  |  |  | Reserved |  |  |  | COLBS |  |  |  |
| Access |              |    |    |    | RW    |    |    |    |          |    |    |    | RW    |    |    |    | RW        |    |    |    | RW     |    |   |   | RW     |   |   |   | RW   |   |   |   |       |  |  |  | RW       |  |  |  |       |  |  |  |

| Bit   | Name      | Function                          | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | Access |
|-------|-----------|-----------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:26 | Reserved  | Reserved                          | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | -      |
| 25:24 | ACTCP     | Number of automatic refresh chips | <ul style="list-style-type: none"> <li>0b00: Chip 0 automatic refresh enabled</li> <li>0b01: Chip 0-1 automatic refresh enabled</li> <li>0b10: Chip 0-2 automatic refresh enabled</li> <li>0b11: Chip 0-3 automatic refresh enabled</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | RW     |
| 23:19 | Reserved  | Reserved                          | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | -      |
| 18:16 | BURST     | Burst transfer length             | <ul style="list-style-type: none"> <li>0b000: 1 transfer</li> <li>0b001: 2 consecutive transfers</li> <li>0b010: 4 consecutive transfers</li> <li>0b011: 8 consecutive transfers</li> <li>0b100: 16 consecutive transfers</li> <li>Others: Setting prohibited</li> </ul> <div>  <b>Note</b> <ul style="list-style-type: none"> <li>The setting of this bit must be consistent with the corresponding bit setting of the Mode Register of the external Memory.</li> <li>When using SMC and DMC at the same time, if BURST is configured as a non-zero value, the AHB bus read and write data bit width of the DMC needs to be less than or equal to the bit width of the external storage device.</li> </ul> </div> | RW     |
| 15:10 | CKEDISPRD | Clock output off period           | Clock output off waiting period setting                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | RW     |
| 9     | CKSTOP    | Clock stop setting                | <ul style="list-style-type: none"> <li>0b0: MEM clock normal output</li> <li>0b1: MEM clock stops when no operation</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | RW     |
| 8     | CKEDIS    | CKE output setting                | <ul style="list-style-type: none"> <li>0b0: CKE output active (H)</li> <li>0b1: When the command FIFO is empty, after the period set by CKEDISPRD, CKE output becomes inactive (L)</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | RW     |
| 7     | APBS      | Auto precharge bit                | <ul style="list-style-type: none"> <li>0b0: 10th bit of address line</li> <li>0b1: 8th bit of address line</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | RW     |

| Bit | Name            | Function             | Description                                                                                                                                                                                                                                                                                                                | Access |
|-----|-----------------|----------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 6:4 | ROWBS           | Row address width    | <ul style="list-style-type: none"> <li>0b000: Row address width is 11 bits</li> <li>0b001: Row address width is 12 bits</li> <li>0b010: Row address width is 13 bits</li> <li>Others: Setting prohibited</li> </ul>                                                                                                        | RW     |
| 3   | <i>Reserved</i> | <i>Reserved</i>      | -                                                                                                                                                                                                                                                                                                                          | -      |
| 2:0 | COLBS           | Column address width | <ul style="list-style-type: none"> <li>0b000: Column address width is 8 bits</li> <li>0b001: Column address width is 9 bits</li> <li>0b010: Column address width is 10 bits</li> <li>0b011: Column address width is 11 bits</li> <li>0b100: Column address width is 12 bits</li> <li>Others: Setting prohibited</li> </ul> | RW     |

#### 40.4.4.8 Timing Configuration Register (DMC\_TMCR\_T\_CASL)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |        |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|--------|
| 0x0014 | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16     |
| Reset  | 0x00000003   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |        |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    | T_CASL |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RW     |

| Bit  | Name            | Function              | Description                       | Access |
|------|-----------------|-----------------------|-----------------------------------|--------|
| 31:3 | <i>Reserved</i> | <i>Reserved</i>       | -                                 | -      |
| 2:0  | T_CASL          | T_CASL period setting | T_CASL setting (CAS latency time) | RW     |

#### 40.4.4.9 Timing Configuration Register (DMC\_TMCR\_T\_DQSS)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |        |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|--------|
| 0x0018 | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16     |
| Reset  | 0x00000001   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |        |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    | T_DQSS |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RW     |

| Bit  | Name            | Function              | Description    | Access |
|------|-----------------|-----------------------|----------------|--------|
| 31:2 | <i>Reserved</i> | <i>Reserved</i>       | -              | -      |
| 1:0  | T_DQSS          | T_DQSS period setting | T_DQSS setting | RW     |

## 40.4.4.10 Timing Configuration Register (DMC\_TMCR\_T\_MRD)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |       |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|-------|---|---|---|---|---|---|---|
| 0x001C | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7     | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000002   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |       |   |   |   |   |   |   |   |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   | T_MRD |   |   |   |   |   |   |   |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   | RW    |   |   |   |   |   |   |   |

| Bit  | Name     | Function             | Description                                | Access |
|------|----------|----------------------|--------------------------------------------|--------|
| 31:7 | Reserved | Reserved             | -                                          | -      |
| 6:0  | T_MRD    | T_MRD period setting | T_MRD setting (mode register command time) | RW     |

## 40.4.4.11 Timing Configuration Register (DMC\_TMCR\_T\_RAS)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |       |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|-------|
| 0x0020 | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0     |
| Reset  | 0x00000007   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |       |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   | T_RAS |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   | RW    |

| Bit  | Name     | Function             | Description                                 | Access |
|------|----------|----------------------|---------------------------------------------|--------|
| 31:4 | Reserved | Reserved             | -                                           | -      |
| 3:0  | T_RAS    | T_RAS period setting | T_RAS setting (RAS to precharge delay time) | RW     |

40.4.4.12 Timing Configuration Register (DMC\_TMCR\_T\_RC)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |      |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|------|---|---|---|
| 0x0024 | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3    | 2 | 1 | 0 |
| Reset  | 0x0000000B   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |      |   |   |   |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   | T_RC |   |   |   |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   | RW   |   |   |   |

| Bit  | Name     | Function            | Description                                              | Access |
|------|----------|---------------------|----------------------------------------------------------|--------|
| 31:4 | Reserved | Reserved            | -                                                        | -      |
| 3:0  | T_RC     | T_RC period setting | T_RC Setting (Active bank x to Active bank x delay time) | RW     |

## 40.4.4.13 Timing Configuration Register (DMC\_TMCR\_T\_RCD)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |         |   |          |         |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---------|---|----------|---------|---|---|---|---|
| 0x0028 | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7       | 6 | 5        | 4       | 3 | 2 | 1 | 0 |
| Reset  | 0x00000035   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |         |   |          |         |   |   |   |   |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   | T_RCD_P |   | Reserved | T_RCD_B |   |   |   |   |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   | RW      |   |          | RW      |   |   |   |   |

| Bit  | Name     | Function                        | Description                                              | Access |
|------|----------|---------------------------------|----------------------------------------------------------|--------|
| 31:7 | Reserved | Reserved                        | -                                                        | -      |
| 6:4  | T_RCD_P  | Additional T_RCD period setting | Additional T_RCD setting                                 | RW     |
| 3    | Reserved | Reserved                        | -                                                        | -      |
| 2:0  | T_RCD_B  | Basic T_RCD period setting      | Basic T_RCD setting, needs to be set to a non-zero value | RW     |

T\_RCD (RAS to CAS minimum delay time) is calculated as follows:

- When CKSTOP=0:
  - ▶ When T\_RCD\_B ≥ T\_RCD\_P+3, the actual T\_RCD width is T\_RCD\_B DMC module clock periods;
  - ▶ When T\_RCD\_B < T\_RCD\_P+3, the actual T\_RCD width is (T\_RCD\_P+3) DMC module clock periods.
- When CKSTOP=1:
  - ▶ When T\_RCD\_B ≥ T\_RCD\_P+2, the actual T\_RCD width is T\_RCD\_B DMC module clock periods;
  - ▶ When T\_RCD\_B < T\_RCD\_P+2, the actual T\_RCD width is (T\_RCD\_P+3) DMC module clock periods.

## 40.4.4.14 Timing Configuration Register (DMC\_TMCR\_T\_RFC)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |         |    |    |    |          |    |   |   |         |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---------|----|----|----|----------|----|---|---|---------|---|---|---|---|---|---|---|
| 0x002C | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15      | 14 | 13 | 12 | 11       | 10 | 9 | 8 | 7       | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00001012   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |         |    |    |    |          |    |   |   |         |   |   |   |   |   |   |   |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | T_RFC_P |    |    |    | Reserved |    |   |   | T_RFC_B |   |   |   |   |   |   |   |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RW      |    |    |    |          |    |   |   | RW      |   |   |   |   |   |   |   |

| Bit   | Name     | Function                         | Description               | Access |
|-------|----------|----------------------------------|---------------------------|--------|
| 31:13 | Reserved | Reserved                         | -                         | -      |
| 12:8  | T_RFC_P  | Additional T_RFC period settings | Additional T_RFC settings | RW     |
| 7:5   | Reserved | Reserved                         | -                         | -      |
| 4:0   | T_RFC_B  | Basic T_RFC period settings      | Basic T_RFC settings      | RW     |

T\_RFC (autorefresh command time) is calculated as follows:

- When  $T\_RFC\_B \geq T\_RFC\_P + 3$ , the actual T\_RFC width is T\_RFC\_B DMC module clock periods;
- When  $T\_RFC\_B < T\_RFC\_P + 3$ , the actual T\_RFC width is (T\_RFC\_P + 3) dMC module clock periods.

## 40.4.4.15 Timing Configuration Register (DMC\_TMCR\_T\_RP)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |        |   |          |        |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|--------|---|----------|--------|---|---|---|---|
| 0x0030 | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7      | 6 | 5        | 4      | 3 | 2 | 1 | 0 |
| Reset  | 0x00000035   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |        |   |          |        |   |   |   |   |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   | T_RP_P |   | Reserved | T_RP_B |   |   |   |   |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   | RW     |   |          | RW     |   |   |   |   |

| Bit  | Name     | Function                        | Description                                             | Access |
|------|----------|---------------------------------|---------------------------------------------------------|--------|
| 31:7 | Reserved | Reserved                        | -                                                       | -      |
| 6:4  | T_RP_P   | Additional T_RP period settings | Additional T_RP settings                                | RW     |
| 3    | Reserved | Reserved                        | -                                                       | -      |
| 2:0  | T_RP_B   | Basic T_RP period setting       | Basic T_RP setting, needs to be set to a non-zero value | RW     |

The calculation method of T\_RP is as follows:

- When CKSTOP=0:
  - ▶ When  $T\_RP\_B \geq T\_RP\_P + 3$ , the actual T\_RP width is T\_RP\_B DMC module clock periods;
  - ▶ When  $T\_RP\_B < T\_RP\_P + 3$ , the actual T\_RP width is (T\_RP\_P+3) DMC module clock periods.
- When CKSTOP=1:
  - ▶ When  $T\_RP\_B \geq T\_RP\_P + 2$ , the actual T\_RP width is T\_RP\_B DMC module clock periods;
  - ▶ When  $T\_RP\_B < T\_RP\_P + 2$ , the actual T\_RP width is (T\_RP\_P+3) DMC module clock periods.

## 40.4.4.16 Timing Configuration Register (DMC\_TMCR\_T\_RRD)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |       |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|-------|---|---|---|
| 0x0034 | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3     | 2 | 1 | 0 |
| Reset  | 0x00000002   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |       |   |   |   |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   | T_RRD |   |   |   |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   | RW    |   |   |   |

| Bit  | Name     | Function             | Description                                               | Access |
|------|----------|----------------------|-----------------------------------------------------------|--------|
| 31:4 | Reserved | Reserved             | -                                                         | -      |
| 3:0  | T_RRD    | T_RRD period setting | T_RRD setting (Active bank x to Active bank y delay time) | RW     |



## 40.4.4.17 Timing Configuration Register (DMC\_TMCR\_T\_WR)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |      |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|------|
| 0x0038 | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0    |
| Reset  | 0x00000003   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |      |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   | T_WR |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   | RW   |

| Bit  | Name     | Function            | Description                                  | Access |
|------|----------|---------------------|----------------------------------------------|--------|
| 31:3 | Reserved | Reserved            | -                                            | -      |
| 2:0  | T_WR     | T_WR period setting | T_WR setting (write to precharge delay time) | RW     |

## 40.4.4.18 Timing Configuration Register (DMC\_TMCR\_T\_WTR)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |       |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|-------|
| 0x003C | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0     |
| Reset  | 0x00000002   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |       |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   | T_WTR |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   | RW    |

| Bit  | Name     | Function             | Description                              | Access |
|------|----------|----------------------|------------------------------------------|--------|
| 31:3 | Reserved | Reserved             | -                                        | -      |
| 2:0  | T_WTR    | T_WTR period setting | T_WTR setting (write to read delay time) | RW     |

40.4.4.19 Timing Configuration Register (DMC\_TMCR\_T\_XP)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |      |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|------|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0x0040 | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18   | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000001   |    |    |    |    |    |    |    |    |    |    |    |    |      |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    | T_XP |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |      |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit  | Name     | Function            | Description                                 | Access |
|------|----------|---------------------|---------------------------------------------|--------|
| 31:8 | Reserved | Reserved            | -                                           | -      |
| 7:0  | T_XP     | T_XP period setting | T_XP setting (exit Power-down command time) | RW     |

## 40.4.4.20 Timing Configuration Register (DMC\_TMCR\_T\_XSR)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |       |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|-------|---|---|---|---|---|---|---|
| 0x0044 | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7     | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x0000000A   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |       |   |   |   |   |   |   |   |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   | T_XSR |   |   |   |   |   |   |   |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   | RW    |   |   |   |   |   |   |   |

| Bit  | Name     | Function             | Description                                    | Access |
|------|----------|----------------------|------------------------------------------------|--------|
| 31:8 | Reserved | Reserved             | -                                              | -      |
| 7:0  | T_XSR    | T_XSR period setting | T_XSR setting (exit self-refresh command time) | RW     |

## 40.4.4.21 Timing Configuration Register (DMC\_TMCR\_T\_ESR)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |       |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|-------|---|---|---|---|---|---|---|
| 0x0048 | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7     | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000014   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |       |   |   |   |   |   |   |   |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   | T_ESR |   |   |   |   |   |   |   |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   | RW    |   |   |   |   |   |   |   |

| Bit  | Name     | Function             | Description                               | Access |
|------|----------|----------------------|-------------------------------------------|--------|
| 31:8 | Reserved | Reserved             | -                                         | -      |
| 7:0  | T_ESR    | T_ESR period setting | T_ESR setting (self-refresh command time) | RW     |

## 40.4.5 NFC-NAND Flash Register

### 40.4.5.1 Status Register (NFC\_STSR)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |         |   |           |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---------|---|-----------|---|---|---|---|---|---|
| 0x28   | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8       | 7 | 6         | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |         |   |           |   |   |   |   |   |   |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |         |   | CHIP_BUSY |   |   |   |   |   |   |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   | RO PECC |   |           |   |   |   |   |   |   |

| Bit  | Name      | Function      | Description                                                                                                                                 | Access |
|------|-----------|---------------|---------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:9 | Reserved  | Reserved      | -                                                                                                                                           | -      |
| 8    | PECC      | ECC status    | <ul style="list-style-type: none"> <li>0b0: NFC does not perform ECC calculation</li> <li>0b1: NFC is performing ECC calculation</li> </ul> | RO     |
| 7:0  | CHIP_BUSY | Device status | Corresponding to BANK7-0 device: <ul style="list-style-type: none"> <li>0b0: Device is busy</li> <li>0b1: Device ready</li> </ul>           | RO     |

### 40.4.5.2 Control Register (NFC\_STCR)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |    |       |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|----|-------|
| 0x08   | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |    |       |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |    |       |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   | RW | OPENP |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   | RW |       |

| Bit  | Name     | Function          | Description                                                                                                                                                                                       | Access |
|------|----------|-------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:1 | Reserved | Reserved          | -                                                                                                                                                                                                 | -      |
| 0    | OPENP    | Open Page Enabled | <ul style="list-style-type: none"> <li>0b0: Do not allow NFC to automatically send a read command after reset</li> <li>0b1: Allow NFC to automatically send a read command after reset</li> </ul> | RW     |

40.4.5.3 Data Register (NFC\_DATR)

| Offset        | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|---------------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0x0000-0x7FFC | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset         | 0xFFFFFFFF   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name          | DATA         |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access        | RW           |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit  | Name | Function | Description                                                                                                                                                              | Access |
|------|------|----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:0 | DATA | Data     | The data register is used to read and write data on the Flash device. The address of the read and write data is determined by the address written by the index register. | RW     |

## 40.4.5.4 Command Register (NFC\_CMDR)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |     |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|-----|---|---|---|---|---|---|---|
| 0x8000 | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7   | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |     |   |   |   |   |   |   |   |
| Name   | ARG          |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   | CMD |   |   |   |   |   |   |   |
| Access | RW           |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   | RW  |   |   |   |   |   |   |   |

| Bit  | Name | Function   | Description                                                      | Access |
|------|------|------------|------------------------------------------------------------------|--------|
| 31:8 | ARG  | Parameters | Used to adjust the command, see the table below for details      | RW     |
| 7:0  | CMD  | Command    | The commands sent to the device are detailed in the table below. | RW     |

The setting methods of all commands and parameters are as follows:

Table 40-94 Command parameter list

| Com-mand (CMD) | Parameter (ARG)   | Function                                                                                                                            |
|----------------|-------------------|-------------------------------------------------------------------------------------------------------------------------------------|
| 0x00           | 0x000000          | Read Command                                                                                                                        |
| 0x10           | 0x000000          | Page program confirm                                                                                                                |
| 0x11           | 0x000000          | A confirmation inserted after the first page of a two-plane page program ends                                                       |
| 0x15           | 0x000000          | The second command in the Page cache program                                                                                        |
| 0x23           | 0x000000          | Read the ECC status. After writing this command, NFC will read ECC data from the extra area of NAND Flash and start ECC calculation |
| 0x60           | 0x810000-0x810007 | Block erase command, parameter is used to indicate the operation object BANK0-BANK7                                                 |
| 0x70           | 0x810000-0x810007 | Read status command, parameter is used to indicate the operation object BANK0-BANK7                                                 |
| 0x80           | 0x000000          | Page program command                                                                                                                |
| 0x81           | 0x000000          | Used for two-plane page program to start writing the second page                                                                    |
| 0x90           | 0x000000-0x000007 | Read ID command, parameter is used to indicate the operation object BANK0-BANK7                                                     |
| 0xD0           | 0x810000-0x810007 | Block erase confirm, parameter is used to indicate the operation object BANK0-BANK7                                                 |
| 0xD1           | 0x810000-0x810007 | Block erase interleaved confirm, the parameter is used to indicate the operation object BANK0-BANK7                                 |

| Com-mand (CMD) | Parameter (ARG)   | Function                                                                                                                                                                                                                                                     |
|----------------|-------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0xE0           | 0x000000          | Start reading data. Used to instruct NFC to start reading data. After writing this command, NFC will open a page for data reading. At this time, CHIP_BUSY will be set. After waiting for CHIP_BUSY to reset, you can read the data register to obtain data. |
| 0xFE           | 0x000000          | After the read operation is completed, write this command to instruct NFC to inactivate the chip select signal CE                                                                                                                                            |
| 0xFF           | 0x000000-0x000007 | Reset command, the parameter is used to indicate the operation object BANK0-BANK7                                                                                                                                                                            |
| 0x05           | 0x810000-0x810007 | The first period of the change read column command                                                                                                                                                                                                           |
| 0x06           | 0x810000-0x810007 | The first period of the change read column enhanced command                                                                                                                                                                                                  |
| 0x31           | 0x830000-0x830007 | The first period of the read cache sequential command and the second period of the read cache random command                                                                                                                                                 |
| 0x32           | 0x810000-0x810007 | Copyback read interleaved command                                                                                                                                                                                                                            |
| 0x35           | 0x810000-0x810007 | Copyback read command                                                                                                                                                                                                                                        |
| 0x3F           | 0x830000-0x830007 | Read cache end                                                                                                                                                                                                                                               |
| 0x78           | 0x800000-0x800007 | Read status enhanced command                                                                                                                                                                                                                                 |
| 0x85           | 0x810000-0x810007 | Copyback program interleaved and Copyback program commands                                                                                                                                                                                                   |
| 0x85           | 0x840000-0x840007 | Change write column and Change row address                                                                                                                                                                                                                   |
| 0xE0           | 0x850000-0x850007 | Second period of the Change read column and Change read column enhanced commands                                                                                                                                                                             |
| 0xEC           | 0x830000-0x830007 | Read parameter page command                                                                                                                                                                                                                                  |
| 0xED           | 0x830000-0x830007 | Read Unique ID command                                                                                                                                                                                                                                       |
| 0xEE           | 0x830000-0x830007 | Get Features command                                                                                                                                                                                                                                         |
| 0xEF           | 0x830000-0x830007 | Set Features command                                                                                                                                                                                                                                         |
| 0xFA           | 0x820000-0x820007 | Reset LUN command                                                                                                                                                                                                                                            |
| 0xFC           | 0x820000-0x820007 | Asynchronous reset command                                                                                                                                                                                                                                   |

40.4.5.5 Index Register 0 (NFC\_IDXR0)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0x8004 | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | IDX0         |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access | RW           |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit  | Name | Function | Description                                                                                 | Access |
|------|------|----------|---------------------------------------------------------------------------------------------|--------|
| 31:0 | IDX0 | Index    | The lower 32 bits of the access address, the upper 8 bits are composed of the IDX0 register | RW     |



## 40.4.5.6 Index Register 1 (NFC\_IDXR1)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |      |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|------|---|---|---|---|---|---|---|
| 0x8048 | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7    | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |      |   |   |   |   |   |   |   |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   | IDX1 |   |   |   |   |   |   |   |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   | RW   |   |   |   |   |   |   |   |

| Bit  | Name     | Function | Description                                                                                 | Access |
|------|----------|----------|---------------------------------------------------------------------------------------------|--------|
| 31:8 | Reserved | Reserved | -                                                                                           | -      |
| 7:0  | IDX1     | Index    | The high 8 bits of the access address and the low 32 bits are composed of the IDX0 register | RW     |

IDXR0 and IDX1 together form the 40-bit address when accessing the NAND Flash device. The relationship between the value written to the register and the address of the actual device depends on the capacity of the device. The following table lists the address structure when accessing various capacity devices:

Table 40-99 Index Register Value and MEM Address Correspondence Table

| Capacity | 2K Page |        |       | 4K Page |        |       | 8K Page |        |       |
|----------|---------|--------|-------|---------|--------|-------|---------|--------|-------|
|          | Page    | Column | Bank  | Page    | Column | Bank  | Page    | Column | Bank  |
| 512Mbits | 26:12   | 11:0   | 29:27 | 26:13   | 12:0   | 29:27 | 26:14   | 13:0   | 29:27 |
| 1Gbits   | 27:12   | 11:0   | 30:28 | 27:13   | 12:0   | 30:28 | 27:14   | 13:0   | 30:28 |
| 2Gbits   | 28:12   | 11:0   | 31:29 | 28:13   | 12:0   | 31:29 | 28:14   | 13:0   | 31:29 |
| 4Gbits   | 29:12   | 11:0   | 32:30 | 29:13   | 12:0   | 32:30 | 29:14   | 13:0   | 32:30 |
| 8Gbits   | 30:12   | 11:0   | 33:31 | 30:13   | 12:0   | 33:31 | 30:14   | 13:0   | 33:31 |
| 16Gbits  | 31:12   | 11:0   | 34:32 | 31:13   | 12:0   | 34:32 | 31:14   | 13:0   | 34:32 |
| 32Gbits  | 32:12   | 11:0   | 35:33 | 32:13   | 12:0   | 35:33 | 32:14   | 13:0   | 35:33 |
| 64Gbits  | 33:12   | 11:0   | 36:34 | 33:13   | 12:0   | 36:34 | 33:14   | 13:0   | 36:34 |

## 40.4.5.7 Basic Configuration Register (NFC\_BACR)

| Offset | Bit Position |    |    |    |    |    |    |    |     |    |    |    |    |    |    |    |          |     |      |    |      |      |          |       |      |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|-----|----|----|----|----|----|----|----|----------|-----|------|----|------|------|----------|-------|------|---|---|---|---|---|---|---|
| 0x8054 | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23  | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15       | 14  | 13   | 12 | 11   | 10   | 9        | 8     | 7    | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00002187   |    |    |    |    |    |    |    |     |    |    |    |    |    |    |    |          |     |      |    |      |      |          |       |      |   |   |   |   |   |   |   |
| Name   | Reserved     |    |    |    |    |    |    |    | SCS |    |    |    |    |    |    |    | Reserved | RAC | ECCM | WP | PAGE | BANK | Reserved | 16BIT | SIZE |   |   |   |   |   |   |   |
| Access |              |    |    |    |    |    |    |    | RW  |    |    |    |    |    |    |    |          | RW  | RW   | RW | RW   | RW   |          | RW    | RW   |   |   |   |   |   |   |   |

| Bit   | Name     | Function             | Description                                                                                                                                                                                                                           | Access |
|-------|----------|----------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:24 | Reserved | Reserved             | -                                                                                                                                                                                                                                     | -      |
| 23:16 | SCS      | Spare column size    | Set the spare column size in words.                                                                                                                                                                                                   | RW     |
| 15:14 | Reserved | Reserved             | -                                                                                                                                                                                                                                     | -      |
| 13    | RAC      | Row address period   | <ul style="list-style-type: none"> <li>0b0: 2 row address periods</li> <li>0b1: 3 row address periods</li> </ul>                                                                                                                      | RW     |
| 12:11 | ECCM     | ECC Mode             | <ul style="list-style-type: none"> <li>0b00: 1-bit ECC mode</li> <li>0b01: 4-bit ECC mode</li> <li>Others: Setting prohibited</li> </ul>                                                                                              | RW     |
| 10    | WP       | Write Protection     | <ul style="list-style-type: none"> <li>0b0: Set the device to write-protected state</li> <li>0b1: Set the device to unprotected state</li> </ul>                                                                                      | RW     |
| 9:8   | PAGE     | Page size            | <ul style="list-style-type: none"> <li>0b00: Setting prohibited</li> <li>0b01: 2K</li> <li>0b10: 4K</li> <li>0b11: 8K</li> </ul>                                                                                                      | RW     |
| 7:6   | BANK     | Number of banks (CE) | <ul style="list-style-type: none"> <li>0b00: 1bank</li> <li>0b01: 2bank</li> <li>0b10: 4bank</li> <li>0b11: 8bank</li> </ul>                                                                                                          | RW     |
| 5:4   | Reserved | Reserved             | -                                                                                                                                                                                                                                     | -      |
| 3     | 16BIT    | Data bit width       | <ul style="list-style-type: none"> <li>0b0: 8-bit</li> <li>0b1: 16-bit</li> </ul>                                                                                                                                                     | RW     |
| 2:0   | SIZE     | Device capacity      | <ul style="list-style-type: none"> <li>0b000: 16Gbits</li> <li>0b001: 32Gbits</li> <li>0b010: 64Gbits</li> <li>0b011: 512Mbits</li> <li>0b100: 1Gbits</li> <li>0b101: 2Gbits</li> <li>0b110: 4Gbits</li> <li>0b111: 8Gbits</li> </ul> | RW     |

## 40.4.5.8 Interrupt Enable Register (NFC\_IENR)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |      |    |    |    |    |    |   |   |        |        |          |       |          |        |         |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|------|----|----|----|----|----|---|---|--------|--------|----------|-------|----------|--------|---------|---|
| 0x8030 | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15   | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7      | 6      | 5        | 4     | 3        | 2      | 1       | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |      |    |    |    |    |    |   |   |        |        |          |       |          |        |         |   |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RBEN |    |    |    |    |    |   |   | ECCDIS | ECCEEN | Reserved | ECCEN | Reserved | ECCEEN | ECCEUEN |   |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RW   |    |    |    |    |    |   |   | RW     | RW     |          | RW    |          | RW     | RW      |   |

| Bit   | Name     | Function                                     | Description                                                                                                                                                                                                                                             | Access |
|-------|----------|----------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:16 | Reserved | Reserved                                     | -                                                                                                                                                                                                                                                       | -      |
| 15:8  | RBEN     | Ready interrupt enabled                      | Corresponding to BANK7-0 device:<br><ul style="list-style-type: none"> <li>0b0: Do not allow interrupts when the access device is ready (RB signal is set)</li> <li>0b1: Allow interrupts when the access device is ready (RB signal is set)</li> </ul> | RW     |
| 7     | ECCDIS   | ECC disabled                                 | <ul style="list-style-type: none"> <li>0b0: Enable ECC function</li> <li>0b1: Disable ECC function</li> </ul>                                                                                                                                           | RW     |
| 6     | ECCEEN   | ECC error interrupt enabled                  | <ul style="list-style-type: none"> <li>0b0: No interrupt is generated when an ECC error occurs</li> <li>0b1: Interrupt is generated when an ECC error occurs</li> </ul>                                                                                 | RW     |
| 5     | Reserved | Reserved                                     | -                                                                                                                                                                                                                                                       | -      |
| 4     | ECCEN    | ECC calculation completion interrupt enabled | <ul style="list-style-type: none"> <li>0b0: No interrupt is generated when ECC calculation is completed</li> <li>0b1: Interrupt is generated when ECC calculation is completed</li> </ul>                                                               | RW     |
| 3:2   | Reserved | Reserved                                     | -                                                                                                                                                                                                                                                       | -      |
| 1     | ECCEEN   | Correctable ECC error interrupt enabled      | <ul style="list-style-type: none"> <li>0b0: No interrupt is generated when a correctable ECC error occurs in 1bitECC mode</li> <li>0b1: Interrupt is generated when a correctable ECC error occurs in 1bitECC mode</li> </ul>                           | RW     |
| 0     | ECCEUEN  | Uncorrectable ECC error interrupt enabled    | <ul style="list-style-type: none"> <li>0b0: No interrupt is generated when an uncorrectable ECC error occurs in 1bitECC mode</li> <li>0b1: Interrupt is generated when an uncorrectable ECC error occurs in 1bitECC mode</li> </ul>                     | RW     |

## 40.4.5.9 Interrupt Status Register (NFC\_ISTR)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |      |    |    |    |    |    |   |   |          |        |          |        |          |        |          |        |  |    |  |    |  |    |  |    |  |    |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|------|----|----|----|----|----|---|---|----------|--------|----------|--------|----------|--------|----------|--------|--|----|--|----|--|----|--|----|--|----|
| 0x8034 | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15   | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7        | 6      | 5        | 4      | 3        | 2      | 1        | 0      |  |    |  |    |  |    |  |    |  |    |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |      |    |    |    |    |    |   |   |          |        |          |        |          |        |          |        |  |    |  |    |  |    |  |    |  |    |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RBST |    |    |    |    |    |   |   | Reserved | ECCEST | Reserved | ECCEST | Reserved | ECCEST | Reserved | ECCEST |  |    |  |    |  |    |  |    |  |    |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RW   |    |    |    |    |    |   |   |          | RW     |          | RW     |          | RW     |          | RW     |  | RW |  | RW |  | RW |  | RW |  | RW |

| Bit   | Name     | Function                          | Description                                                                                                                                                                                                                                    | Access |
|-------|----------|-----------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:16 | Reserved | Reserved                          | -                                                                                                                                                                                                                                              | -      |
| 15:8  | RBST     | Ready status                      | Corresponding to BANK7-0 device:<br>Write 0 to clear this bit, write 1 has no effect<br>Read:<br><ul style="list-style-type: none"> <li>0b0: Access device not ready</li> <li>0b1: Access to the device is ready (RB signal is set)</li> </ul> | RW     |
| 7     | Reserved | Reserved                          | -                                                                                                                                                                                                                                              | -      |
| 6     | ECCEST   | ECC Error Status                  | Write 0 to clear this bit, write 1 has no effect<br>Read:<br><ul style="list-style-type: none"> <li>0b0: 4bitECC error has not occurred</li> <li>0b1: 4bitECC error has occurred</li> </ul>                                                    | RW     |
| 5     | Reserved | Reserved                          | -                                                                                                                                                                                                                                              | -      |
| 4     | ECCEST   | ECC calculation completion status | Write 0 to clear this bit, write 1 has no effect<br>Read:<br><ul style="list-style-type: none"> <li>0b0: ECC calculation is not completed</li> <li>0b1: ECC calculation is completed</li> </ul>                                                | RW     |
| 3:2   | Reserved | Reserved                          | -                                                                                                                                                                                                                                              | -      |
| 1     | ECCEST   | Correctable ECC error status      | Write 0 to clear this bit, write 1 has no effect<br>Read:<br><ul style="list-style-type: none"> <li>0b0: No correctable ECC error occurred in 1bitECC mode</li> <li>0b1: A correctable ECC error occurred in 1bitECC mode</li> </ul>           | RW     |
| 0     | ECCEUST  | Uncorrectable ECC error status    | Write 0 to clear this bit, write 1 has no effect<br>Read:<br><ul style="list-style-type: none"> <li>0b0: No uncorrectable ECC error occurred in 1bitECC mode</li> <li>0b1: An uncorrectable ECC error has occurred in 1bitECC mode</li> </ul>  | RW     |

## 40.4.5.10 Interrupt Result Register (NFC\_IRSR)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |      |    |    |    |    |    |   |   |          |        |          |        |          |         |         |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|------|----|----|----|----|----|---|---|----------|--------|----------|--------|----------|---------|---------|---|
| 0x8038 | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15   | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7        | 6      | 5        | 4      | 3        | 2       | 1       | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |      |    |    |    |    |    |   |   |          |        |          |        |          |         |         |   |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RBRS |    |    |    |    |    |   |   | Reserved | ECCERS | Reserved | ECCCRS | Reserved | ECCECRS | ECCEURS |   |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RO   |    |    |    |    |    |   |   |          | RO     |          | RO     |          | RO      | RO      |   |

| Bit   | Name     | Function                            | Description                                                                                                                                                                                                                                                                         | Access |
|-------|----------|-------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:16 | Reserved | Reserved                            | -                                                                                                                                                                                                                                                                                   | -      |
| 15:8  | RBRS     | Ready interrupt                     | Corresponding to BANK7-0 device:<br>This bit is the logical AND result of NFC_ISTR.RBST and NFC_IENR.RBEN<br><ul style="list-style-type: none"> <li>0b0: Access device ready interrupt is not generated</li> <li>0b1: Access device ready interrupt is generated</li> </ul>         | RO     |
| 7     | Reserved | Reserved                            | -                                                                                                                                                                                                                                                                                   | -      |
| 6     | ECCERS   | ECC Error interrupt                 | This bit is the logical result of NFC_ISTR.ECCEST and NFC_IENR.ECCEEN.<br><ul style="list-style-type: none"> <li>0b0: 4bitECC error interrupt not generated</li> <li>0b1: 4bitECC error interrupt generated</li> </ul>                                                              | RO     |
| 5     | Reserved | Reserved                            | -                                                                                                                                                                                                                                                                                   | -      |
| 4     | ECCCRS   | ECC calculation completed interrupt | This bit is the logical result of NFC_ISTR.ECCCST and NFC_IENR.ECCCEEN<br><ul style="list-style-type: none"> <li>0b0: ECC calculation completed interrupt not generated</li> <li>0b1: ECC calculation completed interrupt generated</li> </ul>                                      | RO     |
| 3:2   | Reserved | Reserved                            | -                                                                                                                                                                                                                                                                                   | -      |
| 1     | ECCECRS  | Correctable ECC error interrupt     | This bit is the AND logic result of NFC_ISTR.ECCECST and NFC_IENR.ECCEEN<br><ul style="list-style-type: none"> <li>0b0: Correctable ECC error interrupt is not generated in 1bitECC mode</li> <li>0b1: Correctable ECC error interrupt is generated in 1bitECC mode</li> </ul>      | RO     |
| 0     | ECCEURS  | Uncorrectable ECC error interrupt   | This bit is the AND logic result of NFC_ISTR.ECCEUST and NFC_IENR.ECCEUEN<br><ul style="list-style-type: none"> <li>0b0: Uncorrectable ECC error interrupt is not generated in 1bitECC mode</li> <li>0b1: Uncorrectable ECC error interrupt is generated in 1bitECC mode</li> </ul> | RO     |

## 40.4.5.11 Timing Configuration Register 0 (NFC\_TMCR0)

| Offset | Bit Position |    |    |    |    |    |    |    |     |    |    |    |    |    |    |    |     |    |    |    |    |    |   |   |    |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|-----|----|----|----|----|----|----|----|-----|----|----|----|----|----|---|---|----|---|---|---|---|---|---|---|
| 0x804C | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23  | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15  | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7  | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x03030202   |    |    |    |    |    |    |    |     |    |    |    |    |    |    |    |     |    |    |    |    |    |   |   |    |   |   |   |   |   |   |   |
| Name   | TH           |    |    |    |    |    |    |    | TRP |    |    |    |    |    |    |    | TWP |    |    |    |    |    |   |   | TS |   |   |   |   |   |   |   |
| Access | RW           |    |    |    |    |    |    |    | RW  |    |    |    |    |    |    |    | RW  |    |    |    |    |    |   |   | RW |   |   |   |   |   |   |   |

| Bit   | Name | Function | Description                                         | Access |
|-------|------|----------|-----------------------------------------------------|--------|
| 31:24 | TH   | -        | Set the hold time of CLE/ALE/CE after WE/RE changes | RW     |
| 23:16 | TRP  | -        | Set the enable (low level) pulse width of RE        | RW     |
| 15:8  | TWP  | -        | Set the enable (low level) pulse width of WE        | RW     |
| 7:0   | TS   | -        | Set the CLE/ALE/CE setup time before WE/RE changes  | RW     |

## 40.4.5.12 Timing Configuration Register 1 (NFC\_TMC1)

| Offset | Bit Position |    |    |    |    |    |    |    |     |    |    |    |    |    |    |    |     |    |    |    |    |    |   |   |     |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|-----|----|----|----|----|----|----|----|-----|----|----|----|----|----|---|---|-----|---|---|---|---|---|---|---|
| 0x8050 | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23  | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15  | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7   | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x28080303   |    |    |    |    |    |    |    |     |    |    |    |    |    |    |    |     |    |    |    |    |    |   |   |     |   |   |   |   |   |   |   |
| Name   | TWB          |    |    |    |    |    |    |    | TRR |    |    |    |    |    |    |    | TRH |    |    |    |    |    |   |   | TWH |   |   |   |   |   |   |   |
| Access | RW           |    |    |    |    |    |    |    | RW  |    |    |    |    |    |    |    | RW  |    |    |    |    |    |   |   | RW  |   |   |   |   |   |   |   |

| Bit   | Name | Function | Description                                         | Access |
|-------|------|----------|-----------------------------------------------------|--------|
| 31:24 | TWB  | -        | Set the time from WE rising edge to RB falling edge | RW     |
| 23:16 | TRR  | -        | Set the time from RB rising edge to RE falling edge | RW     |
| 15:8  | TRH  | -        | Set the disable (high level) pulse width of RE      | RW     |
| 7:0   | TWH  | -        | Set the disable (high level) pulse width of WE      | RW     |

The parameters set by the timing configuration register 0 and the timing configuration register 1 are shown in the figure below:

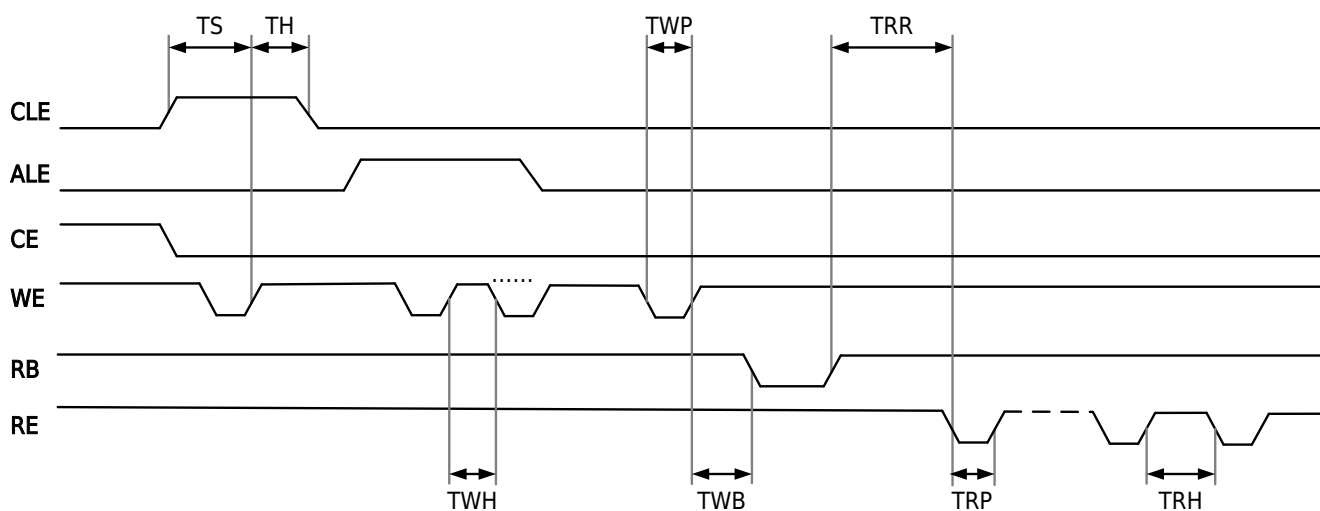


Figure 40-36 NFC timing control diagram 1

## 40.4.5.13 Timing Configuration Register 2 (NFC\_TMCR2)

| Offset | Bit Position |    |    |    |    |    |    |    |      |    |    |    |    |    |    |    |      |    |    |    |    |    |   |   |      |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|------|----|----|----|----|----|----|----|------|----|----|----|----|----|---|---|------|---|---|---|---|---|---|---|
| 0x805C | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23   | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15   | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7    | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x03050D03   |    |    |    |    |    |    |    |      |    |    |    |    |    |    |    |      |    |    |    |    |    |   |   |      |   |   |   |   |   |   |   |
| Name   | TADL         |    |    |    |    |    |    |    | TRTW |    |    |    |    |    |    |    | TWTR |    |    |    |    |    |   |   | TCCS |   |   |   |   |   |   |   |
| Access | RW           |    |    |    |    |    |    |    | RW   |    |    |    |    |    |    |    | RW   |    |    |    |    |    |   |   | RW   |   |   |   |   |   |   |   |

| Bit   | Name | Function | Description                                                                      | Access |
|-------|------|----------|----------------------------------------------------------------------------------|--------|
| 31:24 | TADL | -        | Set the time from ALE to read and write data                                     | RW     |
| 23:16 | TRTW | -        | Set the time from RE rising edge to WE falling edge                              | RW     |
| 15:8  | TWTR | -        | Set the time from WE rising edge to RE falling edge                              | RW     |
| 7:0   | TCCS | -        | Set the delay time after the Change read column and Change write column commands | RW     |

The parameters set by the timing configuration register 2 are shown in the figure below:

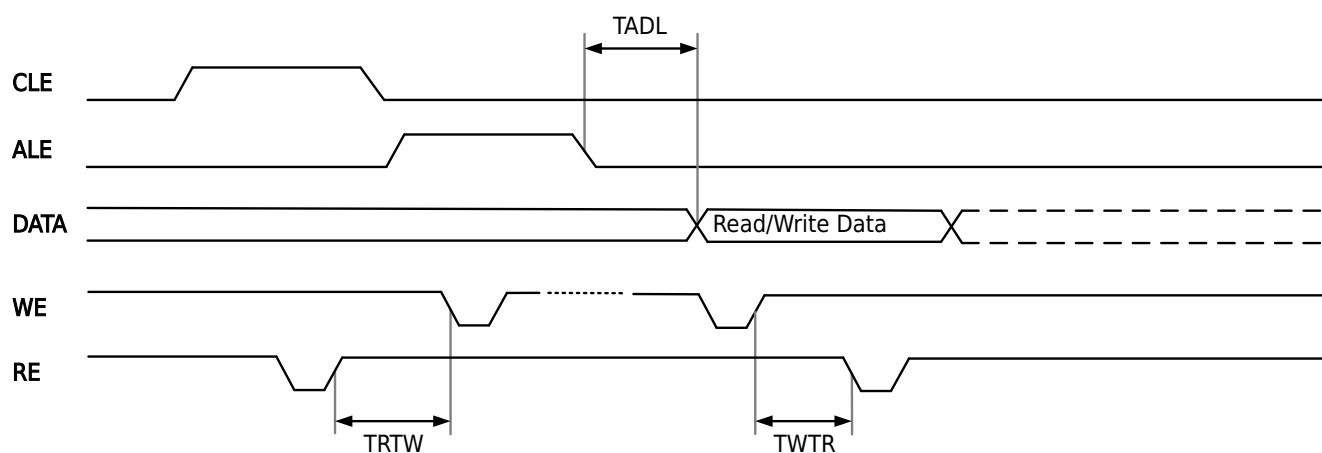


Figure 40-37 NFC Timing Control Diagram 2



#### 40.4.5.14 ECC Check Register (NFC\_ECCR)

##### 1bitECC Mode

In 1bitECC mode, ECCR0-15 (Offset: 0x8060-0x809C) are used as ECC result registers:

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |        |    |    |   |   |   |   |   |   |  |  |  |  |  |  |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|--------|----|----|---|---|---|---|---|---|--|--|--|--|--|--|
| -      | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9  | 8      | 7  | 6  | 5 | 4 | 3 | 2 | 1 | 0 |  |  |  |  |  |  |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |        |    |    |   |   |   |   |   |   |  |  |  |  |  |  |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | ME |    | SE | ERRLOC |    |    |   |   |   |   |   |   |  |  |  |  |  |  |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RO |    | RO |        | SE | RO |   |   |   |   |   |   |  |  |  |  |  |  |

| Bit   | Name     | Function         | Description                                                                                                                                                                                                                                                                                                                                                                                      | Access |
|-------|----------|------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:14 | Reserved | Reserved         | -                                                                                                                                                                                                                                                                                                                                                                                                | -      |
| 13    | ME       | Multi-bit error  | <ul style="list-style-type: none"> <li>0b0: Multi-bit ECC error not detected</li> <li>0b1: Multi-bit ECC error detected</li> </ul>                                                                                                                                                                                                                                                               | RO     |
| 12    | SE       | Single-bit error | <ul style="list-style-type: none"> <li>0b0: Single-bit ECC error not detected</li> <li>0b1: Single-bit ECC error detected</li> </ul>                                                                                                                                                                                                                                                             | RO     |
| 11:0  | ERRLOC   | Error position   | Indicates the position of the error, which is meaningful only when SE is set.<br>[2:0]: Bit position in the error byte <ul style="list-style-type: none"> <li>0b000: Bit0 error</li> <li>.....</li> <li>0b111: Bit7 error</li> </ul> [11:3]: Position of the error byte in 512 bytes <ul style="list-style-type: none"> <li>0x000: 1st Byte</li> <li>.....</li> <li>0x1FF: 512th Byte</li> </ul> | RO     |

##### 4bitECC Mode

In 4bitECC mode, ECCR7-71 (Offset: 0x807C-0x817C) is used as the ECC status register and the syndrome register. As shown in the following table, 512 bytes are used as a segment, and each segment corresponds to 8 ECC syndrome coefficients. For example: The segment 0 syndrome register corresponds to the first 512 bytes of ECC value, and contains a total of 4 registers (segment 0 syndrome coefficient 1 and 2 registers (ECC\_SECTION0\_SYND12), segment 0 syndrome coefficient 3 and 4 registers (ECC\_SECTION0\_SYND34), segment 0 syndrome coefficient 5 and 6 registers (ECC\_SECTION0\_SYND56), segment 0 syndrome coefficient 7 and 8 registers (ECC\_SECTION0\_SYND78) ), each register contains 2 ECC syndrome coefficients.

Table 40-116 ECCR Register List in 4bitECC Mode

| Offset        | Register          | Description                  | Width | Reset value |
|---------------|-------------------|------------------------------|-------|-------------|
| 0x807C        | ECC_STAT          | ECC Status Register          | 32    | 0x00000000  |
| 0x8080-0x808C | ECC_SECTION0_SYND | Segment 0 Companion Register | 32    | 0x00000000  |

| Offset        | Register           | Description                   | Width | Reset value |
|---------------|--------------------|-------------------------------|-------|-------------|
| 0x8090-0x809C | ECC_SECTION1_SYND  | Segment 1 Companion Register  | 32    | 0x00000000  |
| 0x80A0-0x80AC | ECC_SECTION2_SYND  | Segment 2 Companion Register  | 32    | 0x00000000  |
| 0x80B0-0x80BC | ECC_SECTION3_SYND  | Segment 3 Companion Register  | 32    | 0x00000000  |
| 0x80C0-0x80CC | ECC_SECTION4_SYND  | Segment 4 Companion Register  | 32    | 0x00000000  |
| 0x80D0-0x80DC | ECC_SECTION5_SYND  | Segment 5 Companion Register  | 32    | 0x00000000  |
| 0x80E0-0x80EC | ECC_SECTION6_SYND  | Segment 6 Companion Register  | 32    | 0x00000000  |
| 0x80F0-0x80FC | ECC_SECTION7_SYND  | Segment 7 Companion Register  | 32    | 0x00000000  |
| 0x8100-0x810C | ECC_SECTION8_SYND  | Segment 8 Companion Register  | 32    | 0x00000000  |
| 0x8110-0x811C | ECC_SECTION9_SYND  | Segment 9 Companion Register  | 32    | 0x00000000  |
| 0x8120-0x812C | ECC_SECTION10_SYND | Segment 10 Companion Register | 32    | 0x00000000  |
| 0x8130-0x813C | ECC_SECTION11_SYND | Segment 11 Companion Register | 32    | 0x00000000  |
| 0x8140-0x814C | ECC_SECTION12_SYND | Segment 12 Companion Register | 32    | 0x00000000  |
| 0x8150-0x815C | ECC_SECTION13_SYND | Segment 13 Companion Register | 32    | 0x00000000  |
| 0x8160-0x816C | ECC_SECTION14_SYND | Segment 14 Companion Register | 32    | 0x00000000  |
| 0x8170-0x817C | ECC_SECTION15_SYND | Segment 15 Companion Register | 32    | 0x00000000  |

The following table shows the 8bitECC value register corresponding to each segment (512 bytes), taking the companion register 0 (0x8080-0x808C) as an example.

**Table 40-117 Each Segment ECCR Register in 4bitECC Mode**

| Offset | Register            | Segment Syndrome Coefficient Register Name        | Width | Reset Value |
|--------|---------------------|---------------------------------------------------|-------|-------------|
| 0x8080 | ECC_SECTION0_SYND12 | Segment 0 Syndrome Coefficients 1 and 2 Registers | 32    | 0x00000000  |
| 0x8084 | ECC_SECTION0_SYND34 | Segment 0 Syndrome Coefficients 3 and 4 Registers | 32    | 0x00000000  |

| Offset | Register            | Segment Syndrome Coefficient Register Name        | Width | Reset Value |
|--------|---------------------|---------------------------------------------------|-------|-------------|
| 0x8088 | ECC_SECTION0_SYND56 | Segment 0 Syndrome Coefficients 5 and 6 Registers | 32    | 0x00000000  |
| 0x808C | ECC_SECTION0_SYND78 | Segment 0 Syndrome Coefficients 7 and 8 Registers | 32    | 0x00000000  |

**ECC Status Register (ECC\_STAT)**

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |        |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|--------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0x807C | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15     | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |        |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | ECCSEC |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RO     |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit   | Name     | Function       | Description                                                                                                                                                                                                 | Access |
|-------|----------|----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:16 | Reserved | Reserved       | -                                                                                                                                                                                                           | -      |
| 15:0  | ECCSEC   | Error position | Corresponding to paragraph 15-0:<br><ul style="list-style-type: none"> <li>0b0: No ECC error occurred in the corresponding section</li> <li>0b1: ECC error occurred in the corresponding section</li> </ul> | RO     |

## Segment x Syndrome Coefficients 1 and 2 Register (ECC\_SECTIONx\_SYND12) (x=0-15)

Offset: 0x8080+0x10\*x

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |          |    |   |   |    |   |   |   |   |   |   |   |  |  |  |  |  |  |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----------|----|---|---|----|---|---|---|---|---|---|---|--|--|--|--|--|--|
| -      | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11       | 10 | 9 | 8 | 7  | 6 | 5 | 4 | 3 | 2 | 1 | 0 |  |  |  |  |  |  |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |          |    |   |   |    |   |   |   |   |   |   |   |  |  |  |  |  |  |
| Name   | Reserved     |    |    |    | S2 |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | Reserved |    |   |   | S1 |   |   |   |   |   |   |   |  |  |  |  |  |  |
| Access |              |    |    |    | RO |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |          |    |   |   | RO |   |   |   |   |   |   |   |  |  |  |  |  |  |

| Bit   | Name     | Function | Description                     | Access |
|-------|----------|----------|---------------------------------|--------|
| 31:29 | Reserved | Reserved | -                               | -      |
| 28:16 | S2       | -        | $\alpha_2$ syndrome coefficient | RO     |
| 15:13 | Reserved | Reserved | -                               | -      |
| 12:0  | S1       | -        | $\alpha$ syndrome coefficient   | RO     |

## Segment x Syndrome Coefficients 3 and 4 Register (ECC\_SECTIONx\_SYND34) (x=0-15)

Offset: 0x8084+0x10\*x

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |          |    |   |   |    |   |   |   |   |   |   |   |  |  |  |  |  |  |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----------|----|---|---|----|---|---|---|---|---|---|---|--|--|--|--|--|--|
| -      | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11       | 10 | 9 | 8 | 7  | 6 | 5 | 4 | 3 | 2 | 1 | 0 |  |  |  |  |  |  |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |          |    |   |   |    |   |   |   |   |   |   |   |  |  |  |  |  |  |
| Name   | Reserved     |    |    |    | S4 |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | Reserved |    |   |   | S3 |   |   |   |   |   |   |   |  |  |  |  |  |  |
| Access |              |    |    |    | RO |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |          |    |   |   | RO |   |   |   |   |   |   |   |  |  |  |  |  |  |

| Bit   | Name     | Function | Description                     | Access |
|-------|----------|----------|---------------------------------|--------|
| 31:29 | Reserved | Reserved | -                               | -      |
| 28:16 | S4       | -        | $\alpha_4$ syndrome coefficient | RO     |
| 15:13 | Reserved | Reserved | -                               | -      |
| 12:0  | S3       | -        | $\alpha_3$ syndrome coefficient | RO     |

## Segment x Syndrome Coefficients 5 and 6 Register (ECC\_SECTIONx\_SYND56) (x=0-15)

Offset: 0x8088+0x10\*x

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |          |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |  |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|--|--|
| -      | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15       | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |  |  |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |          |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |  |
| Name   | Reserved     |    | S6 |    |    |    |    |    |    |    |    |    |    |    |    |    | Reserved |    | S5 |    |    |    |   |   |   |   |   |   |   |   |   |   |  |  |
| Access |              |    | RO |    |    |    |    |    |    |    |    |    |    |    |    |    |          |    | RO |    |    |    |   |   |   |   |   |   |   |   |   |   |  |  |

| Bit   | Name     | Function | Description                     | Access |
|-------|----------|----------|---------------------------------|--------|
| 31:29 | Reserved | Reserved | -                               | -      |
| 28:16 | S6       | -        | $\alpha_6$ syndrome coefficient | RO     |
| 15:13 | Reserved | Reserved | -                               | -      |
| 12:0  | S5       | -        | $\alpha_5$ syndrome coefficient | RO     |

## Segment x Syndrome Coefficients 7 and 8 Register (ECC\_SECTIONx\_SYND78) (x=0-15)

Offset: 0x808C+0x10\*x

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |          |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |  |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|--|--|
| -      | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15       | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |  |  |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |          |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |  |
| Name   | Reserved     |    | S8 |    |    |    |    |    |    |    |    |    |    |    |    |    | Reserved |    | S7 |    |    |    |   |   |   |   |   |   |   |   |   |   |  |  |
| Access |              |    | RO |    |    |    |    |    |    |    |    |    |    |    |    |    |          |    | RO |    |    |    |   |   |   |   |   |   |   |   |   |   |  |  |

| Bit   | Name     | Function | Description                     | Access |
|-------|----------|----------|---------------------------------|--------|
| 31:29 | Reserved | Reserved | -                               | -      |
| 28:16 | S8       | -        | $\alpha_8$ syndrome coefficient | RO     |
| 15:13 | Reserved | Reserved | -                               | -      |
| 12:0  | S7       | -        | $\alpha_7$ syndrome coefficient | RO     |

## 40.5 Precautions

1. The SMC related registers can only be set in the Ready state; the DMC related registers can only be set in the Config or Low Power state.
2. When using the SMC or DMC module to read external memory, the Memory Device transmits data to the controller after receiving the read command from the controller. The clock used by the controller to sample the data sent back can be selected through SMC\_BACR.CKSEL or DMC\_BACR.CKSEL. For specific settings, please refer to the register section. By selecting a different sampling clock, the time margin of the communication path can be improved.

SMC access in synchronous mode.

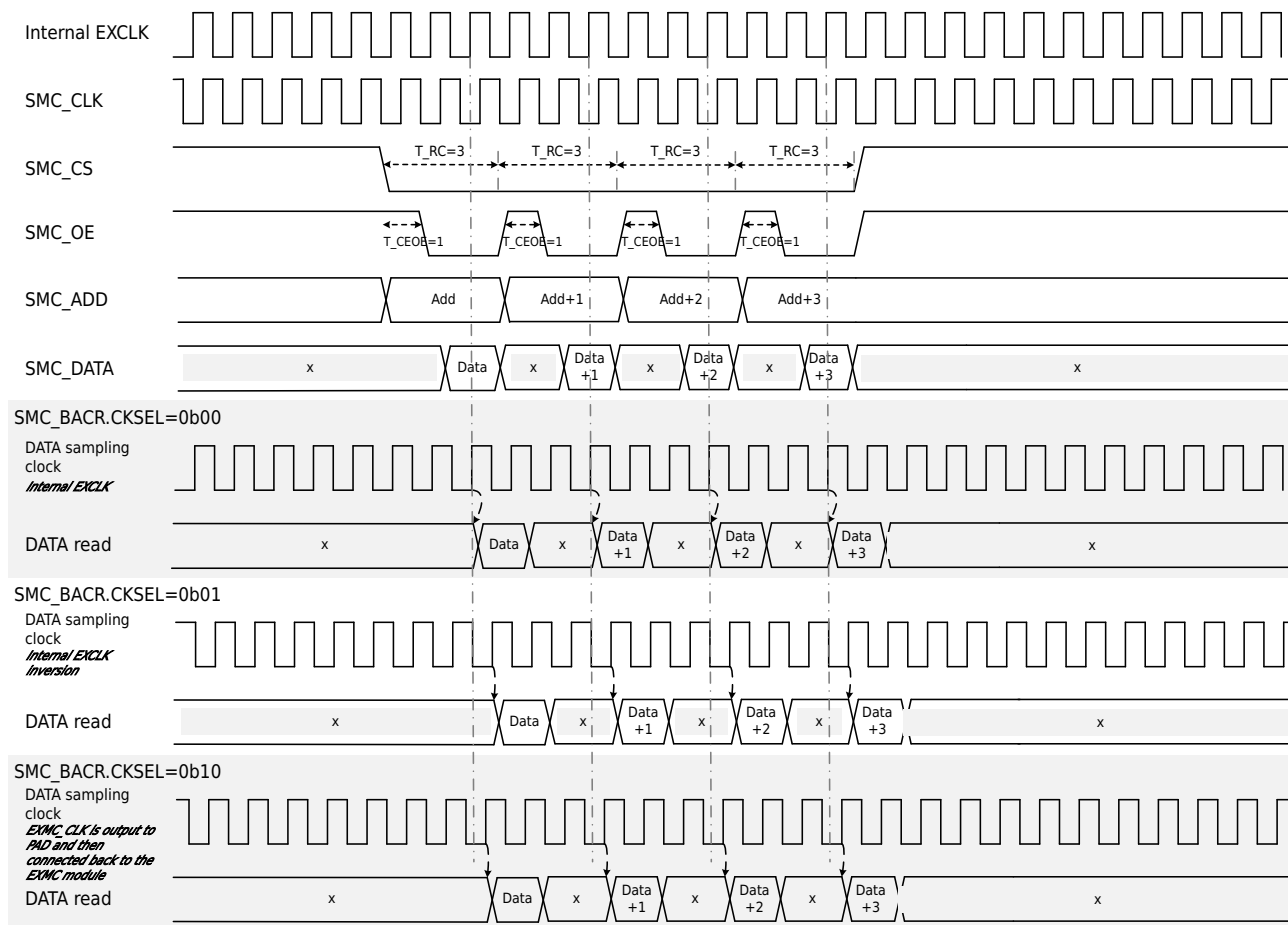


Figure 40-38 External Feedback Clock Access

3. When using the SMC and DMC functions, if you access the NFC register, you need to insert the NFC DUMMY read access after the access to clear the NFC access status.

## 41 Digital Video Port (DVP)

### 41.1 Introduction

Digital Video Port (DVP) is a synchronous parallel interface, which can collect 8-bit, 10-bit, 12-bit or 14-bit high-speed data streams from external CMOS camera modules. Support software synchronization and hardware synchronization. Support data stream acquisition frequency control and window clipping control. Supports data stream acquisition in various formats such as monochrome or raw Bayer format, YCbCr4:2:2, RGB565 progressive video, and compressed data (JPEG).

### 41.2 Features

The following table outlines the basic functions and features of DVP.

**Table 41-1 DVP Basic Functions and Features**

|                 |                                                    |
|-----------------|----------------------------------------------------|
| Basic function  | 8-bit, 10-bit, 12-bit, 14-bit Parallel Interface   |
|                 | Single Frame Mode and Continuous Mode              |
|                 | Software and Hardware Line/Frame Sync              |
|                 | Frame Acquisition Frequency Control                |
|                 | Window Cropping                                    |
|                 | FIFO Control                                       |
| Data format     | Monochrome or Raw Bayer Format                     |
|                 | YCbCr4:2:2 Progressive Video                       |
|                 | RGB565 Progressive Video                           |
|                 | JPEG Compressed Data                               |
| Interrupt event | Frame Start Interrupt and Event                    |
|                 | Frame End Interrupt and Event                      |
|                 | Line Start Interrupt and Event                     |
|                 | Line End Interrupt and Event                       |
|                 | Software Synchronization Error Interrupt and Event |
|                 | FIFO Overflow Error Interrupt and Event            |

## 41.3 Functional Description

### 41.3.1 System Block Diagram

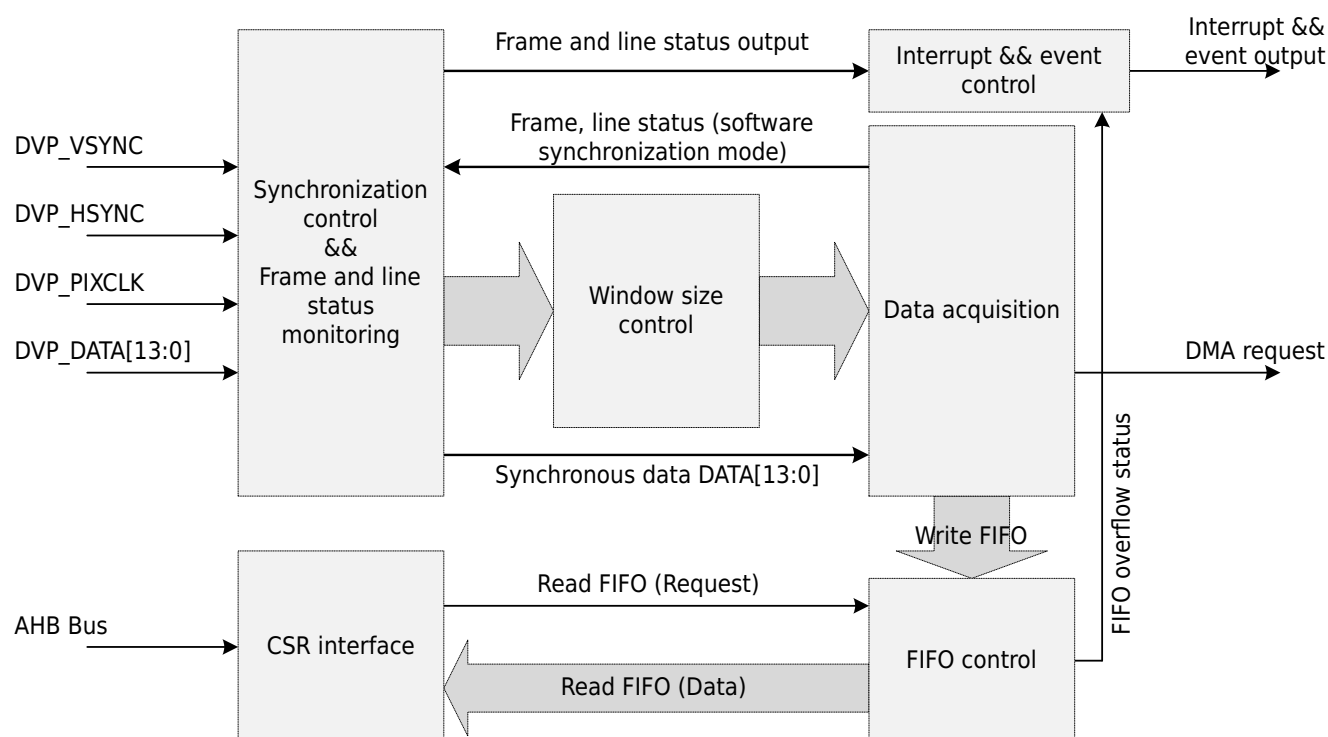


Figure 41-1 DVP Basic Block Diagram

The following table lists the input ports of DVP.

Table 41-2 DVP Port List

| Port Name      | Direction | Function                         |
|----------------|-----------|----------------------------------|
| DVP_VSYNC      | Input     | Frame synchronization input port |
| DVP_HSYNC      | Input     | Line synchronization input port  |
| DVP_PIXCLK     | Input     | Clock input port                 |
| DVP_DATA[13:0] | Input     | Data input port                  |



### 41.3.2 Video Data Format

#### Monochrome Format

- Each pixel is 8 bits.

The following table shows how the data is stored.

Table 41-3 Monochrome Format Video Data Storage Method

| Byte Address | 31:24 | 23:16 | 15:8 | 7:0 |
|--------------|-------|-------|------|-----|
| 0x00         | n+3   | n+2   | n+1  | n   |
| 0x04         | n+7   | n+6   | n+5  | n+4 |

#### YCbCr Format

- Alternating storage of Y, Cb, and Cr in one buffer: CbYCrYCbYCr...
- DVP module support for YCbCr4:2:2 format data transfer with pixel components including Y (luminance), Cb and Cr (Blue chrominance and red chrominance). Each component is encoded in 8 bits, with luminance and chrominance stored alternately.

The following table shows how the data is stored.

Table 41-4 YCbCr Format Video Data Storage Method

| Byte Address | 31:24 | 23:16  | 15:8  | 7:0    |
|--------------|-------|--------|-------|--------|
| 0x00         | Y n+1 | Cr n   | Y n   | Cb n   |
| 0x04         | Y n+3 | Cr n+2 | Y n+2 | Cb n+2 |

#### RGB565 Format

- Alternating storage of RGB signals in one buffer: BGRBGRBGR...
- DVP module support for 16BPP (16 bits per pixel) : RGB565 (2 pixels represented in 32 bits), does not support 24BPP (palette format) and grayscale formats.

The following table shows how the data is stored.

Table 41-5 RGB565 Format Video Data Storage Method

| Byte Address | 31:27 | 26:21 | 20:16 | 15:11 | 10:5  | 4:0   |
|--------------|-------|-------|-------|-------|-------|-------|
| 0x00         | R n+1 | G n+1 | B n+1 | R n   | G n   | B n   |
| 0x04         | R n+3 | G n+3 | B n+3 | R n+3 | G n+3 | B n+3 |

#### JPEG Format

- JPEG image not stored by lines and frames. DVP\_VSYNC signal initiates acquisition, DVP\_HSYNC serves as data enable signal.
- The number of bytes in the active interval of DVP\_HSYNC may not be a multiple of 4.
- DVP module supports JPEG format data transfer (DVP\_CTR.JPEGEN=1).

### 41.3.3 Parallel Port Storage Format

DVP interface consists of 11, 13, 15, or 17 input signals. Supports slave mode only. Collects 8-bit, 10-bit, 12-bit, or 14-bit data based on DVP\_CTR.BITSEL setting.

Data DVP\_DATA output by the camera module synchronizes with pixel clock DVP\_PIXCLK, changing on the rising/falling edge of DVP\_PIXCLK based on its polarity; DVP\_VSYNC indicates frame start/end; DVP\_HSYNC indicates line start/end.

During DVP\_VSYNC or DVP\_HSYNC activity, it represents the synchronization interval, and data transfer is disabled; when both are inactive, it represents the data interval, and data transfer is enable. The signal waveform diagram shows that DVP\_DATA changes on the rising edge of DVP\_PIXCLK (the DVP module is configured to sample data DVP\_DATA on the falling edge of DVP\_PIXCLK). DVP\_VSYNC or DVP\_HSYNC are in synchronization intervals when at high level, and data transmission is disabled during these intervals.

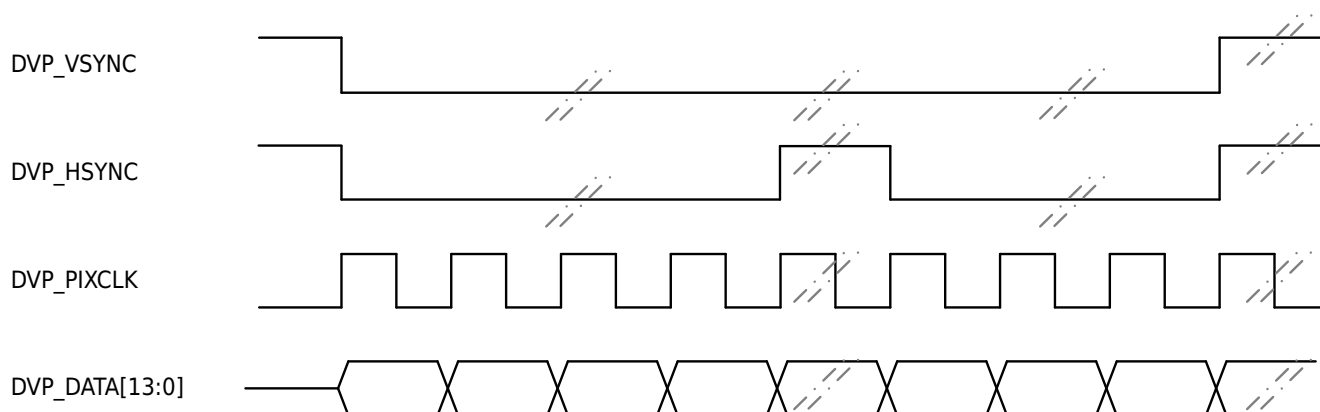


Figure 41-2 DVP Signal Waveform Diagram

## 8-bit Data

When DVP\_CTR.BITSEL=0b00, the DVP interface collects 8-bit data from DVP\_DATA[7:0] and stores it as 8-bit data, ignoring the input of DVP\_DATA[13:8]. In this case, a 32-bit data word is generated for every four pixel clock periods. The following table lists the bit arrangement of captured data bytes in two 32-bit words.

Table 41-6 8-bit Data DVP Storage Method

| Byte Address | 31:24     | 23:16     | 15:8      | 7:0       |
|--------------|-----------|-----------|-----------|-----------|
| 0x00         | Dn+3[7:0] | Dn+2[7:0] | Dn+1[7:0] | Dn[7:0]   |
| 0x04         | Dn+7[7:0] | Dn+6[7:0] | Dn+5[7:0] | Dn+4[7:0] |

## 10-bit Data

When DVP\_CTR.BITSEL=0b01, the DVP interface collects 10-bit data from DVP\_DATA[9:0] and stores it in the 10 least significant bits of a 16-bit word, ignoring the input of DVP\_DATA[13:10]. At that time, the remaining most significant bits (Bit15-10) in the DVP\_DTR register are cleared. In this case, one 32-bit data word is generated every two pixel clock periods, as shown in the table below.

Table 41-7 10-bit Data DVP Storage Method

| Byte Address | 31:24 | 23:16     | 15:8 | 7:0       |
|--------------|-------|-----------|------|-----------|
| 0x00         | 0     | Dn+1[9:0] | 0    | Dn[9:0]   |
| 0x04         | 0     | Dn+3[9:0] | 0    | Dn+2[9:0] |

## 12-bit Data

When DVP\_CTL.BITSEL=0b10, the DVP interface collects 12-bit data from DVP\_DATA[11:0] and stores it in the 12 least significant bits of a 16-bit word, ignoring the input of DVP\_DATA[13:12]. At that time, the remaining most significant bits (Bit15-12) in the DVP\_DTR register are cleared. In this case, one 32-bit data word is generated every two pixel clock periods, as shown in the table below.

Table 41-8 12-bit Data DVP Storage Method

| Byte Address | 31:24 | 23:16      | 15:8 | 7:0        |
|--------------|-------|------------|------|------------|
| 0x00         | 0     | Dn+1[11:0] | 0    | Dn[11:0]   |
| 0x04         | 0     | Dn+3[11:0] | 0    | Dn+2[11:0] |

## 14-bit Data

When DVP\_CTL.BITSEL=0b11, the DVP interface collects 14-bit data from DVP\_DATA[13:0] and stores it in the 14 least significant bits of a 16-bit word. At that time, the remaining most significant bits (Bit15-14) in the DVP\_DTR register are cleared. In this case, one 32-bit data word is generated every two pixel clock periods, as shown in the table below.

Table 41-9 14-bit Data DVP Storage Method

| Byte Address | 31:24 | 23:16      | 15:8 | 7:0        |
|--------------|-------|------------|------|------------|
| 0x00         | 0     | Dn+1[13:0] | 0    | Dn[13:0]   |
| 0x04         | 0     | Dn+3[13:0] | 0    | Dn+2[13:0] |

## 41.3.4 Mode Selection

### 41.3.4.1 Single Frame Mode

In single frame mode (DVP\_CTL.CAPMD=1), only one frame is captured. After setting DVP\_CTL.CAPEN bit to 1, the DVP interface waits for the system to detect frame start and then collects data. Upon receiving a complete frame of data, the CAPEN bit in DVP\_CTL is automatically cleared, and the camera interface stops receiving data. If frame end interrupt is enabled, a corresponding interrupt or event (DVP\_FRAMEND) is generated.

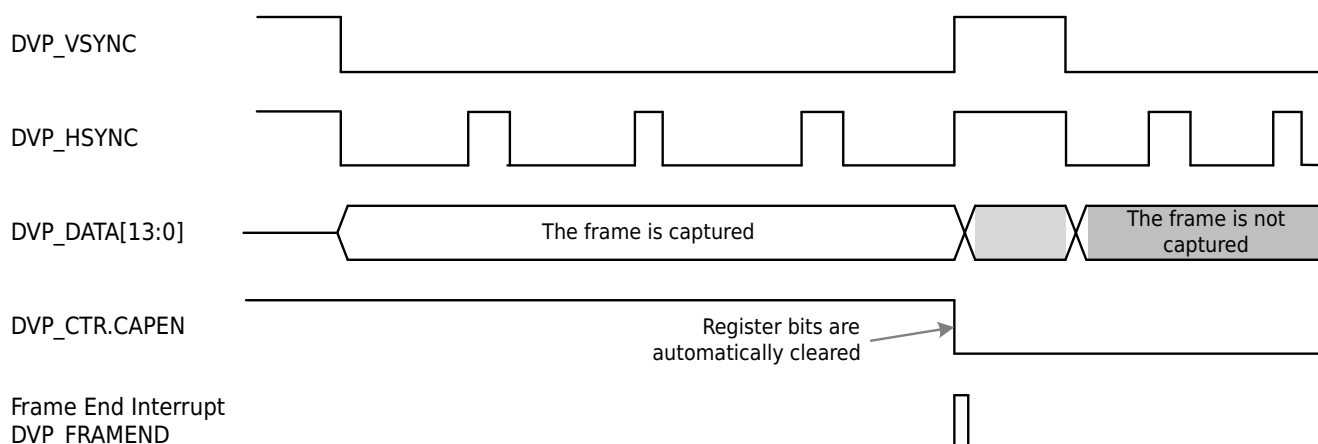


Figure 41-3 Single Frame Mode Data Acquisition Action

### 41.3.4.2 Continuous Mode

In continuous mode (`DVP_CTR.CAPMD=0`), image frames are continuously captured. After setting `DVP_CTR.CAPEN` bit to 1, the DVP interface waits for the system to detect frame start and then collects data. This process continues until the `CAPEN` bit in `DVP_CTR` is cleared. After the CPU issues a software clear action on the `DVP_CTR.CAPEN` bit, data acquisition continues until the current frame ends, then the `DVP_CTR.CAPEN` bit is cleared.

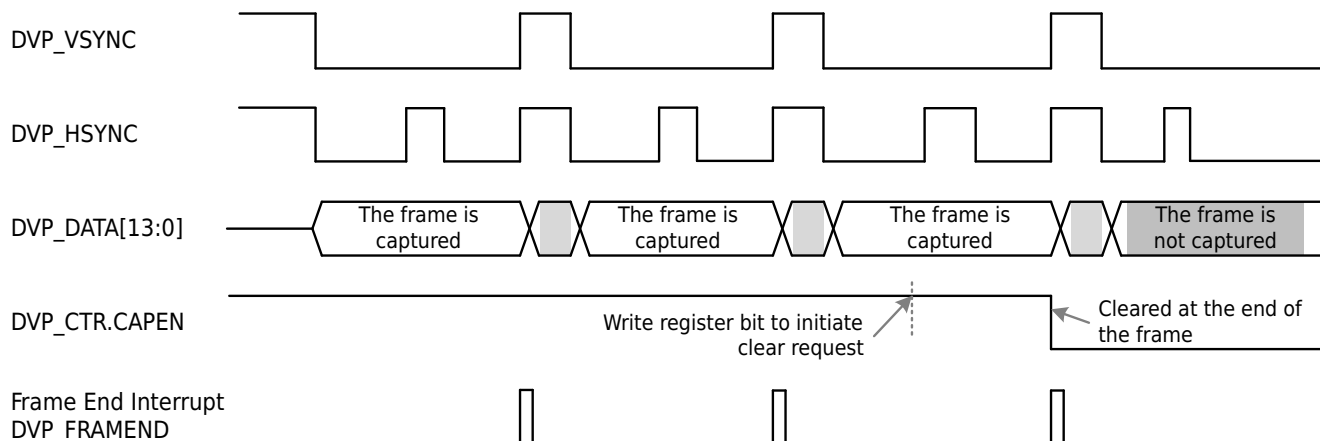


Figure 41-4 Continuous Mode Data Acquisition Action

In continuous mode, the frame capture frequency can be controlled by setting the `DVP_CTR.CAPFRC[1:0]` bits to reduce bandwidth. For example, when `DVP_CTR.CAPFRC[1:0]=0b01`, the camera interface samples the input data at intervals, that is, it collects image data once every frame.

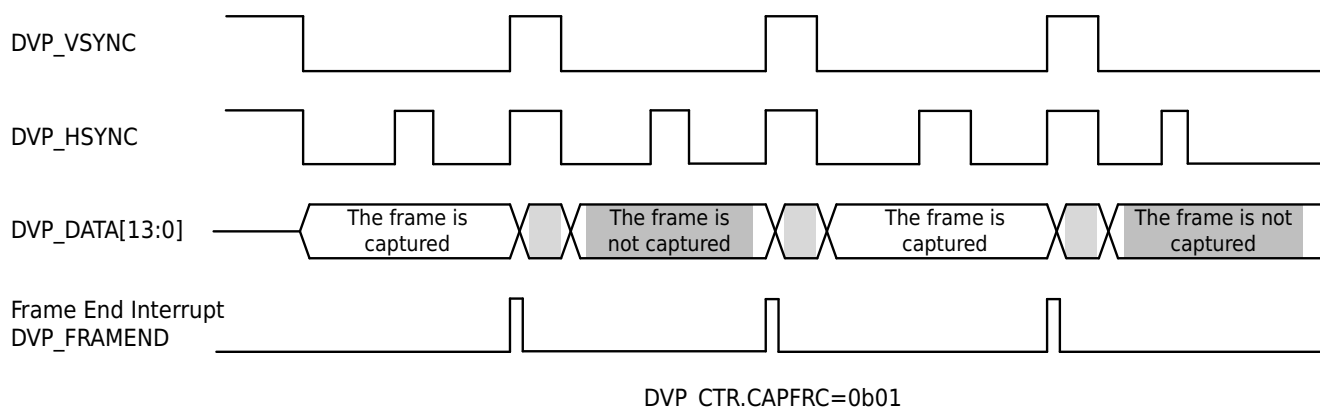


Figure 41-5 Frame Acquisition Frequency Control

## 41.3.5 Synchronization Control

### 41.3.5.1 Software Synchronization

In software synchronization mode, a 32-bit synchronization code embedded in the data stream is used to synchronize the data stream. These synchronization codes use unused `0x00/0xFF` in the data stream. There are four types of synchronization codes, all in the format `0xFF0000XY`. The DVP interface determines frame synchronization status by collecting a 32-bit word of data.

Four software synchronization codes represent the following events:

- Frame Start FS (`DVP_SSYNDR.FSDAT[7:0]`)
- Line Start LS (`DVP_SSYNDR.LSDAT[7:0]`)
- Line End LE (`DVP_SSYNDR.LEDAT[7:0]`)

- Frame End FE (DVP\_SSYNDR.FEDAT[7:0])

The XY in the synchronization code format 0xFF0000XY is determined by the four sets of register bits mentioned above. For example, when the captured data XY matches DVP\_SSYNDR.FSDAT[7:0], it indicates the start of a frame data transfer.

The four synchronization codes also have a bit masking function, allowing only the unmasked bits of the synchronization code to be used for comparison. The masking bits for each synchronization code are set by the corresponding bits in DVP\_SSYNMR. When the masking function is enabled, a single bit can be selected for synchronization code comparison to detect frame/line start and end. This means multiple synchronization codes can represent frame/line start and end as long as the unmasked bits match. For example, when DVP\_SSYNDR.FSDAT[7:0] is set to 0xA5 and DVP\_SSYNMR.FSMASK[7:0] is set to 0xF0, only the upper 4 bits of the data code need to be compared to detect the FS signal.

According to the software synchronization mechanism, DVP only supports 8-bit data format interfaces (DVP\_CTR.BITSEL[1:0]=0b00) and full-frame capture (DVP\_CTR.CAPFRC[1:0]=0b00). Other data width formats or frame skipping modes are not applicable and may produce unpredictable results. JPEG format data transfer is also not supported (DVP\_CTR.JPEGEN=0).

#### 41.3.5.2 Hardware Synchronization

In hardware synchronization mode, two port synchronization signals (DVP\_VSYNC/DVP\_HSYNC) are used. The camera interface module determines the start and end of frames/lines based on the level changes of DVP\_VSYNC/DVP\_HSYNC.

When data stream capture function is enabled in hardware synchronization mode (DVP\_CTR.CAPEN=1), data transfer can start after an inactive edge is detected on the DVP\_VSYNC port. Continuous frames are transferred to multiple consecutive buffers or a buffer with circular features by DMA. A frame transfer end interrupt is activated at the end of each frame (when the frame transfer end interrupt DVP\_FRAMEND is enabled).

#### 41.3.6 Window Cropping

The camera interface can use the cropping function to select a rectangular window from the received image. The starting coordinates and size of the window (horizontal size in pixel clock counts, vertical size in row counts) are specified by the window cropping offset register (DVP\_CPSFTR) and window cropping size register (DVP\_CPSZER).

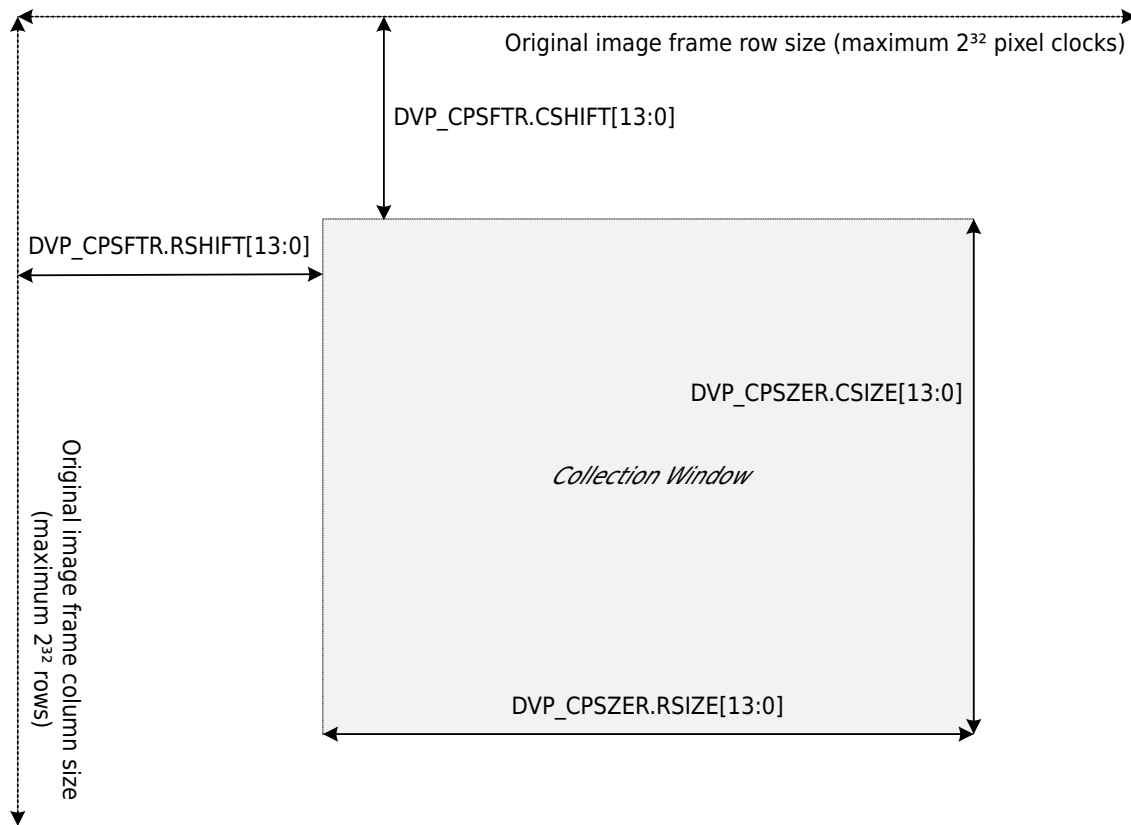


Figure 41-6 DVP Window Configuration Diagram

The starting coordinates of the cropping window are specified as a row number (starting from 0) and a pixel clock count (starting from 0), and the window size is specified as the number of rows and pixel clock counts. If the DVP\_VSYNC signal becomes active before the number of rows specified in the window cropping size register (DVP\_CPSZER) is completed, image capture stops, and a frame transfer end interrupt (DVP\_FRAMEND) is generated if interrupts are enabled. The following figure is a schematic diagram of window cropping data acquisition.

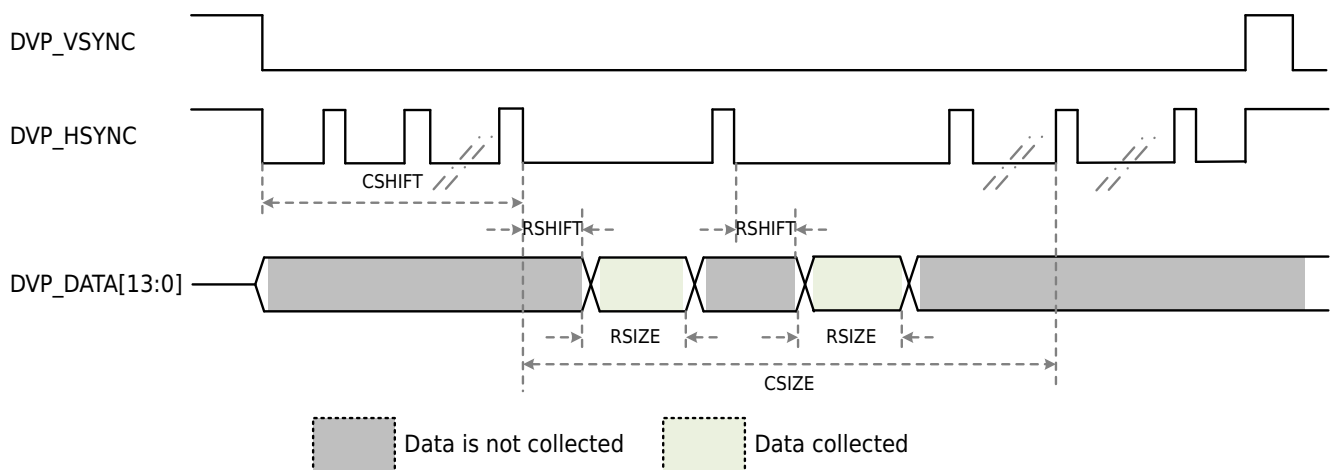


Figure 41-7 Window Cropping Data Acquisition Diagram

When JPEG format data transfer is enabled (`DVP_CTR.JPEGEN=1`), the window cropping function is not applicable, and the window cropping function must be disabled.

### 41.3.7 FIFO Control

To manage the data transfer speed on AHB, an 8-word deep FIFO is equipped in the DVP module with a FIFO controller. The read pointer increments each time data is read from the camera interface FIFO, and the write pointer increments each time data is written to the camera interface FIFO. If the data write speed exceeds the rate that the AHB interface can handle, the FIFO will overflow, and data will be overwritten. At this time, if the FIFO overflow error interrupt is enabled, a corresponding interrupt will be generated.

If a software synchronization error event occurs, the FIFO will be reset, and the DVP interface will wait for the start of a new data frame.

### 41.3.8 DMA Control

When `DVP_CTR.CAPEN=1`, the DMA interface is activated. Each time the camera interface receives a complete 32-bit data block in the data register (`DVP_DTR`), it can trigger a DMA read request. When DMA accesses the DMA data transfer register (`DVP_DMR`) through the AHB interface, it reads the corresponding cached data from the FIFO. This DMA read request is event number 90 in the event list. For more information, refer to the "Interrupt Event List" in the Interrupt Controller (INTC) chapter. For DMA settings, refer to the "DMA Controller" chapter.

When transmitting JPEG data, the number of bytes in a line may not be a multiple of 4. The DVP interface will pad with zeros when it detects the end of a line and has not yet formed a 32-bit word, and trigger a DMA read request.

### 41.3.9 Interrupt and Events

#### 41.3.9.1 Frame Transfer Status Interrupt and Event

During normal image frame transfer, interrupts can be generated for various states:

Table 41-10 Frame Transfer Interrupt

| Interrupt/Event Name | Interrupt/Event Symbol   | Interrupt Enable Bit    |
|----------------------|--------------------------|-------------------------|
| Frame transfer start | <code>DVP_FRAMSTA</code> | <code>IER.FSIEN</code>  |
| Line transfer start  | <code>DVP_LINESTA</code> | <code>IER.LSIEN</code>  |
| Line transfer end    | <code>DVP_LINEEND</code> | <code>IER.LEIEN</code>  |
| Frame transfer end   | <code>DVP_FRAMEND</code> | <code>IER.FEIEEN</code> |

#### 41.3.9.2 Software Synchronization Error Interrupt and Event

In software synchronization mode, if the received synchronization code exhibits an error sequence as shown in the following table, the corresponding status bit (`DVP_STR.SQUERF`) will be set to 1. If the corresponding interrupt enable (`DVP_IER.SQUERIEN`) is set to be valid, a software synchronization error (`DVP_SQUERR`) interrupt will be generated.

Table 41-11 Synchronization Code Error Sequence

| FS | LS | LE | FE |
|----|----|----|----|
| 0  | 0  | 0  | 1  |
| 0  | 0  | 1  | x  |
| 0  | 1  | x  | x  |

| FS | LS | LE | FE |
|----|----|----|----|
| 1  | 0  | 1  | x  |
| 1  | 1  | 0  | 1  |

In the table, 1 indicates the synchronization sequence is received, 0 indicates it is not received, and x indicates whether the sequence is received does not affect the sequence error judgment.

For example, when FS:LS:LE:FE=0b1101, it indicates that after receiving the frame start and line start sequences, the frame end sequence is received, but the line end sequence is not received, resulting in a software synchronization sequence error event.

When a synchronization code error occurs, the internal frame capture control logic will reset, and no data from the current frame will be received until DVP\_STR.SQUERF is cleared, at which point the start of the next frame's frame start sequence will be received and monitored.

#### 41.3.9.3 FIFO Overflow Error Interrupt and Event

When a FIFO overflow occurs, the corresponding status bit (DVP\_STR.FIFOERF) will be set to 1. If the corresponding interrupt enable (DVP\_IER.FIFOERIEN) is set to be valid, a FIFO overflow error (DVP\_FIFOERR) interrupt will be generated.



## 41.4 Register Description

### 41.4.1 Register List

Base address: 0x40055800

Table 41-12 Digital Video Port (DVP) Register Offset

| Offset | Register   | Description                            | Width |
|--------|------------|----------------------------------------|-------|
| 0x0000 | DVP_CTR    | Control Register                       | 32    |
| 0x0004 | DVP_DTR    | Data Register                          | 32    |
| 0x0008 | DVP_STR    | Status Register                        | 32    |
| 0x000C | DVP_IER    | Interrupt Register                     | 32    |
| 0x0010 | DVP_DMR    | DMA Data Transfer Register             | 32    |
| 0x0020 | DVP_SSYNDR | Software Synchronization Data Register | 32    |
| 0x0024 | DVP_SSYNMR | Software Synchronization Mask Register | 32    |
| 0x0028 | DVP_CPSFTR | Window Cropping Offset Register        | 32    |
| 0x002C | DVP_CPSZER | Window Cropping Size Register          | 32    |

## 41.4.2 Control Register (DVP\_CTRL)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |       |          |        |    |        |   |           |           |          |        |        |        |       |       |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-------|----------|--------|----|--------|---|-----------|-----------|----------|--------|--------|--------|-------|-------|---|
| 0x0000 | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14    | 13       | 12     | 11 | 10     | 9 | 8         | 7         | 6        | 5      | 4      | 3      | 2     | 1     | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |       |          |        |    |        |   |           |           |          |        |        |        |       |       |   |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | DVPEN | Reserved | BITSEL |    | CAPFRC |   | VSYNCSSEL | HSYNCSSEL | PIXCKSEL | SWSYNC | JPEGEN | CROPEN | CAPMD | CAPEN |   |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |       |          |        |    |        |   |           |           |          |        |        |        |       |       |   |

| Bit   | Name      | Function                              | Description                                                                                                                                                                                                                                                                             | Access |
|-------|-----------|---------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:15 | Reserved  | Reserved                              | -                                                                                                                                                                                                                                                                                       | -      |
| 14    | DVPEN     | DVP Module Enable                     | <ul style="list-style-type: none"> <li>0b0: DVP module disabled</li> <li>0b1: DVP module enabled</li> </ul>                                                                                                                                                                             | RW     |
| 13:12 | Reserved  | Reserved                              | -                                                                                                                                                                                                                                                                                       | -      |
| 11:10 | BITSEL    | Data Width Selection                  | <ul style="list-style-type: none"> <li>0b00: 8-bit data width</li> <li>0b01: 10-bit data width</li> <li>0b10: 12-bit data width</li> <li>0b11: 14-bit data width</li> </ul>                                                                                                             | RW     |
| 9:8   | CAPFRC    | Frame Acquisition Frequency Selection | <ul style="list-style-type: none"> <li>0b00: Full-frame capture (capture all frames)</li> <li>0b01: Capture every 1 frame (bandwidth reduced by 50%)</li> <li>0b10: Capture every 3 frames (bandwidth reduced by 75%)</li> <li>0b11: Full-frame capture (capture all frames)</li> </ul> | RW     |
| 7     | VSYNCSSEL | Frame Synchronization Level Selection | <ul style="list-style-type: none"> <li>0b0: Synchronization function enabled when DVP_VSYNC is low</li> <li>0b1: Synchronization function enabled when DVP_VSYNC is high</li> </ul>                                                                                                     | RW     |
| 6     | HSYNCSSEL | Line Synchronization Level Selection  | <ul style="list-style-type: none"> <li>0b0: Synchronization function enabled when DVP_HSYNC is low</li> <li>0b1: Synchronization function enabled when DVP_HSYNC is high</li> </ul>                                                                                                     | RW     |
| 5     | PIXCKSEL  | Pixel Clock Selection:                | <ul style="list-style-type: none"> <li>0b0: Data sampled on the falling edge of DVP_PIXCLK</li> <li>0b1: Data sampled on the rising edge of DVP_PIXCLK</li> </ul>                                                                                                                       | RW     |
| 4     | SWSYNC    | Software Synchronization Enable       | <ul style="list-style-type: none"> <li>0b0: Software synchronization disabled</li> <li>0b1: Software synchronization enabled</li> </ul>                                                                                                                                                 | RW     |
| 3     | JPEGEN    | JPEG Transmission Enable              | <ul style="list-style-type: none"> <li>0b0: Non-JPEG format data transfer</li> <li>0b1: JPEG format data transfer</li> </ul>                                                                                                                                                            | RW     |

| Bit | Name   | Function                    | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                      | Access |
|-----|--------|-----------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 2   | CROPEN | Window Cropping Enable      | <ul style="list-style-type: none"> <li>0b0: Window cropping function disabled</li> <li>0b1: Window cropping function enabled</li> </ul>                                                                                                                                                                                                                                                                                                                          | RW     |
| 1   | CAPMD  | Acquisition Mode Selection  | <ul style="list-style-type: none"> <li>0b0: Continuous mode</li> <li>0b1: Single frame mode</li> </ul>                                                                                                                                                                                                                                                                                                                                                           | RW     |
| 0   | CAPEN  | Acquisition Function Enable | <ul style="list-style-type: none"> <li>0b0: Disable capture</li> <li>0b1: Enable capture</li> </ul> <div>  <b>Note</b><br/>           In single frame mode, this bit is automatically cleared after one frame of data ends. In continuous mode, after writing zero to this bit, the clearing operation will only take effect after the end of the current frame.         </div> | RW     |

#### 41.4.3 Data Register (DVP\_DTR)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |  |  |  |  |  |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|--|--|--|--|--|--|
| 0x0004 | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |  |  |  |  |  |  |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |  |  |  |  |  |
| Name   | DATA         |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |  |  |  |  |  |
| Access | RO           |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |  |  |  |  |  |

| Bit  | Name | Function     | Description                  | Access |
|------|------|--------------|------------------------------|--------|
| 31:0 | DATA | Acquire Data | Current set of data captured | RO     |

## 41.4.4 Status Register (DVP\_STR)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |         |        |     |     |     |     |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---------|--------|-----|-----|-----|-----|
| 0x0008 | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5       | 4      | 3   | 2   | 1   | 0   |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |         |        |     |     |     |     |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   | FIFOERF | SQUERF | FEF | LEF | LSF | FSF |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   | RW      | RW     | RW  | RW  | RW  | RW  |

| Bit  | Name     | Function                            | Description                                                                                                                                             | Access |
|------|----------|-------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:6 | Reserved | Reserved                            | -                                                                                                                                                       | -      |
| 5    | FIFOERF  | FIFO Overflow Error Flag            | <ul style="list-style-type: none"> <li>0b0: No FIFO overflow error occurred</li> <li>0b1: FIFO overflow error occurred</li> </ul>                       | RW     |
| 4    | SQUERF   | Software Synchronization Error Flag | <ul style="list-style-type: none"> <li>0b0: No software synchronization error occurred</li> <li>0b1: Software synchronization error occurred</li> </ul> | RW     |
| 3    | FEF      | Frame Transfer End Flag             | <ul style="list-style-type: none"> <li>0b0: Frame transfer end not detected</li> <li>0b1: Frame transfer end detected</li> </ul>                        | RW     |
| 2    | LEF      | Line Transfer End Flag              | <ul style="list-style-type: none"> <li>0b0: Line transfer end not detected</li> <li>0b1: Line transfer end detected</li> </ul>                          | RW     |
| 1    | LSF      | Line Transfer Start Flag            | <ul style="list-style-type: none"> <li>0b0: Line transfer start not detected</li> <li>0b1: Line transfer start detected</li> </ul>                      | RW     |
| 0    | FSF      | Frame Transfer Start Flag           | <ul style="list-style-type: none"> <li>0b0: Frame transfer start not detected</li> <li>0b1: Frame transfer start detected</li> </ul>                    | RW     |

## 41.4.5 Interrupt Register (DVP\_IER)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |           |          |       |       |       |       |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|-----------|----------|-------|-------|-------|-------|
| 0x000C | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5         | 4        | 3     | 2     | 1     | 0     |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |           |          |       |       |       |       |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   | FIFOERIEN | SQUERIEN | FEIEN | LEIEN | LSIEN | FSIEN |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   | RW        | RW       | RW    | RW    | RW    | RW    |

| Bit  | Name      | Function                                         | Description                                                                         | Access |
|------|-----------|--------------------------------------------------|-------------------------------------------------------------------------------------|--------|
| 31:6 | Reserved  | Reserved                                         | -                                                                                   | -      |
| 5    | FIFOERIEN | FIFO Overflow Error Interrupt Enabled            | <ul style="list-style-type: none"> <li>0b0: Disable</li> <li>0b1: Enable</li> </ul> | RW     |
| 4    | SQUERIEN  | Software Synchronization Error Interrupt Enabled | <ul style="list-style-type: none"> <li>0b0: Disable</li> <li>0b1: Enable</li> </ul> | RW     |
| 3    | FEIEN     | Frame Transfer End Interrupt Enabled             | <ul style="list-style-type: none"> <li>0b0: Disable</li> <li>0b1: Enable</li> </ul> | RW     |
| 2    | LEIEN     | Line Transfer End Interrupt Enabled              | <ul style="list-style-type: none"> <li>0b0: Disable</li> <li>0b1: Enable</li> </ul> | RW     |
| 1    | LSIEN     | Line Transfer Start Interrupt Enabled            | <ul style="list-style-type: none"> <li>0b0: Disable</li> <li>0b1: Enable</li> </ul> | RW     |
| 0    | FSIEN     | Frame Transfer Start Interrupt Enabled           | <ul style="list-style-type: none"> <li>0b0: Disable</li> <li>0b1: Enable</li> </ul> | RW     |

## 41.4.6 DMA Data Transfer Register (DVP\_DMR)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0x0010 | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | DMR          |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access | RO           |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit  | Name | Function           | Description                                                       | Access |
|------|------|--------------------|-------------------------------------------------------------------|--------|
| 31:0 | DMR  | DMA Transfer Value | Read from this register to get the current data in the read FIFO. | RO     |

#### 41.4.7 Software Synchronization Data Register (DVP\_SSYNDR)

| Offset | Bit Position |    |    |    |    |    |    |    |       |    |    |    |    |    |    |    |       |    |    |    |    |    |   |   |       |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|-------|----|----|----|----|----|----|----|-------|----|----|----|----|----|---|---|-------|---|---|---|---|---|---|---|
| 0x0020 | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23    | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15    | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7     | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |       |    |    |    |    |    |    |    |       |    |    |    |    |    |   |   |       |   |   |   |   |   |   |   |
| Name   | FEDAT        |    |    |    |    |    |    |    | LEDAT |    |    |    |    |    |    |    | LSDAT |    |    |    |    |    |   |   | FSDAT |   |   |   |   |   |   |   |
| Access | RW           |    |    |    |    |    |    |    | RW    |    |    |    |    |    |    |    | RW    |    |    |    |    |    |   |   | RW    |   |   |   |   |   |   |   |

| Bit   | Name  | Function                                   | Description                                                            | Access |
|-------|-------|--------------------------------------------|------------------------------------------------------------------------|--------|
| 31:24 | FEDAT | Frame End Syn-<br>chronization<br>Code     | Frame end synchronization code in software syn-<br>chronization mode   | RW     |
| 23:16 | LEDAT | Line End Syn-<br>chronization<br>Code      | Line end synchronization code in software syn-<br>chronization mode    | RW     |
| 15:8  | LSDAT | Line Start Syn-<br>chronization<br>Code    | Line start synchronization code in software syn-<br>chronization mode  | RW     |
| 7:0   | FSDAT | Frame Start Syn-<br>chroniza-<br>tion Code | Frame start synchronization code in software syn-<br>chronization mode | RW     |

#### 41.4.8 Software Synchronization Mask Register (DVP\_SSYNMR)

| Offset | Bit Position |    |    |    |    |    |    |    |       |    |    |    |    |    |    |    |       |    |    |    |    |    |   |   |       |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|-------|----|----|----|----|----|----|----|-------|----|----|----|----|----|---|---|-------|---|---|---|---|---|---|---|
| 0x0024 | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23    | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15    | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7     | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0xFFFFFFFF   |    |    |    |    |    |    |    |       |    |    |    |    |    |    |    |       |    |    |    |    |    |   |   |       |   |   |   |   |   |   |   |
| Name   | FEMSK        |    |    |    |    |    |    |    | LEMSK |    |    |    |    |    |    |    | LSMSK |    |    |    |    |    |   |   | FSMSK |   |   |   |   |   |   |   |
| Access | RW           |    |    |    |    |    |    |    | RW    |    |    |    |    |    |    |    | RW    |    |    |    |    |    |   |   | RW    |   |   |   |   |   |   |   |

| Bit   | Name  | Function                 | Description                                                 | Access |
|-------|-------|--------------------------|-------------------------------------------------------------|--------|
| 31:24 | FEMSK | Frame End Mask<br>Code   | Frame end mask code in software synchronization<br>mode     | RW     |
| 23:16 | LEMSK | Line End Mask<br>Code    | Line end mask code in software synchronization<br>mode      | RW     |
| 15:8  | LSMSK | Line Start Mask<br>Code  | Line start mask code in software synchronization<br>mode    | RW     |
| 7:0   | FSMSK | Frame Start<br>Mask Code | Frame start mask code in software synchroniza-<br>tion mode | RW     |

## 41.4.9 Window Cropping Offset Register (DVP\_CPSFTR)

| Offset | Bit Position |        |    |    |    |    |    |    |    |    |    |    |    |    |    |          |        |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|--------|----|----|----|----|----|----|----|----|----|----|----|----|----|----------|--------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0x0028 | 31           | 30     | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16       | 15     | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |        |    |    |    |    |    |    |    |    |    |    |    |    |    |          |        |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | Reserved     | CSHIFT |    |    |    |    |    |    |    |    |    |    |    |    |    | Reserved | RSHIFT |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access |              | RW     |    |    |    |    |    |    |    |    |    |    |    |    |    |          | RW     |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit   | Name     | Function      | Description                                                                                                                                                                                                                                                                                                                                                  | Access |
|-------|----------|---------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:30 | Reserved | Reserved      | -                                                                                                                                                                                                                                                                                                                                                            | -      |
| 29:16 | CSHIFT   | Column Offset | Indicates the starting row number of the window                                                                                                                                                                                                                                                                                                              | RW     |
| 15:14 | Reserved | Reserved      | -                                                                                                                                                                                                                                                                                                                                                            | -      |
| 13:0  | RSHIFT   | Row Offset    | Indicates the starting column number (clock pixel count) of the window<br><br><div>  <b>Note</b><br/>           In software synchronization mode, the row offset is shifted in multiples of 4. Bit1-Bit0 are readable and writable but functionally invalid.         </div> | RW     |

## 41.4.10 Window Cropping Size Register (DVP\_CPSZER)

| Offset | Bit Position |       |    |    |    |    |    |    |    |    |    |    |    |    |    |          |       |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|-------|----|----|----|----|----|----|----|----|----|----|----|----|----|----------|-------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0x002C | 31           | 30    | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16       | 15    | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |       |    |    |    |    |    |    |    |    |    |    |    |    |    |          |       |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | Reserved     | CSIZE |    |    |    |    |    |    |    |    |    |    |    |    |    | Reserved | RSIZE |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access |              | RW    |    |    |    |    |    |    |    |    |    |    |    |    |    |          | RW    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit   | Name     | Function    | Description                                                                                                                                                                                                                                                                                                                                               | Access |
|-------|----------|-------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:30 | Reserved | Reserved    | -                                                                                                                                                                                                                                                                                                                                                         | -      |
| 29:16 | CSIZE    | Column Size | Indicates the size of each column in the window, i.e., the number of rows in the window                                                                                                                                                                                                                                                                   | RW     |
| 15:14 | Reserved | Reserved    | -                                                                                                                                                                                                                                                                                                                                                         | -      |
| 13:0  | RSIZE    | Row Size    | Indicates the size of each row in the window (clock pixel count), i.e., the number of columns in the window<br><br><div>  <b>Note</b><br/>           The column size of the window must be set to a multiple of 4. Bit1-Bit0 are invalid and fixed at 0.         </div> | RW     |



## 42 Cryptographic Coprocessing Module (CPM)

### 42.1 Introduction

The Cryptographic Coprocessing Module (CPM) includes four sub-modules: HASH Secure Hash Algorithm, TRNG True Random Number Generator, PKE Public Key Engine, and SKE Symmetric Encryption Algorithm Module.

- HASH is the SHA-2 version of SHA-256 (Secure Hash Algorithm), which complies with the national standard "FIPS PUB 180-3" issued by the US National Institute of Standards and Technology, and can generate a 256-bit message digest output for messages with a length of up to  $2^{64}$  bits. Supports HMAC (Hash-based Message Authentication Code) applications, using the SHA-256 hash function to authenticate messages.
- TRNG is a random number generator based on continuous analog noise, providing 64-bit random numbers.
- PKE implements hardware acceleration for asymmetric signature/verification or encryption/decryption algorithms.
- SKE includes three algorithms: AES, DES, and SM4, and eight operating modes: ECB, CBC, CTR, CFB, OFB, GCM, CCM, and CMAC.

### 42.2 Features

#### 42.2.1 PKE Features

- Supports modular arithmetic: modular addition, modular subtraction, modular multiplication, modular exponentiation, and modular inverse
- Supports elliptic curve point operations: point addition, point doubling, point multiplication, and point-on-curve verification
- Supports big number operations: large integer multiplication
- Supports both interrupt and polling modes for status monitoring

#### 42.2.2 SKE Features

SKE supports 5 algorithms: AES-128, AES-192, AES-256, DES and SM4, among which: AES and SM4 support 8 working modes, including ECB, CBC, CTR, CFB, OFB, GCM, CCM and CMAC; DES supports 6 working modes except GCM and CCM.

## 42.3 Functional Description

### 42.3.1 Secure Hash Algorithm (HASH)

#### 42.3.1.1 Algorithm Introduction

The steps of the secure hash algorithm are as follows:

First pad the message so that its length is exactly a number that is only 64 bits less than a multiple of 512. The padding method is to attach a 1 to the back of the message, followed by the required multiple 0s, and then attach a 64-bit message length (before padding) to make the message length exactly an integer multiple of 512 bits.

Secondly, initialize the eight 32-bit variables A, B, C, D, E, F, G, and H in hexadecimal. Then start the main loop of the algorithm, process 512-bit messages at a time, the number of loops is the number of 512-bit packets in the message.

The main loop performs a total of 64 operations, which are called compression functions. Each operation includes shift, circular shift, logical operation, modulo 232 addition, etc. The operation process is shown in the figure below. The final output is the cascade of A, B, C, D, E, F, G, and H. Among them,  $W_t$  is the temporary value used in the  $t$ -th step obtained from the 512-bit message,  $K_t$  is the constant value used in the  $t$ -th step, and  $t$  ( $0 \leq t \leq 63$ ) is a step in the 64-step period.

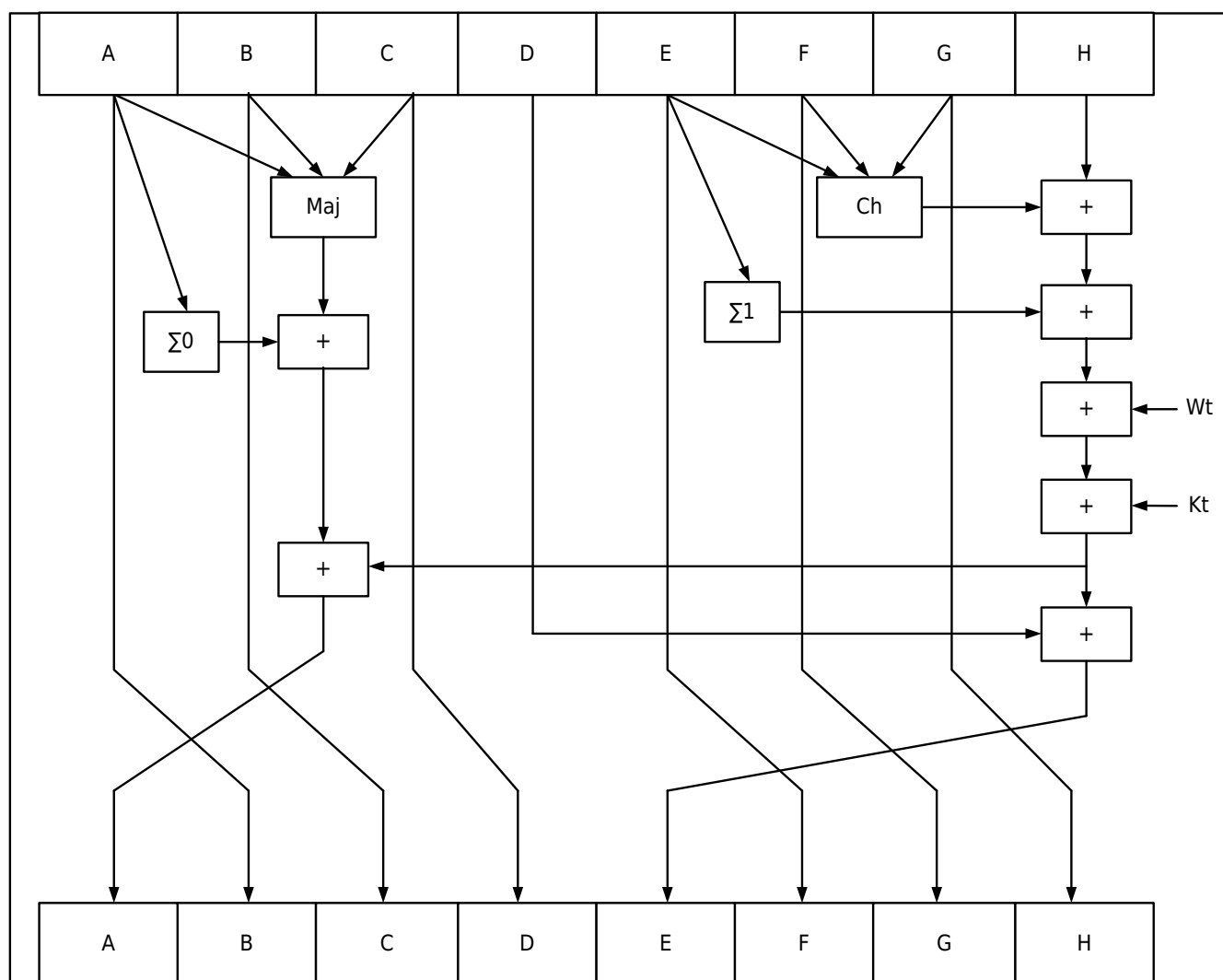


Figure 42-1 HASH Algorithm Flowchart

The HASH module integrates the hardware HMAC function; the HMAC function can also be implemented by directly calling the HASH operation according to the HMAC algorithm through software.

#### 42.3.1.2 Operating Procedures

The operation process of the HASH module is as follows:

- Software operation process

- Step1.** The software fills the original data according to the algorithm rules, and groups the filled messages by 512 bits.
- Step2.** Write the data into the HASH\_DR register.
- Step3.** If this operation is the first group of data in the message group, set HASH\_CR.FST\_GRP=1.
- Step4.** Set HASH\_CR.START=1 to start this module for operation.



##### Note

Steps 3 and 4 can be performed at the same time.

- Step5.** To judge whether the operation of this module is completed, use the following methods: Keep reading HASH\_CR.START until the bit is 0, which means the operation is complete; Or keep reading HASH\_CR.BUSY until the bit is 0, which means the operation is completed; Or set the HASH interrupt to determine the end of the operation through the interrupt method.
- Step6.** If this operation is not the last set of data in the message grouping, return to step 2.
- Step7.** If this operation is the last set of data in the message group, read the summary register (HASH\_HR) to obtain the result of this operation. If you need to perform the calculation again, return to step 1.

- Using DMA operation process:

- Step1.** The messages to be processed are grouped into 512-bit blocks, and the last group of messages needs to be pre-filled into 512-bit blocks according to the rules.
- Step2.** Set the HASH operation start mode of the internal trigger event register to DMA transfer completion and data block transfer completion; set the DMA transfer start mode of the internal trigger event register to HASH operation completion (or other internal trigger events).
- Step3.** Set HASH\_CR.MODE=0b00, HASH operation mode; set HASH\_CR.FST\_GRP=1, and set HASH\_CR.START=1 to start HASH operation. If another internal trigger event is selected to initiate DMA, it is necessary to wait for the trigger event to occur before starting the HASH computation.
- Step4.** Wait for HASH\_CR.CYC\_END=1, the HASH operation is completed, and the bit is cleared.
- Step5.** Read the result from the hash digest register HASH\_HR.



##### Note

When there is only one set of operation data, set both HASH\_CR.FST\_GRP=1 and HASH\_CR.KMSG\_END=1 simultaneously.

#### 42.3.1.3 Message Padding

The padding packet processing steps of SHA-256 are as follows:

- Step1.** Raw message grouping

Divide the original message into L groups of 512 bits in size. Let the total number of bits of the original message be I. If  $I \% 512 < 448$ , then the grouping number L is  $I / 512$ ; if  $I \% 512 \geq 448$ , then the grouping number L is  $I / 512 + 1$ .

**Step2. Add length****a. Add padding:**

Add padding bits at the end of the  $l/512$ th group of the message packet: one 1 and several 0s, the number of 0s can be zero. If  $l\%512 < 448$ , pad the data bits to make the length satisfy  $448 \bmod 512$  (the last 64 bits are reserved to represent the length of the original message); if  $l\%512 \geq 448$ , fill the 512-bit data block of the  $l/512$ th group with a 1 and several 0s, and fill the first 448 bits of the  $L$ th ( $L=l/512+1$ ) th group with 0s.

**b. Add raw message length:**

A 64-bit block representing the original message length, as a 64-bit unsigned integer. Add the original message length to the last 64-bit of the  $L$ th packet.

An example to illustrate the process of filling a group is as follows:

**● Example 1**

The original message is a string of characters "abcde". Its ASCII code is represented by a binary bit string as follows: "01100001 01100010 01100011 01100100 01100101". The steps to add the length are as follows:

1. Add "1". The filled message is "01100001 01100010 01100011 01100100 01100101 1".
2. Add "0". Since the original message length is 40-bit, the number of zeros to be added is  $512-64-40-1=407$ .

The padded message becomes (hex) :

```
61626364 65800000 00000000 00000000
00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000
00000000 00000000
```

3. Add the original message length. The original message length is 40-bit, expressed as two 32-bit words (hexadecimal) : 00000000 00000028.

The padded message becomes (hex) :

```
61626364 65800000 00000000 00000000
00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000028
```

**● Example 2**

The original message is the string "abcbcdcedefdefgefghfghighijhijkijkljklmklmnlmnomnopnopq". Each character can be converted to 8-bit by its ASCII code, so the length of the message is  $l=56 * 8=448$ .

1. Add "1" and "0". The padded message (hexadecimal) is the first message block:

```
61626364 62636465 63646566 64656667
65666768 66676869 6768696A 68696A6B
696A6B6C 6A6B6C6D 6B6C6D6E 6C6D6E6F
6D6E6F70 6E6F7071 80000000 00000000
```

2. Add the original message length. The original message length is 448-bit, expressed as two 32-bit words (in hexadecimal) : 00000000 000001C0.

The padded message (hexadecimal) is the second message block:

```
00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000
00000000 00000000 00000000 000001C0
```

#### 42.3.1.4 HMAC Operation

HMAC (Keyed-Hashing for Message Authentication) binds the message being processed to a key selected by the user in an irreversible manner for message authentication. Basically, HMAC consists of two nested underlying HASH operations with the addition of a shared secret:

$$\text{HMAC}(\text{message}) = \text{Hash}[(\text{key XOR opad}) \mid \text{Hash}((\text{key XOR ipad}) \mid \text{message})]$$

Where:

- iPad: 64 bytes (SHA-256) 0x36.
- opad: 64 bytes (SHA-256) 0x5C.
- |: represents the concatenation operator.

##### 42.3.1.4.1 HMAC Operation Process

- Step1.** Set HASH\_CR.MODE=0b01 to select HMAC operation mode;
- Step2.** If the key length exceeds 64 bytes, set HASH\_CR.LKEY=1 (in this case, HMAC will use a key as specified by the standard, replacing the given key).
- Step3.** Write the key into the HASH\_DR register until all key group operations are completed.
- Step4.** Write the message into the HASH\_DR register until all message group operations are completed.
- Step5.** Read the final hash result from the HASH\_HR register.

For data loading during the HMAC operation, either a software operation flow or DMA operation flow can be chosen. Specific examples are provided below:

##### ● HMAC Software Operation Flow

- Step1.** Set HASH\_CR.MODE=0b01 to select HMAC operation mode; Set HASH\_CR.LKEY=1 to select the long key mode.
- Step2.** Write the key group data into HASH\_DR register;
- Step3.** If it is the first data group, set HASH\_CR.FST\_GRP=1. Set HASH\_CR.START=1 to start key operation.
- Step4.** Wait until HASH\_CR.BUSY=0 indicating operation completion.
- Step5.** If it is not the last data group, repeat steps 2-4;
- Step6.** If it is the last data group, set HASH\_CR.KMSG\_END=1, and then set HASH\_CR.START=1 to start the last key operation. Wait until HASH\_CR.CYC\_END=1, indicating key operation completion, then clear this bit.
- Step7.** Write the message group data into the HASH\_DR register.
- Step8.** If it is the first message group, set HASH\_CR.FST\_GRP=1. Set HASH\_CR.START=1 to start the message operation.
- Step9.** Wait until HASH\_CR.BUSY=0 indicating operation completion.
- Step10.** If it is not the last message group, repeat steps 7-9.
- Step11.** If it is the last message group, set HASH\_CR.KMSG\_END=1, and then set HASH\_CR.START=1 to start the final message operation. Wait until HASH\_CR.CYC\_END=1, indicating all HMAC operations are completed, then clear this bit.
- Step12.** Read the final message authentication result from the HASH\_HR register.

### ● HMAC Uses DMA Operation Process

- Step1.** Set DMA to data block transmission mode. The size of the data block is 16 words. Both the key and the message need to be grouped according to the size of 16 words. If it is less than 16 words, please fill it according to the HASH message filling rules. Configure the DMA data block source address as the key address;
- Step2.** Set the HASH operation start mode of the internal trigger event register to DMA transfer completion and data block transfer completion; set the DMA transfer start mode of the internal trigger event register to HASH operation completion (or other internal trigger events);
- Step3.** Set HASH\_CR.MODE=0b01, HMAC operation mode; configure HASH\_CR.LKEY bit according to the key length; set HASH\_CR.FST\_GRP=1, HASH\_CR.START=1 to start the key operation; if other internal trigger events are selected, you need to wait for the trigger event to start the operation;
- Step4.** After waiting for HASH\_CR.CYC\_END=1, the key operation is completed and then clear this bit;
- Step5.** Stop DMA and point the DMA data block source address to the message address;
- Step6.** Set HASH\_CR.FST\_GRP=1, HASH\_CR.START=1 to start the message operation; or wait for other trigger events to start the message operation.
- Step7.** After waiting for the HASH\_CR.CYC\_END=1 message operation to complete, clear this bit;
- Step8.** Read the final message authentication result in the HASH\_HR register.



#### 42.3.1.5.2 Hardware Trigger Event Selection

The internal hardware trigger event selection registers HASH\_TRGSELA and HASH\_TRGSELB are configured to start the HASH operation after data transmission completion, typically used in conjunction with DMA. For example, when DMA\_2 is selected to transfer data and start HASH operation, first, configure DMA2\_TRGSELx (x=0-7) to select HASH\_INT to start DMA transfer; secondly, configure HASH\_TRGSELA to select DMA\_BTCx (x=0-7), indicating that a data block (512-bit) transfer is completed and the HASH operation is started, and configure HASH\_TRGSELB to select DMA\_TCx (x=0-7), indicating that all data block transfers are completed. After starting the HASH operation, DMA and HASH trigger each other until all data blocks have been transferred and processed.

### 42.3.2 True Random Number Generator (TRNG)

#### 42.3.2.1 Module Block Diagram

The TRNG module provides a true random number generator to generate a 64-bit random number.

The system block diagram of the TRNG is shown below. The random number generator is an analog random number oscillator circuit, which is used to obtain random noise; the algorithm module captures random noise and saves the result to the data module, and outputs it through the bus; the control module controls the mode and start of TRNG.

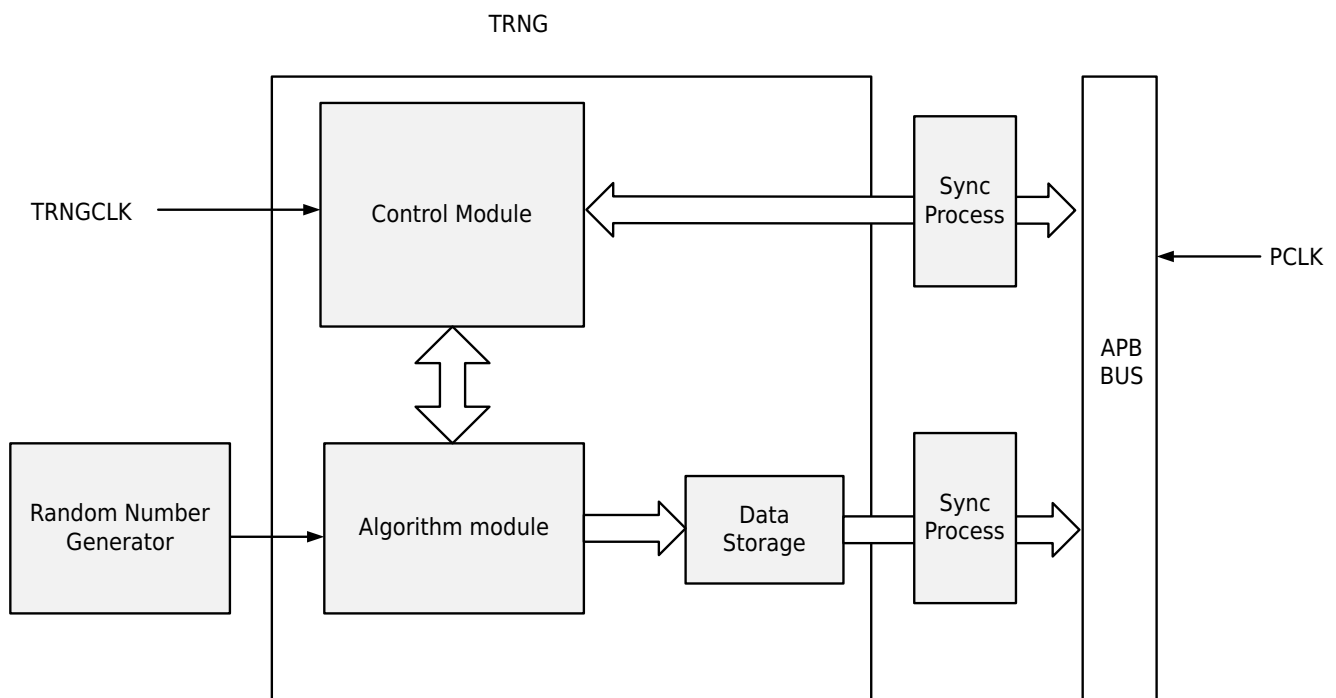


Figure 42-2 TRNG System Block Diagram

#### 42.3.2.2 Operating Procedures

The true random number generation process is as follows:

- Step1.** Turn on the random number generator circuit (set TRNG\_CR.EN bit to 1).
- Step2.** Configure the random number generation mode (set TRNG\_MR).
- Step3.** Start random number generation (set TRNG\_CR.RUN bit to 1).
- Step4.** Read random number (read TRNG\_DR).
- Step5.** Turn off the random number generator circuit (set TRNG\_CR.EN bit to 0).



### 42.3.2.3 Interrupt and Events

When the random number generation is finished, the TRNG\_CR.RUN bit is cleared by hardware and generates a random number generation interrupt request (TRNG\_END). When the random number generation ends, an event output is also generated, which can trigger the linkage of other modules.

## 42.3.3 Public Key Engine (PKE)

### 42.3.3.1 Module Introduction

The PKE (Public Key Engine) system of this product consists of one computation module and two dedicated RAM blocks. Compatible with software drivers, the system supports RSA (RSA1024/RSA2048/RSA3072/RSA4096), ECC (brainpoolp 160/brainpoolp 256/brainpoolp 512/SECP 192/SECP 224/SECP 256/SECP 384/SECP 521), and SM2.

Table 42-2 Operation Speed

| Operation mode         |                                 |                           | Speed (ops/sec) | Module Clock Frequency |
|------------------------|---------------------------------|---------------------------|-----------------|------------------------|
| Hardware Maximum Speed | 256-bit Point Multiplication    |                           | 75              | 120MHz                 |
|                        | 1024-bit Modular Exponentiation |                           | 32.4            |                        |
| Software Maximum Speed | ECC SECP256R1                   | ECDSA/ECDH Key Generation | 72              |                        |
|                        |                                 | ECDH Key Exchange         | 75              |                        |
|                        |                                 | ECDSA Signature           | 72              |                        |
|                        |                                 | ECDSA Verification        | 36              |                        |
|                        | SM2                             | SM2 Key Generation        | 72              |                        |
|                        |                                 | SM2 Key Exchange          | 45              |                        |
|                        |                                 | SM2 Signature             | 72              |                        |
|                        |                                 | SM2 Verification          | 36              |                        |
|                        |                                 | SM2 Encryption            | 36              |                        |
|                        |                                 | SM2 Decryption            | 69              |                        |
|                        | ECC SECP384R1                   | ECDSA/ECDH Key Generation | 30              |                        |
|                        |                                 | ECDH Key Exchange         | 30              |                        |
|                        |                                 | ECDSA Signature           | 30              |                        |
|                        |                                 | ECDSA Verification        | 14.4            |                        |
|                        | ECC SECP521R1                   | ECDSA/ECDH Key Generation | 13.2            |                        |
|                        |                                 | ECDH Key Exchange         | 13.8            |                        |
|                        |                                 | ECDSA Signature           | 13.2            |                        |
|                        |                                 | ECDSA Verification        | 6.6             |                        |

| Operation mode |          |                              | Speed (ops/sec) | Module Clock Frequency |
|----------------|----------|------------------------------|-----------------|------------------------|
|                | RSA 1024 | Key Generation (e=65537)     | 3               | 120MHz                 |
|                |          | CRT Key Generation (e=65537) | 3               |                        |
|                |          | Signature/Decryption         | 30              |                        |
|                |          | Verification/Encryption      | 828             |                        |
|                |          | CRT Signature/Decryption     | 96              |                        |
|                | RSA 2048 | Key Generation (e=65537)     | 1.32            |                        |
|                |          | CRT Key Generation (e=65537) | 1.26            |                        |
|                |          | Signature/Decryption         | 4.2             |                        |
|                |          | Verification/Encryption      | 240             |                        |
|                |          | CRT Signature/Decryption     | 15              |                        |
|                | RSA 4096 | Key Generation (e=65537)     | 16.2            |                        |
|                |          | CRT Key Generation (e=65537) | 15.6            |                        |
|                |          | Signature/Decryption         | 0.6             |                        |
|                |          | Verification/Encryption      | 66              |                        |
|                |          | CRT Signing/Decryption       | 1.8             |                        |

#### 42.3.3.2 Module Block Diagram

The PKE module offloads the CPU from complex public-key cryptographic computations. The CPU only needs to configure the input parameters, and the PKE will automatically perform the specified operation based on the current configuration. Currently, the PKE supports modular exponentiation for RSA and point multiplication for ECC. The CPU can monitor the PKE's operation status via polling or interrupt.

The PKE consists of a program storage unit (ROM), an Instruction Execution Unit (IEU), a 32-bit Arithmetic Logic Unit (ALU), a register file, and an interface module.

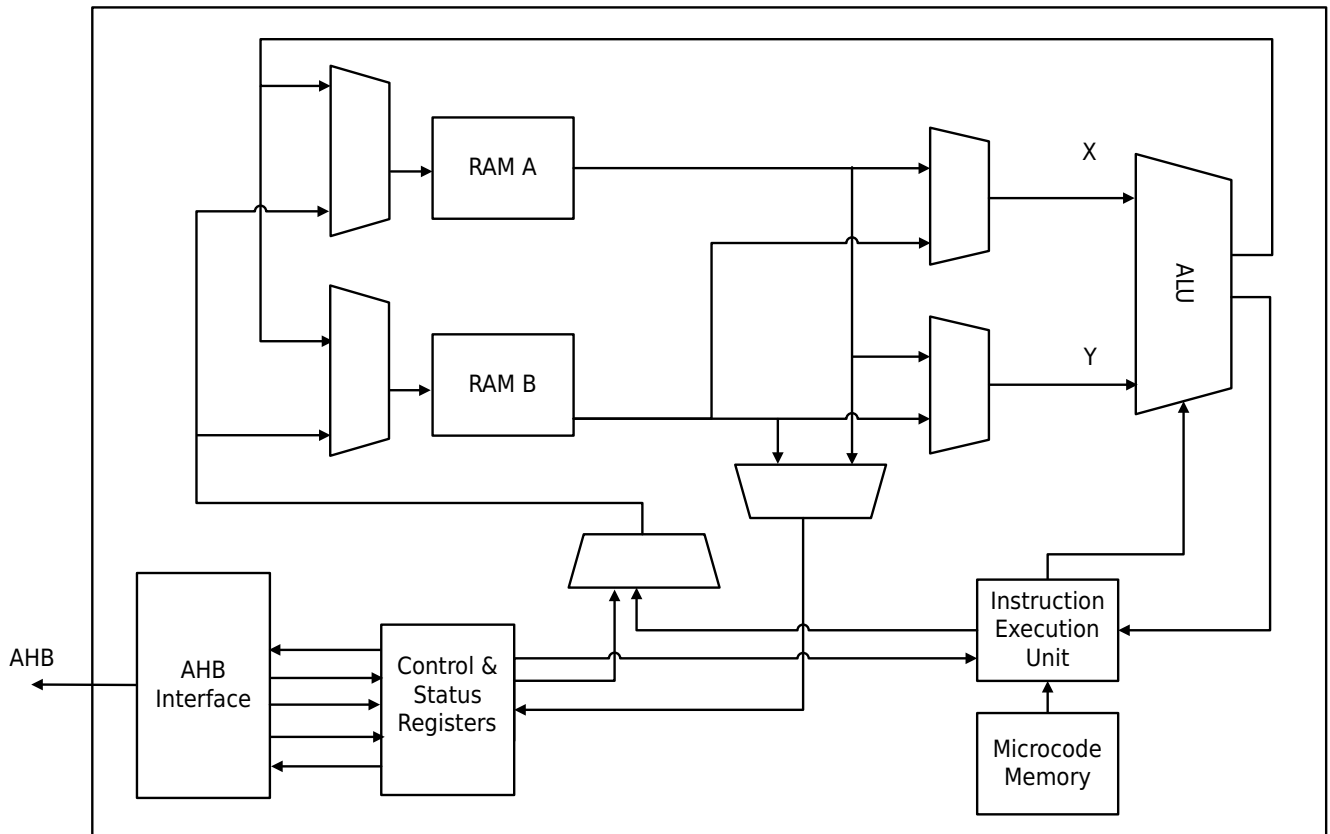


Figure 42-3 PKE System Block Diagram

#### 42.3.3.3 Programming Model

All external memory used by the PKE is mapped into the bus address space. This address mapping primarily includes all operands and computation results accessible by the CPU. The operands include the modulus, exponent, and certain intermediate variables.

In addition, the address space also contains PKE control and status registers. The CPU can configure and monitor the PKE module through these control and status registers.

In typical public-key cryptographic operations, the shortest operand is 192 bits. When writing data into the data RAM, word-endianness issues may arise. In the PKE module, words are arranged in little-endian order. The minimum operand width supported by the PKE is 32 bits (one word). If the operand width is not a multiple of 32 bits, zero-padding must be added to the higher bits before inputting the operand externally.

After triggering a PKE operation, the running status of the PKE can be checked through the status register. The completion of the target operation can be detected via polling or interrupt.



#### Note

The data RAM is word-aligned and does not support byte alignment.

Table 42-3 RAM Operand Mapping

| RAM  | Operand Space | ECC     |         |          | RSA     |          |          |          |
|------|---------------|---------|---------|----------|---------|----------|----------|----------|
|      |               | 256-bit | 512-bit | 1024-bit | 512-bit | 1024-bit | 2048-bit | 4096-bit |
| RAM0 | A0            | 0x1400  | 0x1400  | 0x1400   | 0x1400  | 0x1400   | 0x1400   | 0x1400   |
|      | A1            | 0x1424  | 0x1444  | 0x1484   | 0x1444  | 0x1484   | 0x1504   | 0x1604   |
|      | A2            | 0x1448  | 0x1488  | 0x1508   | 0x1488  | 0x1508   | 0x1608   | 0x1808   |
|      | A3            | 0x146C  | 0x14CC  | 0x158C   | 0x14CC  | 0x158C   | 0x170C   | 0x1A0C   |
|      | A4            | 0x1490  | 0x1510  | 0x1610   | 0x1510  | 0x1610   | 0x1810   | 0x1C10   |
|      | A5            | 0x14B4  | 0x1554  | 0x1694   | -       | -        | -        | -        |
|      | A6            | 0x14D8  | 0x1598  | 0x1718   | -       | -        | -        | -        |
|      | A7            | 0x14FC  | 0x15DC  | 0x179C   | -       | -        | -        | -        |
|      | A8            | 0x1520  | 0x1620  | 0x1820   | -       | -        | -        | -        |
|      | A9            | 0x1544  | 0x1664  | 0x18A4   | -       | -        | -        | -        |
| RAM1 | B0            | 0x2000  | 0x2000  | 0x2000   | 0x2000  | 0x2000   | 0x2000   | 0x2000   |
|      | B1            | 0x2024  | 0x2044  | 0x2084   | 0x2044  | 0x2084   | 0x2104   | 0x2204   |
|      | B2            | 0x2048  | 0x2088  | 0x2108   | 0x2088  | 0x2108   | 0x2208   | 0x2408   |
|      | B3            | 0x206C  | 0x20CC  | 0x218C   | 0x20CC  | 0x218C   | 0x230C   | 0x260C   |
|      | B4            | 0x2090  | 0x2110  | 0x2210   | 0x2110  | 0x2210   | 0x2410   | 0x2810   |
|      | B5            | 0x20B4  | 0x2154  | 0x2294   | -       | -        | -        | -        |
|      | B6            | 0x20D8  | 0x2198  | 0x2318   | -       | -        | -        | -        |
|      | B7            | 0x20FC  | 0x21DC  | 0x239C   | -       | -        | -        | -        |
|      | B8            | 0x2120  | 0x2220  | 0x2420   | -       | -        | -        | -        |
|      | B9            | 0x2144  | 0x2264  | 0x24A4   | -       | -        | -        | -        |

The table above shows the address allocation of the two RAM blocks under different operand lengths for ECC and RSA modes. Operand registers are distributed across two data RAMs, distinguished by the prefixes A and B. The addresses listed in the table are CPU-accessible addresses: RAMA has an address offset of 0x1400, while RAMB has an address offset of 0x2000. The actual memory space used in RAM is larger than what is shown in the table, as part of the space is reserved for storing intermediate variables that are not accessible by the CPU.

- In ECC mode, the PKE uses register spaces 0 to 9 in both RAMs. Each register space supports three configurations: (8+1) words, (16+1) words, and (32+1) words, with 1 extra word reserved for possible carry storage.
- In RSA mode, the PKE uses register spaces 0 to 4 in both RAMs. Each register space supports four configurations: (16+1) words, (32+1) words, (64+1) words, and (128+1) words, with 1 extra word reserved for possible carry storage.

#### 42.3.3.4 Programming Entry

All programs are based on the procedure entry address shown in the table below.

Table 42-4 Programming Entry Address

| Procedure Name | Procedure Entry Address | Description                            |
|----------------|-------------------------|----------------------------------------|
| PDBL           | 0x04                    | Elliptic Curve Point Doubling          |
| PADD           | 0x08                    | Elliptic Curve Point Addition          |
| PVER           | 0x0C                    | Verify If Point Lies on Elliptic Curve |
| PMUL           | 0x10                    | Elliptic Curve Point Multiplication    |

| Procedure Name | Procedure Entry Address | Description               |
|----------------|-------------------------|---------------------------|
| MODEXP         | 0x14                    | Modular Exponentiation    |
| MODMUL         | 0x18                    | Modular Multiplication    |
| MODINV         | 0x1C                    | Modular Inversion         |
| MODADD         | 0x20                    | Modular Addition          |
| MODSUB         | 0x24                    | Modular Subtraction       |
| CAL_PRE_MON    | 0x28                    | Montgomery Precomputation |
| IMUL           | 0x2C                    | Big Number Multiplication |

## 42.3.4 Symmetric Encryption Algorithm Module (SKE)

### 42.3.4.1 Algorithm Introduction

The symmetric encryption algorithm module (SKE) contains a symmetric encryption algorithm engine and supports the following functions:

- AES encryption and decryption algorithm
- DES encryption and decryption algorithm
- SM4 encryption and decryption algorithm
- Support 8 operating modes, which are ECB, CBC, CTR, CFB, OFB, GCM, CCM and CMAC, and support data interleaving in GCM, CCM and CMAC modes.
- Support encryption and decryption completion interrupt

The AES encryption and decryption algorithm complies with the description of the algorithm principle in "FIPS PUB 197". The block length is fixed at 128 bits, and the key length supports 128/192/256 bits.

The DES encryption and decryption algorithm complies with the description of the algorithm principle in "FIPS PUB 46-3". The block length is fixed at 64 bits and the key length is 64 bits.

The SM4 encryption and decryption algorithm complies with the description of the algorithm principle in "GMT0002-2012". The block length is fixed at 128 bits and the key length is 128 bits.

### 42.3.4.2 Module Block Diagram

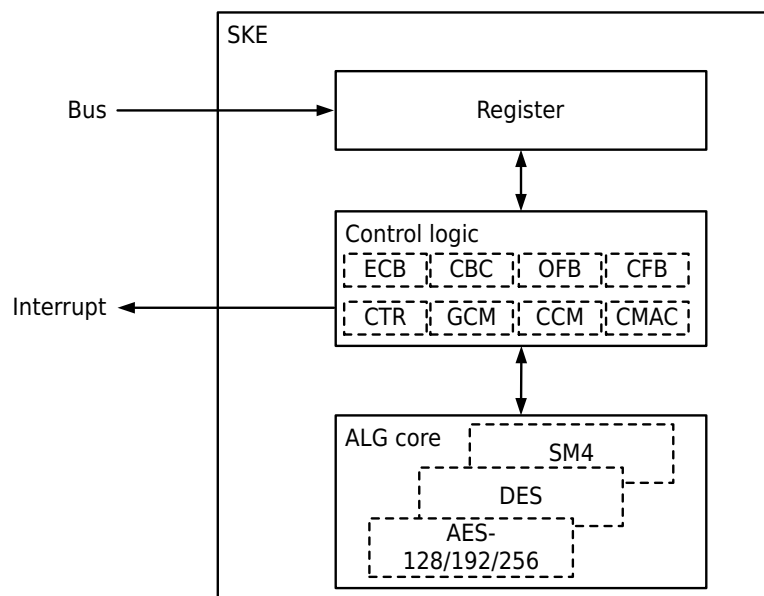


Figure 42-4 SKE System Block Diagram

### 42.3.4.3 Working Mode Description

- ECB: Electronic Code Book mode defined by NIST SP800-38A
- CBC: Cipher Block Chaining mode defined by NIST SP800-38A
- CTR: Counter mode defined by NIST SP800-38A, specifically CTR-32, refer to RFC 3686 standard
- CFB: Cipher Feedback mode defined by NIST SP800-38A, specifically CFB-128
- OFB: Output Feedback mode defined by NIST SP800-38A
- GCM: Galois/Counter mode defined by NIST SP800-38D
- CCM: Counter with CBC-MAC mode defined by NIST SP800-38C
- CMAC: Cipher-based Message Authentication Code mode defined by NIST SP800-38B

### 42.3.4.4 Operation period

Below are the operation period counts for commonly used modes under each algorithm:

Table 42-5 AES Hardware Operation Period Count (Grouping size: 128-bit)

| Key length | ECB | CBC | CTR | CFB | OFB | CMAC | GCM |     |         |     | CCM |     |         |     |
|------------|-----|-----|-----|-----|-----|------|-----|-----|---------|-----|-----|-----|---------|-----|
|            |     |     |     |     |     |      | CFG | AAD | Enc/Dec | TAG | CFG | AAD | Enc/Dec | TAG |
| 128 bits   | 13  | 14  | 14  | 14  | 14  | 18   | 1   | 10  | 18      | 19  | 1   | 18  | 18      | 2   |
| 192 bits   | 15  | 16  | 16  | 16  | 16  | 20   | 1   | 10  | 20      | 21  | 1   | 20  | 20      | 2   |
| 256 bits   | 17  | 18  | 18  | 18  | 18  | 22   | 1   | 10  | 22      | 23  | 1   | 22  | 22      | 2   |

Table 42-6 SM4 Hardware Operation Period Number (Grouping size: 128-bit)

| Key length | ECB | CBC | CTR | CFB | OFB | GCM |     |         |     | CCM |     |         |     |
|------------|-----|-----|-----|-----|-----|-----|-----|---------|-----|-----|-----|---------|-----|
|            |     |     |     |     |     | CFG | AAD | Enc/Dec | TAG | CFG | AAD | Enc/Dec | TAG |
| 128 bits   | 36  | 37  | 37  | 37  | 37  | 1   | 10  | 41      | 42  | 1   | 41  | 41      | 2   |

Table 42-7 DES Hardware Operation Period Count (Grouping size: 64-bit)

| Key length | ECB | CBC | CTR | CFB | OFB |
|------------|-----|-----|-----|-----|-----|
| 64 bits    | 35  | 36  | 36  | 36  | 36  |

#### 42.3.4.5 Operating Procedures

The SKE decryption operation flow is as follows:

1. After power-on reset, SKE is idle, waiting to be called.
2. Determine whether the configuration in the SKE needs to be updated. If so, follow the configuration update procedure; otherwise, proceed directly to the data encryption/decryption process. If the configuration is not updated, the SKE will use the previously stored configuration for operations. Even if there is no switch key (KEY), initial value (IV) and other configurations, the configuration needs to be updated when switching data packets.
  - During Configuration Update:
    - a. Write the AAD length and ciphertext length to SKE\_AADx/SKE\_CLENx (applicable only in GCM/CCM mode).
    - b. Write the required configuration to SKE\_CFG, select encryption or decryption, and ensure that SKE\_CFG.UP\_CFG is 1.
    - c. Write the key to SKE\_KEYx. It is not necessary to input the key every time. If the key has already been written previously, there is no need to re-enter the same key. The SKE will use the previously stored key for computation.
    - d. The initial value is written through SKE\_IVx. In segmented interleaving for CCM, GCM, or CMAC modes, if operating in a middle or final segment, the intermediate values SKE\_MID\_MACx and SKE\_MID\_IVx must be provided.
    - e. Write 1 to SKE\_CTRL to trigger SKE to perform calculations.
    - f. Poll the flag bit corresponding to SKE\_SR2 until the flag bit is 1.
    - g. When the flag bit in SKE\_SR2 is detected to be 1, a write operation is performed on SKE\_SR2 to clear it.
    - h. Write 0 to SKE\_CFG.UP\_CFG; at this point, the configuration update process ends.
  - When encrypting and decrypting data:
    - a. (Optional) When AAD is input (only applicable in GCM/CCM mode), the following steps need to be performed:
      - 1) Make sure SKE\_DIN\_CR.LAST=0b0 and input the first set of AAD by writing to register SKE\_DINx.
      - 2) Write 1 to SKE\_CTRL to trigger SKE to perform calculations.
      - 3) Poll the flag bit corresponding to SKE\_SR2 until the flag bit is 1.
      - 4) When the flag bit in SKE\_SR2 is detected to be 1, a write operation is performed on SKE\_SR2 to clear it.
      - 5) Repeat steps 1) to 4). When inputting the last set of AAD, write SKE\_DIN\_CR.LAST=0b1 and then repeat steps 1) to 4).
    - b. Make sure SKE\_DIN\_CR.LAST=0b0.
    - c. By writing register SKE\_DINx, input the next set of data to be encrypted/decrypted.
    - d. Write 1 to SKE\_CTRL.START to trigger SKE to perform calculations.
    - e. Poll the SKE\_SR2.DONE flag until it is 1.
    - f. When SKE\_SR2.DONE is detected to be 1, the flag is cleared.
    - g. Repeat steps c-f. Note: If you are entering the last set of data and it is GCM/CCM/CMAC mode, you need to set SKE\_DIN\_CR.LAST=0b1 before entering the data.
    - h. After the last set of data encryption/decryption is completed, make sure SKE\_DIN\_CR.LAST=0b0.

- i. Read SKE\_DOUT to obtain the operation result; if it is the initial segment or middle segment of segmented interleaving, SKE\_MID\_MACx and SKE\_MID\_IVx need to be read; if TAG needs to be generated, repeat steps b-e again and read SKE\_DOUTx.

At this point, SKE has completed a complete call and can loop the above steps to make multiple calls.

#### 42.3.4.6 Register Address Mapping

SKE's SKE\_KEYx, SKE\_IVx, SKE\_DINx, SKE\_DOUTx, SKE\_MID\_IVx, and SKE\_MID\_MACx registers all use little-endian storage. The SKE\_KEYx register is used as an example as follows. Other registers are similar:

- DES 64-bit key length: 0x00, 0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07

| Register | Write Value |
|----------|-------------|
| SKE_KEY1 | 0x03020100  |
| SKE_KEY2 | 0x07060504  |

- AES 128-bit key length: 0x00, 0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07, 0x08, 0x09, 0x0A, 0x0B, 0x0C, 0x0D, 0x0E, 0x0F

| Register | Write Value |
|----------|-------------|
| SKE_KEY1 | 0x03020100  |
| SKE_KEY2 | 0x07060504  |
| SKE_KEY3 | 0x0B0A0908  |
| SKE_KEY4 | 0x0F0E0D0C  |

- AES 192-bit key length: 0x00, 0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07, 0x08, 0x09, 0x0A, 0x0B, 0x0C, 0x0D, 0x0E, 0x0F, 0x10, 0x11, 0x12, 0x13, 0x14, 0x15, 0x16, 0x17

| Register | Write Value |
|----------|-------------|
| SKE_KEY1 | 0x03020100  |
| SKE_KEY2 | 0x07060504  |
| SKE_KEY3 | 0x0B0A0908  |
| SKE_KEY4 | 0x0F0E0D0C  |
| SKE_KEY5 | 0x13121110  |
| SKE_KEY6 | 0x17161514  |

- AES 256-bit key length: 0x00, 0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07, 0x08, 0x09, 0x0A, 0x0B, 0x0C, 0x0D, 0x0E, 0x0F, 0x10, 0x11, 0x12, 0x13, 0x14, 0x15, 0x16, 0x17, 0x18, 0x19, 0x1A, 0x1B, 0x1C, 0x1D, 0x1E, 0x1F

| Register | Write Value |
|----------|-------------|
| SKE_KEY1 | 0x03020100  |
| SKE_KEY2 | 0x07060504  |
| SKE_KEY3 | 0x0B0A0908  |
| SKE_KEY4 | 0x0F0E0D0C  |
| SKE_KEY5 | 0x13121110  |



| Register | Write Value |
|----------|-------------|
| SKE_KEY6 | 0x17161614  |
| SKE_KEY7 | 0x1B1A1918  |
| SKE_KEY8 | 0x1F1E1D1C  |

#### 42.3.4.7 Data Example

2 packets of 256-bit plaintext:

0x6BC1BEE2 2E409F96 E93D7E11 7393172A AE2D8A57 1E03AC9C 9EB76FAC 45AF8E51

128-bit key:

0x2B7E1516 28AED2A6 ABF71588 09CF4F3C

256-bit ciphertext:

0x3AD77BB4 0D7A3660 A89ECAf3 2466EF97 F5D3D585 03B9699D E785895A 96FDBAAF

Table 42-12 128-bit AES ECB Mode Operation Register Example

| Before Encryption                    |                         |          |                         |
|--------------------------------------|-------------------------|----------|-------------------------|
| Register                             | Write Value (key)       | Register | Write Value (Plaintext) |
| SKE_KEY1                             | 0x16157E2B              | SKE_DIN1 | 0xE2BEC16B              |
| SKE_KEY2                             | 0xA6D2AE28              | SKE_DIN2 | 0x969F402E              |
| SKE_KEY3                             | 0x8815F7AB              | SKE_DIN3 | 0x117E3DE9              |
| SKE_KEY4                             | 0x3C4FCF09              | SKE_DIN4 | 0x2A179373              |
| After the First Round of Encryption  |                         |          |                         |
| Register                             | Read Value (Ciphertext) | Register | Write Value (Plaintext) |
| SKE_DOUT1                            | 0xB47BD73A              | SKE_DIN1 | 0x578A2DAE              |
| SKE_DOUT2                            | 0x60367A0D              | SKE_DIN2 | 0x9CAC031E              |
| SKE_DOUT3                            | 0xF3CA9EA8              | SKE_DIN3 | 0xAC6FB79E              |
| SKE_DOUT4                            | 0x97EF6624              | SKE_DIN4 | 0x518EAF45              |
| After the Second Round of Encryption |                         |          |                         |
| Register                             | Read Value (Ciphertext) | Register | Write Value (Plaintext) |
| SKE_DOUT1                            | 0x85D5D3F5              | -        | -                       |
| SKE_DOUT2                            | 0x9D69B903              | -        | -                       |
| SKE_DOUT3                            | 0x5A8985E7              | -        | -                       |
| SKE_DOUT4                            | 0xAFBAFD96              | -        | -                       |

#### 42.3.4.8 Interrupt

SKE contains 1 interrupt. When the encryption and decryption operation of SKE is completed, the SKE\_SR2.SKE\_DONE flag is set to 1. If SKE\_CFG.IRQEN=1 at this time, an interrupt signal is generated. The SKE\_SR2.SKE\_DONE flag bit can be cleared by software.

## 42.4 Register Description

### 42.4.1 Register List

HASH base address: 0x40008400

Table 42-13 Secure Hash Algorithm (HASH) Register Offset

| Offset            | Register                 | Description                     | Width |
|-------------------|--------------------------|---------------------------------|-------|
| 0x0000            | <a href="#">HASH_CR</a>  | HASH Control Register           | 32    |
| 0x0010+0x4*(7-n)  | <a href="#">HASH_HRn</a> | HASH Summary Register n (n=0-7) | 32    |
| 0x0040+0x4*(15-n) | <a href="#">HASH_DRn</a> | HASH Data Register n (n=0-15)   | 32    |

TRNG base address: 0x40042000

Table 42-14 True Random Number Generator (TRNG) Register Offset

| Offset       | Register                 | Description                  | Width |
|--------------|--------------------------|------------------------------|-------|
| 0x0000       | <a href="#">TRNG_CR</a>  | TRNG Control Register        | 32    |
| 0x0004       | <a href="#">TRNG_MR</a>  | TRNG Mode Register           | 32    |
| 0x000C+0x4*n | <a href="#">TRNG_DRn</a> | TRNG Data Register n (n=0-1) | 32    |

PKE base address: 0x4000B000

Table 42-15 Public Key Engine (PKE) Register Offset

| Offset        | Register                      | Description                          | Width |
|---------------|-------------------------------|--------------------------------------|-------|
| 0x1000        | <a href="#">PKE_CTRL</a>      | PKE Control Register                 | 32    |
| 0x1004        | <a href="#">PKE_CFG</a>       | PKE Configuration Register           | 32    |
| 0x1008        | <a href="#">PKE_MC_PTR</a>    | PKE Procedure Entry Register         | 32    |
| 0x100C        | <a href="#">PKE_RISR</a>      | PKE Interrupt Source Register        | 32    |
| 0x1010        | <a href="#">PKE_IMCR</a>      | PKE Interrupt Enable Register        | 32    |
| 0x1014        | <a href="#">PKE_MISR</a>      | PKE Interrupt Output Register        | 32    |
| 0x1024        | <a href="#">PKE_RT_CODE</a>   | PKE Return Log Register              | 32    |
| 0x1050        | <a href="#">PKE_EXE_CFG</a>   | PKE Operation Configuration Register | 32    |
| 0x10FC        | <a href="#">VERSION</a>       | PKE Version Register                 | 32    |
| 0x1400-0x1E10 | <a href="#">PKE_MEM_RAM0</a>  | PKE Operation Operand RAM0           | 32    |
| 0x2000-0x2A10 | <a href="#">PKE_MEM_RAM1</a>  | PKE Operation Operand RAM1           | 32    |
| 0x0000        | <a href="#">PKE_RAM0_CKCR</a> | PKE RAM0 Check Control Register      | 32    |
| 0x0004        | <a href="#">PKE_RAM0_CKSR</a> | PKE RAM0 Check Status Register       | 32    |

| Offset | Register                            | Description                                    | Width |
|--------|-------------------------------------|------------------------------------------------|-------|
| 0x0008 | <a href="#">PKE_RAM0_EIEN</a>       | PKE RAM0 Error Injection Enable Register       | 32    |
| 0x000C | <a href="#">PKE_RAM0_EIBIT0</a>     | PKE RAM0 Error Injection Bit Enable Register 0 | 32    |
| 0x0010 | <a href="#">PKE_RAM0_EIBIT1</a>     | PKE RAM0 Error Injection Bit Enable Register 1 | 32    |
| 0x0014 | <a href="#">PKE_RAM0_ECCERRADDR</a> | PKE RAM0 Error Address Register                | 32    |
| 0x0020 | <a href="#">PKE_RAM1_CKCR</a>       | PKE RAM1 Check Control Register                | 32    |
| 0x0024 | <a href="#">PKE_RAM1_CKSR</a>       | PKE RAM1 Check Status Register                 | 32    |
| 0x0028 | <a href="#">PKE_RAM1_EIEN</a>       | PKE RAM1 Error Injection Enable Register       | 32    |
| 0x002C | <a href="#">PKE_RAM1_EIBIT0</a>     | PKE RAM1 Error Injection Bit Enable Register 0 | 32    |
| 0x0030 | <a href="#">PKE_RAM1_EIBIT1</a>     | PKE RAM1 Error Injection Bit Enable Register 1 | 32    |
| 0x0034 | <a href="#">PKE_RAM1_ECCERRADDR</a> | PKE RAM1 Error Address Register                | 32    |

SKE base address: 0x4000E000

Table 42-16 Symmetric Encryption Algorithm Module (SKE) Register Offset

| Offset           | Register                       | Description                               | Width |
|------------------|--------------------------------|-------------------------------------------|-------|
| 0x0000           | <a href="#">SKE_CTRL</a>       | SKE Control Register                      | 32    |
| 0x0004           | <a href="#">SKE_CFG</a>        | SKE Configuration Register                | 32    |
| 0x0008           | <a href="#">SKE_SR1</a>        | SKE Status Register 1                     | 32    |
| 0x000C           | <a href="#">SKE_SR2</a>        | SKE Status Register 2                     | 32    |
| 0x0010+0x4*(x-1) | <a href="#">SKE_KEYx</a>       | SKE Key Register x (x=1-8)                | 32    |
| 0x0060+0x4*(x-1) | <a href="#">SKE_AADx</a>       | SKE AAD Length Register x (x=1-2)         | 32    |
| 0x0068+0x4*(x-1) | <a href="#">SKE_CLENx</a>      | SKE Data Length Register x (x=1-2)        | 32    |
| 0x0070+0x4*(x-1) | <a href="#">SKE_IVx</a>        | SKE Initial Value Register x (x=1-4)      | 32    |
| 0x0080           | <a href="#">SKE_DIN_CR</a>     | SKE Data Flag Register                    | 32    |
| 0x0090+0x4*(x-1) | <a href="#">SKE_DINx</a>       | SKE Data Input Register x (x=1-4)         | 32    |
| 0x00B0+0x4*(x-1) | <a href="#">SKE_DOUTx</a>      | SKE Data Output Register x (x=1-4)        | 32    |
| 0x00FC           | <a href="#">SKE_VERSION</a>    | SKE Version Register                      | 32    |
| 0x01A0+0x4*(x-1) | <a href="#">SKE_MID_IVx</a>    | SKE Intermediate Value Register x (x=1-4) | 32    |
| 0x01B0+0x4*(x-1) | <a href="#">SKE_MID_MACx</a>   | SKE Intermediate MAC Register x (x=1-4)   | 32    |
| 0x0060+0x4*(x-1) | <a href="#">SKE_TOTAL_AADx</a> | SKE Total AAD Length Register x (x=1-2)   | 32    |

| Offset           | Register                        | Description                              | Width |
|------------------|---------------------------------|------------------------------------------|-------|
| 0x0068+0x4*(x-1) | <a href="#">SKE_TOTAL_CLENx</a> | SKE Total Data Length Register x (x=1-2) | 32    |

## 42.4.2 HASH Register

### 42.4.2.1 HASH Control Register (HASH\_CR)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |          |    |    |      |          |    |          |   |         |   |      |   |          |          |      |    |         |  |          |      |          |  |         |    |       |  |  |  |  |    |      |  |  |  |  |  |  |    |          |  |    |         |  |  |  |  |    |       |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----------|----|----|------|----------|----|----------|---|---------|---|------|---|----------|----------|------|----|---------|--|----------|------|----------|--|---------|----|-------|--|--|--|--|----|------|--|--|--|--|--|--|----|----------|--|----|---------|--|--|--|--|----|-------|
| 0x0000 | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15       | 14 | 13 | 12   | 11       | 10 | 9        | 8 | 7       | 6 | 5    | 4 | 3        | 2        | 1    | 0  |         |  |          |      |          |  |         |    |       |  |  |  |  |    |      |  |  |  |  |  |  |    |          |  |    |         |  |  |  |  |    |       |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |          |    |    |      |          |    |          |   |         |   |      |   |          |          |      |    |         |  |          |      |          |  |         |    |       |  |  |  |  |    |      |  |  |  |  |  |  |    |          |  |    |         |  |  |  |  |    |       |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | DATATYPE |    |    |      | Reserved |    | HMAC_END |   | CYC_END |   | BUSY |   | Reserved |          | LKEY |    | MODE    |  | Reserved |      | KMSG_END |  | FST_GRP |    | START |  |  |  |  |    |      |  |  |  |  |  |  |    |          |  |    |         |  |  |  |  |    |       |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RW       |    | RW | HEIE |          | RW | HCIE     |   |         |   |      |   | RW       | HMAC_END |      | RW | CYC_END |  | RO       | BUSY |          |  |         | RW | LKEY  |  |  |  |  | RW | MODE |  |  |  |  |  |  | RW | KMSG_END |  | RW | FST_GRP |  |  |  |  | RW | START |

| Bit   | Name     | Function                                                          | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                            | Access |
|-------|----------|-------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:18 | Reserved | Reserved                                                          | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | -      |
| 17:16 | DATATYPE | Input Data Sequence                                               | <ul style="list-style-type: none"> <li>0b00: Reverse byte sequence within a word</li> <li>0b01: Reverse sequence by half word within a word</li> <li>0b10: Bitwise reverse sequence within a word</li> <li>0b11: Keep the original data sequence</li> </ul>                                                                                                                                                                                                            | RW     |
| 15    | HEIE     | Interrupt enable for completion of each data group operation      | <ul style="list-style-type: none"> <li>0b0: Interrupt disabled upon completion of each data group operation</li> <li>0b1: Interrupt enabled upon completion of each data group operation</li> </ul>                                                                                                                                                                                                                                                                    | RW     |
| 14    | HCIE     | Interrupt enable upon completion of all key or message operations | <ul style="list-style-type: none"> <li>0b0: Interrupt disabled upon completion of all key or message operations</li> <li>0b1: Interrupt enabled upon completion of all key or message operations</li> </ul>                                                                                                                                                                                                                                                            | RW     |
| 13:11 | Reserved | Reserved                                                          | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | -      |
| 10    | HMAC_END | HMAC Operation Complete Flag                                      | <ul style="list-style-type: none"> <li>0b0: HMAC operation not completed</li> <li>0b1: HMAC operation completed</li> </ul> <div>  <b>Note</b><br/>           When this flag is set to '1', writing '0' will clear it. Note: Software should clear this flag (write '0') promptly after confirming it, otherwise the flag will remain set to '1'.         </div>                     | RW     |
| 9     | CYC_END  | Key or Message Operation Complete Flag                            | <ul style="list-style-type: none"> <li>0b0: Key or message operation not completed</li> <li>0b1: Key or message operation completed</li> </ul> <div>  <b>Note</b><br/>           When this flag is set to '1', writing '0' will clear it. Note: Software should clear this flag (write '0') promptly after confirming it, otherwise the flag will remain set to '1'.         </div> | RW     |

| Bit | Name            | Function                      | Description                                                                                                                                                                                                                                                                                   | Access |
|-----|-----------------|-------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 8   | BUSY            | Conversion Flag               | <ul style="list-style-type: none"> <li>0b0: HASH operation is idle</li> <li>0b1: HASH operation in progress</li> </ul>                                                                                                                                                                        | RO     |
| 7   | <i>Reserved</i> | <i>Reserved</i>               | -                                                                                                                                                                                                                                                                                             | -      |
| 6   | LKEY            | Long Key Selection            | <ul style="list-style-type: none"> <li>0b0: Short key (<math>\leq 64</math> bytes)</li> <li>0b1: Long key (<math>&gt; 64</math> bytes)</li> </ul> This bit selects between a short key ( $\leq 64$ bytes) and a long key ( $> 64$ bytes) in HMAC mode. Valid only when MODE=0b01 (HMAC mode). | RW     |
| 5:4 | MODE            | Operating Mode                | <ul style="list-style-type: none"> <li>0b00: SHA-256 operation mode</li> <li>0b01: HMAC operation mode</li> <li>0b1X: Setting prohibited</li> </ul>                                                                                                                                           | RW     |
| 3   | <i>Reserved</i> | <i>Reserved</i>               | -                                                                                                                                                                                                                                                                                             | -      |
| 2   | KMSG_END        | Last Group of Key or Message  | <ul style="list-style-type: none"> <li>0b0: Not the last group of key or message</li> <li>0b1: Last group of key or message</li> </ul> This bit is automatically cleared to '0' after the last group of key or message has been processed.                                                    | RW     |
| 1   | FST_GRP         | First Group of Key or Message | <ul style="list-style-type: none"> <li>0b0: Not the first group of key or message</li> <li>0b1: First group of key or message</li> </ul> This bit is automatically cleared to '0' after the first group of key or message has been processed.                                                 | RW     |
| 0   | START           | Start                         | 0b0: The module operation is finished or not started<br>0b1: Start the module for operation                                                                                                                                                                                                   | RW     |

**Note**

- The operation method of the START bit is: after the software writes 1 to this bit, the module will start to run; after the end of this run, the hardware will automatically clear this bit to 0; if the software queries this bit to 0, it means that this run is completed.
- The write operation to this register can only be performed when the module is not in the operation state (that is, when the START bit is 0), otherwise the hardware will automatically ignore the write operation. Read operations are not subject to this restriction.
- If you want to change the sequence of input data, you need to configure DATATYPE before inputting data.

#### 42.4.2.2 HASH Summary Register n (HASH\_HRn) (n=0-7)

The 8 HASH summary registers (HASH\_HR0-HASH\_HR7) correspond to 256-bit HASH summary. HASH\_HR0 corresponds to the summary Bit31-Bit0, HASH\_HR1 corresponds to the summary Bit63-Bit32, and so on.

Offset: 0x0010+0x4\*(7-n)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| '      | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | HASH         |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access | RO           |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit  | Name | Function      | Description                                                                                                  | Access |
|------|------|---------------|--------------------------------------------------------------------------------------------------------------|--------|
| 31:0 | HASH | Summary Value | After the operation of the module is completed, the message summary can be obtained by reading this register | RO     |



#### Note

- This register is spliced by 8 32-bit registers. During the access, 8 32-bit registers are operated in turn, and the 32-bit register corresponding to the low address stores the high word of the message summary.
- The reading of this register can only be carried out when the module is not in the operation state (HASH\_CR.START=0), otherwise the reading of this register will get all 0.

42.4.2.3 HASH Data Register n (HASH\_DRn) (n=0-15)

16 HASH data registers (HASH\_DR0-HASH\_DR15) correspond to 512-bit HASH data. HASH\_DR0 corresponds to data Bit31-Bit0, HASH\_DR1 corresponds to data Bit63-Bit32, and so on.

Offset: 0x0040+0x4\*(15-n)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| '      | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | DATA         |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access | RW           |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit  | Name | Function      | Description                                    | Access |
|------|------|---------------|------------------------------------------------|--------|
| 31:0 | DATA | Data register | Used to write messages before block operations | RW     |

 Note

- This register is spliced by 16 32-bit registers. During the access, 16 32-bit registers are operated in turn, and the 32-bit register corresponding to the low address stores the high word of the data.
- Writing to this register can only be performed when the module is not in the operation state (HASH\_CR.START=0), otherwise the hardware will automatically ignore the writing operation to this register.
- When this module is in operation state (HASH\_CR.START=1), reading this register will always result in all 0. In the non-operation state (HASH\_CR.START=0), the data read from this register (the data is the intermediate result of the calculation process) is arranged in the format of HASH\_CR.DATATYPE=0.



## 42.4.3 TRNG Register

### 42.4.3.1 TRNG Control Register (TRNG\_CR)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |     |   |    |  |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|-----|---|----|--|
| 0x0000 | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1   | 0 |    |  |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |     |   |    |  |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   | RUN |   | EN |  |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   | RW  |   | RW |  |

| Bit  | Name     | Function                       | Description                                                                                                                                                                                                                                                                                                                                                                                      | Access |
|------|----------|--------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:2 | Reserved | Reserved                       | -                                                                                                                                                                                                                                                                                                                                                                                                | -      |
| 1    | RUN      | Random number operation starts | <ul style="list-style-type: none"> <li>0b0: Random number generation stopped</li> <li>0b1: Random number generation started</li> </ul> <div>  <b>Note</b><br/>           The software writes 1 to generate a new 64-bit random number; after the operation is completed, the hardware clears it.         </div> | RW     |
| 0    | EN       | Analog oscillator enabled      | <ul style="list-style-type: none"> <li>0b0: Turn off the analog random number generator circuit</li> <li>0b1: Turn on the analog random number generator circuit</li> </ul>                                                                                                                                                                                                                      | RW     |

## 42.4.3.2 TRNG Mode Register (TRNG\_MR)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |     |   |          |      |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|-----|---|----------|------|---|---|
| 0x0004 | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5   | 4 | 3        | 2    | 1 | 0 |
| Reset  | 0x00000012   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |     |   |          |      |   |   |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   | CNT |   | Reserved | LOAD |   |   |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   | RW  |   |          |      |   |   |

| Bit  | Name     | Function                | Description                                                                                                                                                                                                                                                                                                      | Access |
|------|----------|-------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:5 | Reserved | Reserved                | -                                                                                                                                                                                                                                                                                                                | -      |
| 4:2  | CNT      | Shift count control bit | When capturing random noise, the number of shift control bits <ul style="list-style-type: none"> <li>0b000-0b010: Setting prohibited</li> <li>0b011: Shift 32 times</li> <li>0b100: Shift 64 times</li> <li>0b101: Shift 128 times</li> <li>0b110: Shift 256 times</li> <li>0b111: Setting prohibited</li> </ul> | RW     |
| 1    | Reserved | Reserved                | Reads as 1, Write as 1                                                                                                                                                                                                                                                                                           | -      |
| 0    | LOAD     | Load control bit        | Whether the data register is loaded with a new initial value from the random number generator before the random number is generated <ul style="list-style-type: none"> <li>0b0: Do not load a new initialization value</li> <li>0b1: Load a new initialization value</li> </ul>                                  | RW     |

42.4.3.3 TRNG Data Register n (TRNG\_DRn) (n=0-1)

2 TRNG data registers TRNG\_DR0-TRNG\_DR1 correspond to 64-bit TRNG data. TRNG\_DR0 corresponds to data Bit31-Bit0, TRNG\_DR1 corresponds to data Bit63-Bit32.

Offset: 0x000C+0x4\*n

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| '      | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0XXXXXXXXX   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | DATA         |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access | RO           |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit  | Name | Function      | Description   | Access |
|------|------|---------------|---------------|--------|
| 31:0 | DATA | Random number | Random number | RO     |



Note

The TRNG\_DR register is composed of two 32-bit registers.

## 42.4.4 PKE Register

### 42.4.4.1 PKE Control Register (PKE\_CTRL)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |       |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|-------|
| 0x1000 | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0     |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |       |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   | START |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   | RW    |

| Bit  | Name     | Function          | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | Access |
|------|----------|-------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:1 | Reserved | Reserved          | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | -      |
| 0    | START    | Start calculation | Start PKE calculation <ul style="list-style-type: none"> <li>0b0: PKE calculation has not started</li> <li>0b1: PKE will start calculation in the next clock period</li> </ul> <div>  <b>Note</b><br/>           Writing 0 has no effect and will not change the current value of this bit; the hardware will automatically clear it after the PKE calculation is completed.         </div> | RW     |

## 42.4.4.2 PKE Configuration Register (PKE\_CFG)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |            |    |          |    |               |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|------------|----|----------|----|---------------|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0x1004 | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18         | 17 | 16       | 15 | 14            | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00020100   |    |    |    |    |    |    |    |    |    |    |    |    |            |    |          |    |               |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    | BASE_RADIX |    | Reserved |    | PARTIAL_RADIX |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    | RW         |    |          |    | RW            |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit   | Name          | Function            | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | Access |
|-------|---------------|---------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:19 | Reserved      | Reserved            | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | -      |
| 18:16 | BASE_RADIX    | Basic Configuration | <p>This field indicates the bit width base for the operation. At the same time, the base also indicates the space required to store the operand in the data RAM:</p> <ul style="list-style-type: none"> <li>● 2: 256-bit</li> <li>● 3: 512-bit</li> <li>● 4: 1024-bit</li> <li>● 5: 2048-bit</li> <li>● 6: 4096-bit</li> <li>● Others: Setting prohibited</li> </ul> <div>  <b>Note</b> <ul style="list-style-type: none"> <li>● For RSA modulus operations, the value of this field should be 4, 5, or 6;</li> <li>● For ECC point operations, the value of this field should be 2, 3, or 4.</li> </ul> </div> | RW     |
| 15:13 | Reserved      | Reserved            | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | -      |
| 12:0  | PARTIAL_RADIX | Other Configuration | <p>PARTIAL_RADIX determines the actual bit width that the operation needs to use during the calculation process. The value of this field indicates the number of bits, and the bit width of the operand is PARTIAL_RADIX bits.</p> <p>If SECP521R1 operation is to be performed, BASE_RADIX needs to be configured to 4 and PARTIAL_RADIX needs to be configured to 512.</p>                                                                                                                                                                                                                                                                                                                       | RW     |

## 42.4.4.3 PKE Procedure Entry Register (PKE\_MC\_PTR)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |      |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|------|----|---|---|---|---|---|---|---|---|---|---|
| 0x1008 | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11   | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |      |    |   |   |   |   |   |   |   |   |   |   |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | ADDR |    |   |   |   |   |   |   |   |   |   |   |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RW   |    |   |   |   |   |   |   |   |   |   |   |

| Bit   | Name     | Function                | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | Access |
|-------|----------|-------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:12 | Reserved | Reserved                | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | -      |
| 11:0  | ADDR     | Procedure entry address | PKE microcode execution entry<br> <b>Note</b><br>This register can only be overwritten when PKE is not working. Any write operation when PKE is working will be ignored.<br><br>During the operation of PKE, this field will also be updated in real time; it always points to the address of the next instruction to be executed.<br><br>It should be noted that instructions are word-aligned. Therefore, the lowest 2 bits of this field are 0. | RW     |

## 42.4.4.4 PKE Interrupt Source Register (PKE\_RISR)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |          |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|----------|
| 0x100C | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0        |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |          |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   | CORE_RIS |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   | RW       |

| Bit  | Name     | Function                  | Description                                                                                                                                                                                                                                                                            | Access |
|------|----------|---------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:1 | Reserved | Reserved                  | -                                                                                                                                                                                                                                                                                      | -      |
| 0    | CORE_RIS | Operation completion flag | Write 0 to clear the flag, write 1 has no effect<br>Read:<br><ul style="list-style-type: none"> <li>0b0: PKE CPU mode has not completed the operation</li> <li>0b1: PKE CPU mode has completed the operation. If PKE_IMCR.CORE_IRQEN is enabled, the interrupt is generated</li> </ul> | RW     |

## 42.4.4.5 PKE Interrupt Enable Register (PKE\_IMCR)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |            |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|------------|
| 0x1010 | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |            |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |            |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   | CORE_IROEN |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   | RW         |

| Bit  | Name       | Function                  | Description                                                                                                                                                     | Access |
|------|------------|---------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:1 | Reserved   | Reserved                  | -                                                                                                                                                               | -      |
| 0    | CORE_IRQEN | CPU mode interrupt enable | <ul style="list-style-type: none"> <li>0b0: Disable PKE to generate interrupt in CPU mode</li> <li>0b1: Enable PKE to generate interrupt in CPU mode</li> </ul> | RW     |

## 42.4.4.6 PKE Interrupt Output Register (PKE\_MISR)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |         |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---------|
| 0x1014 | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0       |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |         |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   | CORE MI |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   | RO      |

| Bit  | Name     | Function                | Description                                                                                                                                                | Access |
|------|----------|-------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:1 | Reserved | Reserved                | -                                                                                                                                                          | -      |
| 0    | CORE_MI  | CPU mode interrupt flag | <ul style="list-style-type: none"> <li>0b0: PKE does not generate an interrupt in CPU mode</li> <li>0b1: PKE generates an interrupt in CPU mode</li> </ul> | RO     |

## 42.4.4.7 PKE Return Log Register (PKE\_RT\_CODE)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |          |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|----------|---|---|---|
| 0x1024 | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3        | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |          |   |   |   |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   | STOP_LOG |   |   |   |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   | RO       |   |   |   |

| Bit  | Name     | Function | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | Access |
|------|----------|----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:4 | Reserved | Reserved | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | -      |
| 3:0  | STOP_LOG | Stop log | <p>Used to indicate the reason why PKE stopped</p> <p>If PKE stopped because the operation was completed, the value of this field is 0. If the value of this field is non-zero, the PKE operation was not completed, some exceptions were encountered, and external processing is required, and the result is unavailable</p> <ul style="list-style-type: none"> <li>● 0: Normal stop</li> <li>● 2: No active modular inverse result</li> <li>● 3: The point is not on the curve (CTRL.CMD: PVER)</li> <li>● Other: Reserved</li> </ul> | RO     |



## 42.4.4.8 PKE Operation Configuration Register (PKE\_EXE\_CFG)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |      |    |      |    |         |    |         |    |         |  |         |  |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|------|----|------|----|---------|----|---------|----|---------|--|---------|--|
| 0x1050 | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5    | 4  | 3    | 2  | 1       | 0  |         |    |         |  |         |  |
| Reset  | 0x00000015   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |      |    |      |    |         |    |         |    |         |  |         |  |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   | OMON |    | OAFF |    | IMON_R1 |    | IAFF_R1 |    | IMON_R0 |  | IAFF_R0 |  |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   | RO   | RW | RW   | RW | RW      | RW | RW      | RW |         |  |         |  |

| Bit  | Name     | Function                                    | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | Access |
|------|----------|---------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:6 | Reserved | Reserved                                    | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | -      |
| 5    | OMON     | Enable Montgomery output                    | <ul style="list-style-type: none"> <li>0b0: Output is in normal format</li> <li>0b1: Output is in Montgomery format</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                           | RO     |
| 4    | OAFF     | Enable output in affine coordinate system   | <ul style="list-style-type: none"> <li>0b0: The output point is a point on the Jacobi coordinate system</li> <li>0b1: The output point is a point on the affine coordinate system</li> </ul> <div>  <b>Note</b><br/>           This bit setting is only valid for ECC operation.         </div>                                                                                                                                                                                        | RW     |
| 3    | IMON_R1  | Enable Montgomery form input R1             | <ul style="list-style-type: none"> <li>0b0: Input data is in normal form; when modular multiplication is involved, if this bit is low, the number in its scope will be converted to Montgomery form before the operation is performed</li> <li>0b1: Input data is in Montgomery form</li> </ul>                                                                                                                                                                                                                                                                          | RW     |
| 2    | IAFF_R1  | Enable input R1 in affine coordinate system | <ul style="list-style-type: none"> <li>0b0: The input point is a point in the Jacobian coordinate system; when it comes to modular multiplication, if this bit is low, the point in its scope will be converted to a point in the Jacobian coordinate system before the operation is performed</li> <li>0b1: The input point is a point in the affine coordinate system</li> </ul> <div>  <b>Note</b><br/>           This bit setting is only valid for ECC operation.         </div> | RW     |
| 1    | IMON_R0  | Enable Montgomery form input R0             | <ul style="list-style-type: none"> <li>0b0: Input data is in normal form; when modular multiplication is involved, if this bit is low, the number in its scope will be converted to Montgomery form before the operation is performed</li> <li>0b1: Input data is in Montgomery form</li> </ul>                                                                                                                                                                                                                                                                          | RW     |

| Bit | Name    | Function                                    | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | Access |
|-----|---------|---------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 0   | IAFF_R0 | Enable input R0 in affine coordinate system | <ul style="list-style-type: none"><li>● 0b0: The input point is a point in the Jacobian coordinate system; when it comes to modular multiplication, if this bit is low, the point in its scope will be converted to a point in the Jacobian coordinate system before the operation is performed</li><li>● 0b1: The input point is a point in the affine coordinate system</li></ul> <div> <b>Note</b><br/>This bit setting is only valid for ECC operation.</div> | RW     |

## 42.4.4.9 PKE Version Register (VERSION)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |          |    |    |    |    |    |   |   |     |   |   |   |     |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----------|----|----|----|----|----|---|---|-----|---|---|---|-----|---|---|---|
| 0x10FC | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15       | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7   | 6 | 5 | 4 | 3   | 2 | 1 | 0 |
| Reset  | 0xC4060010   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |          |    |    |    |    |    |   |   |     |   |   |   |     |   |   |   |
| Name   | PROJECT      |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | Reserved |    |    |    |    |    |   |   | MAR |   |   |   | MIR |   |   |   |
|        |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |          |    |    |    |    |    |   |   |     |   |   |   |     |   |   |   |
| Access | RO           |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |          |    |    |    |    |    |   |   | RO  |   |   |   | RO  |   |   |   |

| Bit   | Name     | Function          | Description | Access |
|-------|----------|-------------------|-------------|--------|
| 31:16 | PROJECT  | Project No.       | -           | RO     |
| 15:8  | Reserved | Reserved          | -           | -      |
| 7:4   | MAR      | Major Version No. | -           | RO     |
| 3:0   | MIR      | Minor Version No. | -           | RO     |

#### 42.4.4.10 PKE Operation Operand RAM0 (PKE\_MEM\_RAM0)

This register is used to store operation data. Depending on the operation mode and budget length, registers with different offset addresses are selected for operation. For details, see "[Programming Model](#)".

Offset: 0x1400-0x1E10

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |  |  |  |  |  |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|--|--|--|--|--|--|
| -      | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |  |  |  |  |  |  |
| Reset  | 0xFFFFFFFF   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |  |  |  |  |  |
| Name   | DATA         |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |  |  |  |  |  |
| Access | RW           |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |  |  |  |  |  |

| Bit  | Name | Function         | Description                                   | Access |
|------|------|------------------|-----------------------------------------------|--------|
| 31:0 | DATA | Operational Data | This field is used to store calculation data. | RW     |

#### 42.4.4.11 PKE Operation Operand RAM1 (PKE\_MEM\_RAM1)

This register is used to store operation data. Depending on the operation mode and budget length, registers with different offset addresses are selected for operation. For details, see "[Programming Model](#)".

Offset: 0x2000-0x2A10

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |  |  |  |  |  |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|--|--|--|--|--|--|
| -      | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |  |  |  |  |  |  |
| Reset  | 0xFFFFFFFF   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |  |  |  |  |  |
| Name   | DATA         |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |  |  |  |  |  |
| Access | RW           |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |  |  |  |  |  |

| Bit  | Name | Function         | Description                                   | Access |
|------|------|------------------|-----------------------------------------------|--------|
| 31:0 | DATA | Operational Data | This field is used to store calculation data. | RW     |

## 42.4.4.12 PKE RAM0 Check Control Register (PKE\_RAM0\_CKCR)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |        |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|--------|
| 0x0000 | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0      |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |        |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   | ECCMOD |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   | RW     |

| Bit  | Name     | Function                          | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | Access |
|------|----------|-----------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:2 | Reserved | Reserved                          | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | -      |
| 1:0  | ECCMOD   | ECC Check Enable Bit for PKE RAM0 | <ul style="list-style-type: none"> <li>● 0b00: Disable ECC check function.</li> <li>● 0b01: <ul style="list-style-type: none"> <li>▶ If a 1-bit error occurs, ECC corrects the error without generating a 1-bit error flag or interrupt.</li> <li>▶ If a 2-bit error occurs, ECC detects the error, sets the 2-bit error flag, and generates an interrupt.</li> </ul> </li> <li>● 0b10: <ul style="list-style-type: none"> <li>▶ If a 1-bit error occurs, ECC corrects the error, sets the 1-bit error flag, and no interrupt is generated.</li> <li>▶ If a 2-bit error occurs, ECC detects the error, sets the 2-bit error flag, and generates an interrupt.</li> </ul> </li> <li>● 0b11: <ul style="list-style-type: none"> <li>▶ If a 1-bit error occurs, ECC corrects the error, sets the 1-bit error flag, and generates an interrupt.</li> <li>▶ If a 2-bit error occurs, ECC detects the error, sets the 2-bit error flag, and generates an interrupt.</li> </ul> </li> </ul> | RW     |

## 42.4.4.13 PKE RAM0 Check Status Register (PKE\_RAM0\_CKSR)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |          |          |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|----------|----------|
| 0x0004 | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1        | 0        |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |          |          |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   | RAM_2ERR | RAM_1ERR |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   | RW       | RW       |

| Bit  | Name     | Function                      | Description                                                                                                                                                                 | Access |
|------|----------|-------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:2 | Reserved | Reserved                      | -                                                                                                                                                                           | -      |
| 1    | RAM_2ERR | PKE RAM0 ECC 2-bit Error Flag | Write 1 to clear the flag, write 0 has no effect<br>Read:<br><ul style="list-style-type: none"> <li>0b0: No 2-bit ECC error</li> <li>0b1: 2-bit ECC error occurs</li> </ul> | RW     |
| 0    | RAM_1ERR | PKE RAM0 ECC 1-bit Error Flag | Write 1 to clear the flag, write 0 has no effect<br>Read:<br><ul style="list-style-type: none"> <li>0b0: No 1-bit ECC error</li> <li>0b1: 1-bit ECC error occurs</li> </ul> | RW     |

## 42.4.4.14 PKE RAM0 Error Injection Enable Register (PKE\_RAM0\_EIEN)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |      |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|------|
| 0x0008 | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0    |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |      |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   | EIEN |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   | RW   |

| Bit  | Name     | Function                | Description                                                                                                           | Access |
|------|----------|-------------------------|-----------------------------------------------------------------------------------------------------------------------|--------|
| 31:1 | Reserved | Reserved                | -                                                                                                                     | -      |
| 0    | EIEN     | Error injection enabled | <ul style="list-style-type: none"> <li>0b0: Error injection disabled</li> <li>0b1: Error injection enabled</li> </ul> | RW     |

42.4.4.15 PKE RAM0 Error Injection Enable Register 0 (PKE\_RAM0\_EIBIT0)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0x000C | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | EIBIT[31:0]  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access | RW           |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit  | Name        | Function                    | Description                                                                                                                                                                                                                                                                                                                                                                                      | Access |
|------|-------------|-----------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:0 | EIBIT[31:0] | Error injection bit enabled | <div>Controls whether bits 31-0 of PKE RAM0 output data are inverted:</div> <ul style="list-style-type: none"><li>0b0: No inversion, no ECC error injection for this bit.</li><li>0b1: Invert, inject ECC error for this bit.</li></ul> <div> <b>Note</b><br/>Each bit corresponds to an output data bit.</div> | RW     |

## 42.4.4.16 PKE RAM0 Error Injection Enable Register 1 (PKE\_RAM0\_EIBIT1)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |              |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|--------------|---|---|---|---|---|---|---|
| 0x0010 | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7            | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |              |   |   |   |   |   |   |   |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   | EIBIT[38:32] |   |   |   |   |   |   |   |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   | RW           |   |   |   |   |   |   |   |

| Bit  | Name         | Function                    | Description                                                                                                                                                                                                                                                                                                                                                                                                       | Access |
|------|--------------|-----------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:7 | Reserved     | Reserved                    | -                                                                                                                                                                                                                                                                                                                                                                                                                 | -      |
| 6:0  | EIBIT[38:32] | Error injection bit enabled | Controls whether bits 38-32 of PKE RAM0 output data are inverted: <ul style="list-style-type: none"> <li>0b0: No inversion, no ECC error injection for this bit.</li> <li>0b1: Inversion, inject ECC error for this bit.</li> </ul> <div>  <b>Note</b><br/>           Each bit corresponds to an output data bit.         </div> | RW     |

**Note**

Bits 38-32 of the PKE RAM0 output data are ECC bits.



42.4.4.17 PKE RAM0 Error Address Register (PKE\_RAM0\_ECCERRADDR)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |            |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|------------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0x0014 | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15         | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |            |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | ECCERRADDR |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |            |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit   | Name       | Function            | Description                                                                  | Access |
|-------|------------|---------------------|------------------------------------------------------------------------------|--------|
| 31:13 | Reserved   | Reserved            | -                                                                            | -      |
| 12:0  | ECCERRADDR | Check Error Address | Record the offset address when PKE RAM0 generates a 1-bit or 2-bit ECC error | RO     |

## 42.4.4.18 PKE RAM1 Check Control Register (PKE\_RAM1\_CKCR)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |        |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|--------|
| 0x0020 | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0      |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |        |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   | ECCMOD |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   | RW     |

| Bit  | Name     | Function                          | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | Access |
|------|----------|-----------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:2 | Reserved | Reserved                          | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | -      |
| 1:0  | ECCMOD   | ECC Check Enable Bit for PKE RAM1 | <ul style="list-style-type: none"> <li>● 0b00: Disable ECC check function.</li> <li>● 0b01: <ul style="list-style-type: none"> <li>▶ If a 1-bit error occurs, ECC corrects the error without generating a 1-bit error flag or interrupt.</li> <li>▶ If a 2-bit error occurs, ECC detects the error, sets the 2-bit error flag, and generates an interrupt.</li> </ul> </li> <li>● 0b10: <ul style="list-style-type: none"> <li>▶ If a 1-bit error occurs, ECC corrects the error, sets the 1-bit error flag, and no interrupt is generated.</li> <li>▶ If a 2-bit error occurs, ECC detects the error, sets the 2-bit error flag, and generates an interrupt.</li> </ul> </li> <li>● 0b11: <ul style="list-style-type: none"> <li>▶ If a 1-bit error occurs, ECC corrects the error, sets the 1-bit error flag, and generates an interrupt.</li> <li>▶ If a 2-bit error occurs, ECC detects the error, sets the 2-bit error flag, and generates an interrupt.</li> </ul> </li> </ul> | RW     |

## 42.4.4.19 PKE RAM1 Check Status Register (PKE\_RAM1\_CKSR)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |          |          |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|----------|----------|
| 0x0024 | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1        | 0        |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |          |          |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   | RAM_2ERR | RAM_1ERR |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   | RW       | RW       |

| Bit  | Name     | Function                      | Description                                                                                                                                                                 | Access |
|------|----------|-------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:2 | Reserved | Reserved                      | -                                                                                                                                                                           | -      |
| 1    | RAM_2ERR | PKE RAM1 ECC 2-bit Error Flag | Write 1 to clear the flag, write 0 has no effect<br>Read:<br><ul style="list-style-type: none"> <li>0b0: No 2-bit ECC error</li> <li>0b1: 2-bit ECC error occurs</li> </ul> | RW     |
| 0    | RAM_1ERR | PKE RAM1 ECC 1-bit Error Flag | Write 1 to clear the flag, write 0 has no effect<br>Read:<br><ul style="list-style-type: none"> <li>0b0: No 1-bit ECC error</li> <li>0b1: 1-bit ECC error occurs</li> </ul> | RW     |

## 42.4.4.20 PKE RAM1 Error Injection Enable Register (PKE\_RAM1\_EIEN)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |      |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|------|
| 0x0028 | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0    |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |      |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   | EIEN |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   | RW   |

| Bit  | Name     | Function                | Description                                                                                                           | Access |
|------|----------|-------------------------|-----------------------------------------------------------------------------------------------------------------------|--------|
| 31:1 | Reserved | Reserved                | -                                                                                                                     | -      |
| 0    | EIEN     | Fault injection enabled | <ul style="list-style-type: none"> <li>0b0: Error injection disabled</li> <li>0b1: Error injection enabled</li> </ul> | RW     |

## 42.4.4.21 PKE RAM1 Error Injection Enable Register 0 (PKE\_RAM1\_EIBIT0)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0x002C | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | EIBIT[31:0]  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access | RW           |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit  | Name        | Function                    | Description                                                                                                                                                                                                                                                                                                                                                                                               | Access |
|------|-------------|-----------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:0 | EIBIT[31:0] | Error injection bit enabled | <p>Controls whether bits 31-0 of PKE RAM1 output data are inverted:</p> <ul style="list-style-type: none"> <li>● 0b0: No inversion, no ECC error injection for this bit.</li> <li>● 0b1: Inversion, inject ECC error for this bit.</li> </ul> <div>  <b>Note</b><br/> Each bit corresponds to an output data bit. </div> | RW     |

## 42.4.4.22 PKE RAM1 Error Injection Enable Register 1 (PKE\_RAM1\_EIBIT1)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |              |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|--------------|---|---|---|---|---|---|---|
| 0x0030 | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7            | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |              |   |   |   |   |   |   |   |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   | EIBIT[38:32] |   |   |   |   |   |   |   |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   | RW           |   |   |   |   |   |   |   |

| Bit  | Name         | Function                    | Description                                                                                                                                                                                                                                                                                                                                                                                                       | Access |
|------|--------------|-----------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:7 | Reserved     | Reserved                    | -                                                                                                                                                                                                                                                                                                                                                                                                                 | -      |
| 6:0  | EIBIT[38:32] | Error injection bit enabled | Controls whether bits 38-32 of PKE RAM1 output data are inverted: <ul style="list-style-type: none"> <li>0b0: No inversion, no ECC error injection for this bit.</li> <li>0b1: Inversion, inject ECC error for this bit.</li> </ul> <div>  <b>Note</b><br/>           Each bit corresponds to an output data bit.         </div> | RW     |

**Note**

Bits 38-32 of the PKE RAM1 output data are ECC bits.

## 42.4.4.23 PKE RAM1 Error Address Register (PKE\_RAM1\_ECCERRADDR)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |            |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|------------|----|---|---|---|---|---|---|---|---|---|---|
| 0x0034 | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11         | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |            |    |   |   |   |   |   |   |   |   |   |   |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | ECCERRADDR |    |   |   |   |   |   |   |   |   |   |   |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |            |    |   |   |   |   |   |   |   |   |   |   |

| Bit   | Name       | Function            | Description                                                          | Access |
|-------|------------|---------------------|----------------------------------------------------------------------|--------|
| 31:13 | Reserved   | Reserved            | -                                                                    | -      |
| 12:0  | ECCERRADDR | Check Error Address | Record the offset when PKE RAM1 generates a 1-bit or 2-bit ECC error | RO     |

## 42.4.5 SKE Register

## 42.4.5.1 SKE Control Register (SKE\_CTRL)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |       |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|-------|
| 0x0000 | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0     |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |       |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   | START |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   | WO    |

| Bit  | Name     | Function | Description                                                                                                                                                                                                                                                                                                                                                                                                                                         | Access |
|------|----------|----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:1 | Reserved | Reserved | -                                                                                                                                                                                                                                                                                                                                                                                                                                                   | -      |
| 0    | START    | Start    | Write 1 to start SKE, write 0 has no effect<br><div>  <b>Note</b> <ul style="list-style-type: none"> <li>When SKE_CFG.UP_CFG is 1, SKE is started to update the configuration without performing data operations;</li> <li>When SKE_CFG.UP_CFG is 0, SKE is started to perform data operations, and the data input uses the value of SKE_DINx.</li> </ul> </div> | WO     |

## 42.4.5.2 SKE Configuration Register (SKE\_CFG)

| Offset | Bit Position |    |          |    |           |    |           |    |          |    |       |    |          |    |        |    |     |    |          |    |    |    |   |   |   |   |     |   |   |   |   |   |
|--------|--------------|----|----------|----|-----------|----|-----------|----|----------|----|-------|----|----------|----|--------|----|-----|----|----------|----|----|----|---|---|---|---|-----|---|---|---|---|---|
| 0x0004 | 31           | 30 | 29       | 28 | 27        | 26 | 25        | 24 | 23       | 22 | 21    | 20 | 19       | 18 | 17     | 16 | 15  | 14 | 13       | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5   | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |          |    |           |    |           |    |          |    |       |    |          |    |        |    |     |    |          |    |    |    |   |   |   |   |     |   |   |   |   |   |
| Name   | MODE         |    | Reserved |    | DATA_TYPE |    | SEC_STAGE |    | Reserved |    | IRQEN |    | Reserved |    | UP_CFG |    | DEC |    | Reserved |    |    |    |   |   |   |   | ALG |   |   |   |   |   |
| Access | RW           |    |          |    | RW        |    | RW        |    |          |    | RW    |    |          |    | RW     |    | RW  |    |          |    |    |    |   |   |   |   | RW  |   |   |   |   |   |

| Bit   | Name      | Function                                     | Description                                                                                                                                                                                                                                                                                                        | Access |
|-------|-----------|----------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:28 | MODE      | Pattern selection                            | <ul style="list-style-type: none"> <li>● 1: ECB mode</li> <li>● 3: CBC mode</li> <li>● 4: CFB mode</li> <li>● 5: OFB mode</li> <li>● 6: CTR mode</li> <li>● 7: CMAC mode</li> <li>● 9: GCM mode</li> <li>● 10: CCM mode</li> <li>● Others: Setting prohibited</li> </ul>                                           | RW     |
| 27:26 | Reserved  | Reserved                                     | -                                                                                                                                                                                                                                                                                                                  | -      |
| 25:24 | DATA_TYPE | Data format                                  | Data swap for SKE_DIN, SKE_DOUT <sup>1</sup> <ul style="list-style-type: none"> <li>● 0b00: No swap</li> <li>● 0b01: Half-word swap</li> <li>● 0b10: Byte swap</li> <li>● 0b11: Bit swap</li> </ul>                                                                                                                | RW     |
| 23:22 | SEC_STAGE | Data segment identifier in data interleaving | <ul style="list-style-type: none"> <li>● 0b00: No segmentation</li> <li>● 0b01: Initial segment</li> <li>● 0b10: intermediate segment</li> <li>● 0b11: final segment</li> </ul>                                                                                                                                    | RW     |
| 21:19 | Reserved  | Reserved                                     | -                                                                                                                                                                                                                                                                                                                  | -      |
| 17    | IRQEN     | Interrupt enabled                            | <ul style="list-style-type: none"> <li>● 0b0: Disable</li> <li>● 0b1: Enable</li> </ul>                                                                                                                                                                                                                            | RW     |
| 16:13 | Reserved  | Reserved                                     | -                                                                                                                                                                                                                                                                                                                  | -      |
| 12    | UP_CFG    | Update configuration                         | <ul style="list-style-type: none"> <li>● 0b0: Do not update configuration, SKE uses the last input configuration for calculation</li> <li>● 0b1: Update configuration, SKE uses the new input configuration for calculation</li> </ul> The configuration includes: ALG, DEC values in SKE_CFG, and SKE_KEYx value. | RW     |
| 11    | DEC       | Encryption or decryption                     | <ul style="list-style-type: none"> <li>● 0b0: Encryption</li> <li>● 0b1: Decryption</li> </ul>                                                                                                                                                                                                                     | RW     |

| Bit  | Name            | Function            | Description                                                                                                                                                                          | Access |
|------|-----------------|---------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 10:4 | <i>Reserved</i> | <i>Reserved</i>     | -                                                                                                                                                                                    | -      |
| 3:0  | ALG             | Algorithm selection | <ul style="list-style-type: none"> <li>● 1: AES 128</li> <li>● 2: SM4</li> <li>● 3: DES</li> <li>● 4: AES 192</li> <li>● 5: AES 256</li> <li>● Others: Setting prohibited</li> </ul> | RW     |

**Note**

1. DATA\_TYPE is used to store the plaintext to be encrypted or the ciphertext to be decrypted in the SKE\_DIN register before the module operation, and the SKE\_DOUT register to store the encrypted ciphertext or decrypted plaintext after the operation is completed.

Data example: 0x11223344

| DATA_TYPE | Treatment of data | Usage data |
|-----------|-------------------|------------|
| 0x0       | No swap           | 0x11223344 |
| 0x1       | Halfword swap     | 0x33441122 |
| 0x2       | Byte swap         | 0x44332211 |
| 0x3       | Bit swap          | 0x22CC4488 |



## 42.4.5.3 SKE Status Register 1 (SKE\_SR1)

**Note**

Before triggering SKE calculation, you need to read the value of BUSY in this register first, and confirm that it is 0 before performing subsequent operations.

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |             |      |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|-------------|------|
| 0x0008 | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1           | 0    |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |             |      |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   | MID_O_VALID | BUSY |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   | RW          | RO   |

| Bit  | Name         | Function                              | Description                                                                                                                                                                                                                               | Access |
|------|--------------|---------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:2 | Reserved     | Reserved                              | -                                                                                                                                                                                                                                         | -      |
| 1    | MID_O_VAL-ID | Output inter-mediate value valid flag | This bit is automatically set to 1 after calculation is completed; write 0 to clear it<br>Read:<br><ul style="list-style-type: none"> <li>0b0: Output intermediate value invalid</li> <li>0b1: Output intermediate value valid</li> </ul> | RW     |
| 0    | BUSY         | Busy flag                             | <ul style="list-style-type: none"> <li>0b0: SKE is in idle state and can receive commands or data input</li> <li>0b1: SKE is in running state and cannot receive commands or data input</li> </ul>                                        | RO     |

## 42.4.5.4 SKE Status Register 2 (SKE\_SR2)

**Note**

Before triggering SKE calculation, the value of this register must be cleared.

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |          |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|----------|
| 0x000C | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0        |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |          |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   | SKE DONE |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   | RW       |

| Bit  | Name     | Function                                | Description                                                                                                                                                                        | Access |
|------|----------|-----------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:1 | Reserved | Reserved                                | -                                                                                                                                                                                  | -      |
| 0    | SKE_DONE | SKE operation completion indication bit | Write 0 to clear, write 1 has no effect<br>Read:<br><ul style="list-style-type: none"> <li>0b0: SKE operation is not completed</li> <li>0b1: SKE operation is completed</li> </ul> | RW     |

## 42.4.5.5 SKE Key Register (SKE\_KEYx) (x=1-8)

Offset: 0x0010+0x4\*(x-1)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |     |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|-----|
| -      | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0   |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |     |
| Name   |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   | KEY |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   | WO  |

| Bit  | Name | Function  | Description                                                                                                                                                     | Access |
|------|------|-----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:0 | KEY  | Key input | This register is used as the encryption and decryption key. It is only valid when the key is entered when SKE_CFG.UP_CFG=1, and only supports 32-bit operation. | WO     |

## 42.4.5.6 SKE AAD Length Register (SKE\_AADx) (x=1-2)

Offset: 0x0060+0x4\*(x-1)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| -      | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | A_LEN        |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access | RW           |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit  | Name  | Function       | Description                                | Access |
|------|-------|----------------|--------------------------------------------|--------|
| 31:0 | A_LEN | AAD Bit Length | This register is valid in GCM and CCM mode | RW     |

## 42.4.5.7 SKE Data Length Register (SKE\_CLENx) (x=1-2)

Offset: 0x0068+0x4\*(x-1)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| -      | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | C_LEN        |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access | RW           |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit  | Name  | Function              | Description                                | Access |
|------|-------|-----------------------|--------------------------------------------|--------|
| 31:0 | C_LEN | Ciphertext Bit Length | This register is valid in GCM and CCM mode | RW     |

## 42.4.5.8 SKE Initial Value Register (SKE\_IVx) (x=1-4)

Offset: 0x0070+0x4\*(x-1)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| '      | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | IV           |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access | RW           |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit  | Name | Function      | Description                                                               | Access |
|------|------|---------------|---------------------------------------------------------------------------|--------|
| 31:0 | IV   | Initial Value | This register is valid in non-ECB mode and only supports 32-bit operation | RW     |

## 42.4.5.9 SKE Data Flag Register (SKE\_DIN\_CR)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |      |          |    |    |    |    |   |   |   |          |   |   |   |   |   |   |  |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|------|----------|----|----|----|----|---|---|---|----------|---|---|---|---|---|---|--|
| 0x0080 | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15   | 14       | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6        | 5 | 4 | 3 | 2 | 1 | 0 |  |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |      |          |    |    |    |    |   |   |   |          |   |   |   |   |   |   |  |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | LAST | Reserved |    |    |    |    |   |   |   | LAST_LEN |   |   |   |   |   |   |  |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RW   |          |    |    |    |    |   |   |   | RW       |   |   |   |   |   |   |  |

| Bit   | Name     | Function                           | Description                                                                                                                                                                                                                                                                                                                                                                                             | Access |
|-------|----------|------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:17 | Reserved | Reserved                           | -                                                                                                                                                                                                                                                                                                                                                                                                       | -      |
| 16    | LAST     | Data end flag                      | Write 0 to clear, need to clear after calculation<br>Read:<br><ul style="list-style-type: none"> <li>0b0: Current data does not include the end of data</li> <li>0b1: Current data includes the end of data</li> </ul> <div>  <b>Note</b><br/>           This bit is only valid in GCM, CCM, CMAC mode.         </div> | RW     |
| 15:8  | Reserved | Reserved                           | -                                                                                                                                                                                                                                                                                                                                                                                                       | -      |
| 7:0   | LAST_LEN | The last block length in CMAC mode | In bits, the calculation needs to be cleared (the actual length of the data, excluding padding bits)                                                                                                                                                                                                                                                                                                    | RW     |

## 42.4.5.10 SKE Data Input Register (SKE\_DINx) (x=1-4)

Offset: 0x0090+0x4\*(x-1)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| -      | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | DATA_IN      |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access | WO           |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit  | Name    | Function   | Description                                                                                                | Access |
|------|---------|------------|------------------------------------------------------------------------------------------------------------|--------|
| 31:0 | DATA_IN | Data Input | This register only supports 32-bit operations. In CMAC mode, the data fed into this register is un-padded. | WO     |

## 42.4.5.11 SKE Data Output Register (SKE\_DOUTx) (x=1-4)

Offset: 0x00B0+0x4\*(x-1)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| '      | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | DATA_OUT     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access | RO           |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit  | Name     | Function    | Description                                                                               | Access |
|------|----------|-------------|-------------------------------------------------------------------------------------------|--------|
| 31:0 | DATA_OUT | Data Output | When SKE_SR2.CORE_DONE=1, the value of this register can be read as the operation result. | RO     |

## 42.4.5.12 SKE Version Register (SKE\_VERSION)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |          |    |    |    |    |    |   |   |     |   |   |   |     |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----------|----|----|----|----|----|---|---|-----|---|---|---|-----|---|---|---|
| 0x00FC | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15       | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7   | 6 | 5 | 4 | 3   | 2 | 1 | 0 |
| Reset  | 0x001D0011   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |          |    |    |    |    |    |   |   |     |   |   |   |     |   |   |   |
| Name   | PROJECT      |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | Reserved |    |    |    |    |    |   |   | MAR |   |   |   | MIR |   |   |   |
| Access | RO           |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |          |    |    |    |    |    |   |   | RO  |   |   |   | RO  |   |   |   |

| Bit   | Name     | Function          | Description | Access |
|-------|----------|-------------------|-------------|--------|
| 31:16 | PROJECT  | Project No.       | -           | RO     |
| 15:8  | Reserved | Reserved          | -           | -      |
| 7:4   | MAR      | Major Version No. | -           | RO     |
| 3:0   | MIR      | Minor Version No. | -           | RO     |

## 42.4.5.13 SKE Intermediate Value Register (SKE\_MID\_IVx) (x=1-4)

Offset: 0x01A0+0x4\*(x-1)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| -      | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | MID_IV       |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access | RW           |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit  | Name   | Function                                               | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                             | Access |
|------|--------|--------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:0 | MID_IV | Initial vector intermediate value in data interleaving | <p>This register is valid when SKE_CFG.SEC_STAGE is not 0 and only supports 32-bit operation. Used for the initial, intermediate, and final segments of GCM, CCM, and CMAC.</p> <ul style="list-style-type: none"> <li>Before the calculation starts (intermediate segment and final segment), write the corresponding initial value;</li> <li>At the end of the calculation (initial segment and intermediate segment), read the output intermediate value.</li> </ul> | RW     |

**Note**

Intermediate values include input values and output values, which are accessed using the same address, and input values cannot be read after they are written. When SKE\_SR1.MID\_O\_VALID=1, it means that the output value in this register is valid.

## 42.4.5.14 SKE Intermediate MAC Register (SKE\_MID\_MACx) (x=1-4)

Offset: 0x01B0+0x4\*(x-1)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |  |  |  |  |  |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|--|--|--|--|--|--|
| -      | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |  |  |  |  |  |  |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |  |  |  |  |  |
| Name   | MID_MAC      |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |  |  |  |  |  |
| Access | RW           |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |  |  |  |  |  |

| Bit  | Name    | Function                                        | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | Access |
|------|---------|-------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:0 | MID_MAC | MAC/TAG intermediate value in data interleaving | <p>This register is valid when SKE_CFG.SEC_STAGE is not 0 and only supports 32-bit operation. Used for the initial segment, intermediate segment and final segment of GCM and CCM.</p> <ul style="list-style-type: none"> <li>Before the calculation starts (intermediate segment and final segment), write the corresponding initial value;</li> <li>At the end of the calculation (initial segment and intermediate segment), read the output intermediate value.</li> </ul> | RW     |

**Note**

The intermediate value includes the input value and the output value, which are accessed using the same address, and the input value cannot be read after it is written. When SKE\_SR1.MID\_O\_VALID=1, it means that the output value in this register is valid.

## 42.4.5.15 SKE AAD Total Data Length Register (SKE\_TOTAL\_AADx) (x=1-2)

Offset: 0x0060+0x4\*(x-1)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| ,      | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | TOTAL_A_LEN  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access | RW           |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit  | Name        | Function                                     | Description                                                                                                                            | Access |
|------|-------------|----------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:0 | TOTAL_A_LEN | Total bit length of AAD in data interleaving | This register is valid when SKE_CFG.SEC_STAGE=3.<br>Used in the final segment of GCM mode, it identifies the total length of data AAD. | RW     |

**Note**

The value of this register indicates the length of AAD of the entire GCM transaction, not the length of AAD of the current segment.

## 42.4.5.16 SKE Total Data Length Register (SKE\_TOTAL\_CLENx) (x=1-2)

Offset: 0x0068+0x4\*(x-1)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| ,      | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | TOTAL_C_LEN  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access | RW           |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit  | Name        | Function                                         | Description                                                                                                                                | Access |
|------|-------------|--------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:0 | TOTAL_C_LEN | Total bit length of Payload in data interleaving | This register is valid when SKE_CFG.SEC_STAGE=3.<br>Used in the final segment of GCM mode, it identifies the total length of data Payload. | RW     |



## 43 CRC Operation (CRC)

### 43.1 Introduction

In many applications, CRC algorithms are required to verify the integrity and correctness of data. Especially in data transmission, CRC checking is widely used. This module can use 11 algorithms, including CRC16/CCITT, CRC16/CCITT-FALSE, CRC16/XMODEM, CRC16/IBM, CRC16/MAXIM, CRC16/USB, CRC16/MODBUS, CRC16/X25, CRC32, CRC32/MPEG-2 and CRC64, to calculate and check the data.

### 43.2 Functional Description

#### 43.2.1 Function Block Diagram

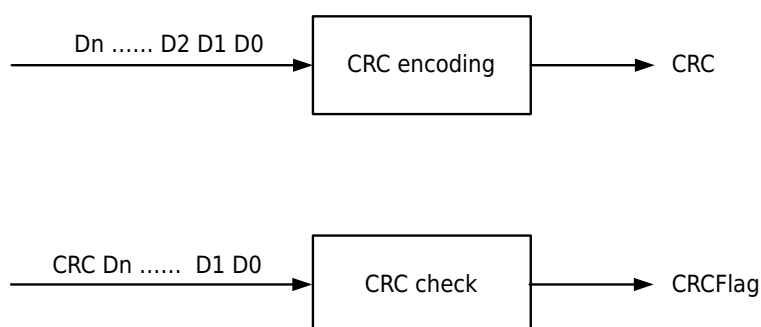


Figure 43-1 CRC Application Diagram

#### 43.2.2 CRC Parameter Model

The CRC algorithm of this module complies with the definition of ISO/IEC13239, using 16-bit, 32-bit, and 64-bit CRC respectively. The corresponding parameter models for each CRC algorithm are shown in the table below.

Table 43-1 CRC Parameter Model

| CRC Algorithm Name | Polynomial Formula                                                                | Initial Value | Result XOR Value | Input Inversion | Output Inversion |
|--------------------|-----------------------------------------------------------------------------------|---------------|------------------|-----------------|------------------|
| CRC16/IBM          | $X^{16}+X^{15}+X^2+1$                                                             | 0x0000        | 0x0000           | Yes             | Yes              |
| CRC16/MAXIM        | $X^{16}+X^{15}+X^2+1$                                                             | 0x0000        | 0xFFFF           | Yes             | Yes              |
| CRC16/USB          | $X^{16}+X^{15}+X^2+1$                                                             | 0xFFFF        | 0xFFFF           | Yes             | Yes              |
| CRC16/MODBUS       | $X^{16}+X^{15}+X^2+1$                                                             | 0xFFFF        | 0x0000           | Yes             | Yes              |
| CRC16/CCITT        | $X^{16}+X^{12}+X^5+1$                                                             | 0x0000        | 0x0000           | Yes             | Yes              |
| CRC16/CCITT-FALSE  | $X^{16}+X^{12}+X^5+1$                                                             | 0xFFFF        | 0x0000           | No              | No               |
| CRC16/X25          | $X^{16}+X^{12}+X^5+1$                                                             | 0xFFFF        | 0xFFFF           | Yes             | Yes              |
| CRC16/XMODEM       | $X^{16}+X^{12}+X^5+1$                                                             | 0x0000        | 0x0000           | No              | No               |
| CRC32              | $X^{32}+X^{26}+X^{23}+X^{22}+X^{16}+X^{12}+X^{11}+X^{10}+X^8+X^7+X^5+X^4+X^2+X+1$ | 0xFFFFFFFF    | 0xFFFFFFFF       | Yes             | Yes              |

| CRC Algorithm Name | Polynomial Formula                                                                | Initial Value      | Result XOR Value   | Input Inversion | Output Inversion |
|--------------------|-----------------------------------------------------------------------------------|--------------------|--------------------|-----------------|------------------|
| CRC32/MPEG-2       | $X^{32}+X^{26}+X^{23}+X^{22}+X^{16}+X^{12}+X^{11}+X^{10}+X^8+X^7+X^5+X^4+X^2+X+1$ | 0xFFFFFFFF         | 0x00000000         | No              | No               |
| CRC64              | $X^{64}+X^4+X^3+X+1$                                                              | 0xFFFFFFFFFFFFFFFF | 0xFFFFFFFFFFFFFFFF | Yes             | Yes              |

The functions of this module include:

- CRC encoding and CRC check
- 3 bit width access methods: 8-bit, 16-bit, 32-bit:
  - ▶ Example of input data with 8-bit bit width: 0x00, 0x11, 0x22, 0x33, 0x44, 0x55, 0x66, 0x77;
  - ▶ Example of input data with 16-bit bit width: 0x1100, 0x3322, 0x5544, 0x7766;
  - ▶ Example of input data with 32-bit bit width: 0x33221100, 0x77665544.

### 43.2.3 CRC16 Encoding Mode

CRC encoding is the process of encoding the original data to calculate its CRC value. The operating procedures are as follows:

1. Write 0b0 to CRC\_CR.CR and write corresponding data to CRC\_CR.SELECT, selecting the corresponding CRC16 specification algorithm.
2. Initialize CRC calculation by writing 0x0000 or 0xFFFF to CRC\_RESULT1[15:0] according to the [CRC parameter model](#).
3. Organize the original data to be encoded into 8-bit/16-bit/32-bit formats and write them into CRC\_DAT register sequentially.



#### Note

When writing the CRC value in 8-bit format, write the low bits first and then the high bits.

4. Read CRC\_RESULT1[15:0] to obtain the 16-bit CRC encoded value.

### 43.2.4 CRC16 Check Mode

The check mode can verify whether the encoded data has been tampered. The operating procedures are as follows:

1. Write 0b0 to CRC\_CR.CR and write corresponding data to CRC\_CR.SELECT, selecting the corresponding CRC16 specification algorithm.
2. Initialize CRC calculation by writing 0x0000 or 0xFFFF to CRC\_RESULT1[15:0] according to the [CRC parameter model](#).
3. Write the encoded data into CRC\_DAT register sequentially in 8-bit, 16-bit, or 32-bit organization.



#### Note

When writing the CRC value in 8-bit format, write the low bits first and then the high bits.

4. Write the check code into the CRC\_DAT register with a 16-bit bit width.

**Note**

For CRC16/CCITT-FALSE and CRC16/XMODEM, the check code needs to be swapped between the high 8 bits and low 8 bits.

For example: 0x1234-> 0x3412

5. Read the value of CRC\_CR.FLAG. A value of 1 indicates check success, and a value of 0 indicates check failure.

### 43.2.5 CRC32 Encoding Mode

CRC encoding is the process of encoding the original data to calculate its CRC value. The operating procedures are as follows:

1. Write 0b1 to CRC\_CR.CR and write corresponding data to CRC\_CR.SELECT, selecting the corresponding CRC32 specification algorithm.
2. Write 0xFFFFFFFF to CRC\_RESULT1[31:0] to initialize CRC calculation.
3. Organize the original data to be encoded into 8-bit/16-bit/32-bit formats and write them into CRC\_DAT register sequentially.

**Note**

When writing the CRC value in 8-bit format, write the low bits first and then the high bits.

4. Read CRC\_RESULT1[31:0] to obtain the 32-bit CRC encoded value.

### 43.2.6 CRC32 Check Mode

The check mode can verify whether the encoded data has been tampered with. The operating procedures are as follows:

1. Write 0b1 to CRC\_CR.CR and write corresponding data to CRC\_CR.SELECT, selecting the corresponding CRC32 specification algorithm.
2. Write 0xFFFFFFFF to CRC\_RESULT1[31:0] to initialize CRC calculation.
3. Write the encoded data into CRC\_DAT register sequentially in 8-bit, 16-bit, or 32-bit organization.

**Note**

When writing the CRC value in 8-bit format, write the low bits first and then the high bits.

4. Write the check code into the CRC\_DAT register in 8-bit, 16-bit, or 32-bit organization.

**Note**

For CRC32/MPEG-2, during check, the high 16 bits and low 16 bits need to be swapped, and within each 16-bit section, the high 8 bits and low 8 bits also need to be swapped.

For example: 0x12345678->0x78563412

5. Read the value of CRC\_CR.FLAG. A value of 1 indicates check success, and a value of 0 indicates check failure.

### 43.2.7 CRC64 Encoding Mode

CRC encoding is the process of encoding the original data to calculate its CRC value. The operating procedures are as follows:

1. Write 0b1 to CRC\_CR.CR and write corresponding data to CRC\_CR.SELECT, selecting the corresponding CRC64 specification algorithm.

2. Write 0xFFFFFFFF to CRC\_RESULT1[31:0] and CRC\_RESULT2[31:0] to initialize CRC calculation.
3. Organize the original data to be encoded into 8-bit/16-bit/32-bit formats and write them into CRC\_DAT register sequentially.

**Note**

When writing the CRC value in 8-bit format, write the low bits first and then the high bits.

4. Read CRC\_RESULT1[31:0] and CRC\_RESULT2[31:0] to obtain the 64-bit CRC encoded value.

**Note**

CRC64\_DATA= {CRC\_RESULT2[31:0], CRC\_RESULT1[31:0]}.

### 43.2.8 CRC64 Check Mode

The check mode can verify whether the encoded data has been tampered. The operating procedures are as follows:

1. Write 0b1 to CRC\_CR.CR and write corresponding data to CRC\_CR.SELECT, selecting the corresponding CRC64 specification algorithm.
2. Write 0xFFFFFFFF to CRC\_RESULT1[31:0] and CRC\_RESULT2[31:0] to initialize CRC calculation.
3. Write the encoded data into CRC\_DAT register sequentially in 8-bit, 16-bit, or 32-bit organization.

**Note**

When writing the CRC value in 8-bit format, write the low bits first and then the high bits.

4. Write the check code into the CRC\_DAT register in 8-bit, 16-bit, or 32-bit organization.
5. Read the CRC\_CR.FLAG register. A value of 1 indicates check success, and a value of 0 indicates check failure.

### 43.3 Register Description

#### 43.3.1 Register List

Base address: 0x40008C00

Table 43-2 CRC Operation (CRC) Register Offset

| Offset    | Register    | Description           | Width |
|-----------|-------------|-----------------------|-------|
| 0x00      | CRC_CR      | CRC Control Register  | 32    |
| 0x04      | CRC_RESULT1 | CRC Result Register 1 | 32    |
| 0x08      | CRC_RESULT2 | CRC Result Register 2 | 32    |
| 0x80-0xFC | CRC_DAT     | CRC Data Register     | 32    |

## 43.3.2 Control Register (CRC\_CR)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |    |   |        |    |  |      |    |  |    |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|----|---|--------|----|--|------|----|--|----|
| 0x00   | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3  | 2 | 1      | 0  |  |      |    |  |    |
| Reset  | 0x00000001   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |    |   |        |    |  |      |    |  |    |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   | RW |   | SELECT | RO |  | FLAG | RW |  | CR |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |    |   |        |    |  |      |    |  |    |

| Bit  | Name     | Function                | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | Access |
|------|----------|-------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:5 | Reserved | Reserved                | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | -      |
| 4:2  | SELECT   | CRC selection bit       | <b>When CR=0b0, Bit4-2 correspond to CRC algorithm selection:</b> <ul style="list-style-type: none"> <li>● 0b000: CRC16/X25</li> <li>● 0b001: CRC16/CCITT</li> <li>● 0b010: CRC16/CCITT-FALSE</li> <li>● 0b011: CRC16/XMODEM</li> <li>● 0b100: CRC16/IBM</li> <li>● 0b101: CRC16/MAXIM</li> <li>● 0b110: CRC16/USB</li> <li>● 0b111: CRC16/MODBUS</li> </ul> <b>When CR=0b1, Bit4-2 correspond to CRC algorithm selection:</b> <ul style="list-style-type: none"> <li>● 0b000: CRC32</li> <li>● 0b001: CRC32/MPEG-2</li> <li>● 0b010: CRC64</li> <li>● Others: Setting prohibited</li> </ul> | RW     |
| 1    | FLAG     | Check result bit        | <ul style="list-style-type: none"> <li>● 0b0: Current check error</li> <li>● 0b1: Current check correct</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | RO     |
| 0    | CR       | Operational control bit | <ul style="list-style-type: none"> <li>● 0b0: CRC16</li> <li>● 0b1: CRC32/CRC64</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | RW     |

## 43.3.3 Result Register 1 (CRC\_RESULT1)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0x04   | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | RESULT1      |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access | RW           |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit  | Name    | Function   | Description                                                                                                                                                                                                                                                                                                                                                   | Access |
|------|---------|------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:0 | RESULT1 | Result bit | <ul style="list-style-type: none"> <li>When CRC16 is selected, use RESULT1[15:0].</li> <li>When CRC32 is selected, use RESULT1[31:0].</li> <li>When selecting CRC64, CRC_RESULT1 and CRC_RESULT2 are used jointly. CRC_RESULT1.RESULT1[31:0] corresponds to CRC result Bit31-Bit0; CRC_RESULT2.RESULT2[31:0] corresponds to CRC result Bit63-Bit32</li> </ul> | RW     |

## 43.3.4 Result Register 2 (CRC\_RESULT2)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0x08   | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | RESULT2      |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access | RW           |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit  | Name    | Function   | Description                                                                                                                                                                                                                                                   | Access |
|------|---------|------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:0 | RESULT2 | Result bit | When selecting CRC64, CRC_RESULT1 and CRC_RESULT2 are used jointly: <ul style="list-style-type: none"> <li>CRC_RESULT1.RESULT1[31:0] corresponds to CRC result Bit31~Bit0</li> <li>CRC_RESULT2.RESULT2[31:0] corresponds to CRC result Bit63~Bit32</li> </ul> | RW     |

### 43.3.5 Data Register (CRC\_DAT)

| Offset    | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |  |  |  |  |  |
|-----------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|--|--|--|--|--|--|
| 0x80-0xFC | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |  |  |  |  |  |  |
| Reset     | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |  |  |  |  |  |
| Name      | CRC_DAT      |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |  |  |  |  |  |
| Access    | RW           |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |  |  |  |  |  |

| Bit  | Name    | Function           | Description                                          | Access |
|------|---------|--------------------|------------------------------------------------------|--------|
| 31:0 | CRC_DAT | Data for operation | Write the data to be operated on; reads always as 0. | RW     |



**Note**

CRC\_DAT register is at the 0x80 offset from the CRC base address. The 0x84-0xFC offset from the base address are the mapping addresses of the CRC\_DAT register, with the same access and operation permissions as the CRC\_DAT register.



## 44 Data Computing Unit (DCU)

### 44.1 Introduction

The Data Computing Unit (DCU) is a simple data processing block that does not rely on the CPU. Each DCU unit has three data registers capable of performing addition, subtraction, comparison, and window comparison functions on two data values. It can also provide continuously varying digital values to the DAC module through timer triggers to generate triangular and sawtooth wave outputs. This product is equipped with 8 DCU units.

### 44.2 Features

- Supports 4 types of data processing: addition, subtraction, comparison of 2 data values, and window comparison of 3 data values.
- Addition and subtraction operations are performed on data in the DATA0 and DATA1 registers, with results stored in DATA0. Simultaneously, the result is automatically halved, and the halved result is stored in DATA1.
- Addition and subtraction can be triggered by writing to the register or by events from other peripheral circuits.
- In comparison mode, two data values can be compared between DATA0 and DATA1 registers, as well as between DATA0 and DATA2 registers. Interrupt and flag bits can be generated when the comparison results meet the conditions of greater than, less than, or equal to.
- Compare mode can also be used for window comparison by setting DATA1 and DATA2 as the upper and lower limits of the window, and judging whether DATA0 is within or outside the window based on the comparison results between DATA0 and DATA1, and DATA0 and DATA2.
- Operations can be triggered by events from other peripheral circuits, and various interrupts and event signals can be generated based on the operation results. When other peripheral circuits with hardware trigger start function select DCU as the trigger source, the DCU generates an event signal to start the peripheral circuit's action.
- In the triangular wave output mode, a timer is used to trigger the generation of continuously increasing and decreasing digital quantities to the digital-to-analog conversion module (DAC), the latter can then convert these digital quantities into triangular wave output through digital-to-analog conversion
- In the incremental sawtooth wave output mode, a timer is used to trigger the generation of continuously increasing digital quantities to the digital-to-analog conversion module (DAC), the latter can then convert these digital quantities into incremental sawtooth wave outputs through digital-to-analog conversion
- In the decreasing sawtooth wave output mode, a timer is used to trigger the generation of continuously decreasing digital quantities to the digital-to-analog conversion module (DAC), the latter can then convert these digital quantities into decreasing sawtooth wave output through digital-to-analog conversion

### 44.3 Functional Description

#### 44.3.1 Addition Mode

The addition mode calculates the sum of DATA0 and DATA1, where DATA0 is used as the summand and DATA1 is used as the addend. Each write to the DATA1 register performs a  $\text{DATA0} + \text{DATA1}$  addition

operation, with the result stored in DATA0 and the result divided by 2 stored in DATA2. An interrupt and flag bit are generated when the result of DATA0 + DATA1 is greater than 0xFF (8-bit mode) or 0xFFFF (16-bit mode) or 0xFFFFFFFF (32-bit mode).

Application example of addition mode:

- Step1.** Set the DCU\_CTL control register to addition mode with a data width of 16-bit;
- Step2.** Select the interrupt generation condition in the DCU\_INTEVTSEL register;
- Step3.** Write 0xFF00 and 0x55 to DATA0 and DATA1, respectively. The calculation result is 0xFF55, which is saved in DATA0, and 0xFF55/2 is saved in DATA2;
- Step4.** Continue writing 0xFF to DATA1. At this point, the calculation result ( $0xFF55 + 0xFF > 0xFFFF$ ) overflows, generating a result flag. Reading the flag register DCU\_FLAG obtains the result flag;
- Step5.** Write to the flag reset register DCU\_FLAGCLR to clear the flag bit.

### 44.3.2 Subtraction Mode

The subtraction mode calculates the difference between DATA0 and DATA1, where DATA0 acts as the minuend and DATA1 acts as the subtrahend. Each write to the DATA1 register performs a DATA0-DATA1 subtraction operation. The subtraction result is stored in DATA0, and the result divided by 2 is stored in DATA2. An interrupt and flag bit are generated when the DATA0-DATA1 result is less than 0x0 (in 8-bit, 16-bit, or 32-bit mode).

### 44.3.3 Hardware Trigger Start Mode

The DCU can trigger and start operations according to events generated by peripheral circuits. When using hardware trigger start mode, first enable the peripheral circuit trigger function in the function clock control 0 register (FCG0). Each DCU unit can independently select trigger start signals from other peripheral circuit, when select start signals, writing the peripheral circuit source number in the trigger source selection register (DCU\_TRGSEL). For details of the registers, please refer to the Automatic Operation System (AOS). When the peripheral circuit event occurs, the event signal is input to the DCU to trigger the DCU start operation.

Hardware trigger start mode includes trigger plus mode and trigger minus mode:

- In trigger plus mode, each event trigger initiates a DATA0+DATA1 addition. The result is stored in DATA0, and the result divided by 2 is stored in DATA2. If the result exceeds 0xFF (8-bit mode), 0xFFFF (16-bit mode), or 0xFFFFFFFF (32-bit mode), the flag bit is set and an interrupt is generated.
- In trigger minus mode, each trigger initiates a DATA0-DATA1 subtraction. The result is stored in DATA0, and the result divided by 2 is stored in DATA2. When the result of DATA0-DATA1 is less than 0x0 (in 8-bit, 16-bit, and 32-bit mode), the flag bit is set and an interrupt is generated.

Example application of hardware trigger start mode:

- Step1.** The control register DCU\_CTL selects the trigger plus mode, with a data width of 8-bit;
- Step2.** The interrupt condition selection register DCU\_INTEVTSEL selects the operation conditions;
- Step3.** Write 0x00 and 0x56 to DATA0 and DATA1 respectively;
- Step4.** Write the event number in the trigger source selection register DCU\_TRGSEL;
- Step5.** Make the selected peripheral circuit action and generate events, the DCU is triggered to perform an addition operation. The result is 0x56 and saved in DATA0, and 0x56 divided by 2 is saved in DATA2;
- Step6.** After three consecutive triggers, the calculation result overflows, generating a result flag. Reading the flag register DCU\_FLAG obtains the result flag;
- Step7.** Write to the flag reset register DCU\_FLAGCLR to clear the flag bit.

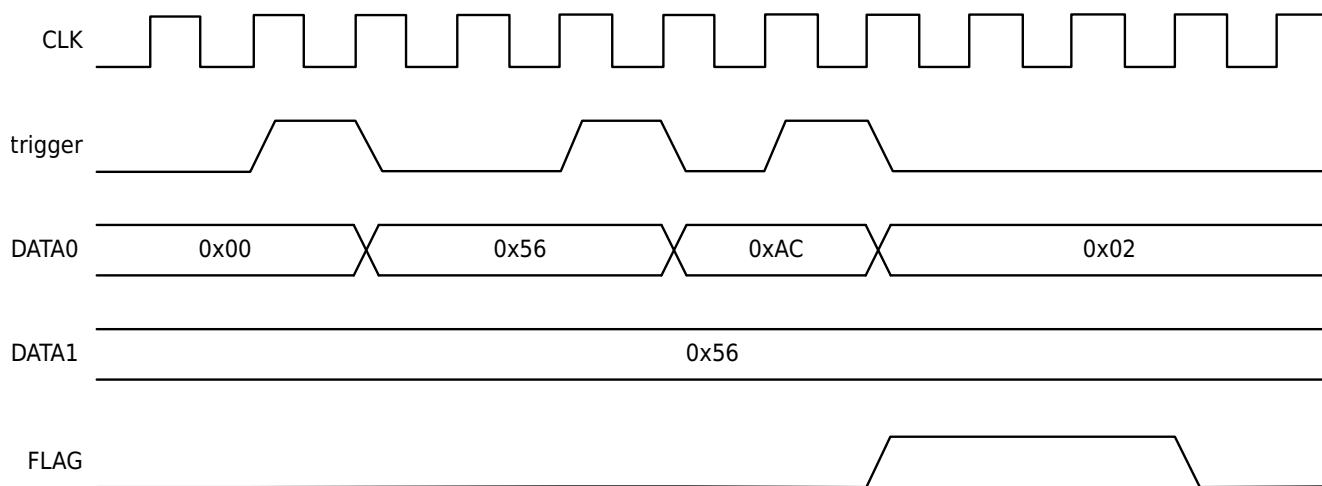


Figure 44-1 Hardware Trigger Start Schematic Diagram

#### 44.3.4 Comparative Mode

Comparing the size of DATA0 and DATA1 and DATA0 and DATA2, it can select when DATA0 is greater than DATA1, DATA0 is less than DATA1, DATA0 is equal to DATA1 and when DATA0 is greater than DATA2, DATA0 is less than DATA2, DATA0 is equal to DATA2, generate a flag and generate an interrupt. In the comparative mode, it can choose the conditions for data comparison, compare after writing DATA0 or write any data register.

Comparative mode application example:

- Step1.** Select comparative mode in the control register DCU\_CTL, with data width of 16-bit. Start comparison after writing to DATA0;
- Step2.** In the interrupt condition selection register DCU\_INTEVTSEL, select to generate a flag when DATA0>DATA1;
- Step3.** Write 0xB BBBB and 0xA AAAA to DATA1 and DATA0 respectively. Since DATA0>DATA1 is not satisfied, no flag is generated;
- Step4.** Write 0x8888 to DATA1. Although DATA0>DATA1 is now satisfied, since it's set to start comparison after writing to DATA0, no flag is generated;
- Step5.** After writing 0x9999 to DATA0, the flag generation condition is met, and a flag is generated;
- Step6.** Write to the flag reset register DCU\_FLAGCLR to clear the flag bit.

#### 44.3.5 Triangle Wave Output Mode

Triangle wave output mode is where DCU uses a timer as the trigger source to provide a continuously changing digital value for the Digital-to-Analog Converter (DAC), which converts these digital values into a triangle wave output. Triangle wave output mode is set through the DCU control register (DCU\_CTL). After setting to triangle wave output mode, the functions of data registers DATA0-2 will change. The data register DATA0 will serve as the data register for the digital values output to the DAC and be reset. DATA0 becomes the output data register, with DATA.DO (Bit0-Bit11) as active bits. DATA1 becomes the amplitude setting register, with DATA1.UPL (Bit16-Bit27) and DATA1.LWL (Bit0-Bit11) as active bits. DATA2 becomes the step setting register, with DATA2.STEP (Bit0-Bit11) as active bits.



#### Note

Only DCU1-4 support triangle wave output mode.

Steps for using triangle wave output mode:

- Step1.** Set the MODE in DCU control register (DCU\_CTL) to 0b1000;
- Step2.** Set the upper and lower limits of the triangle wave amplitude. The upper limit is set by DATA1.UPL[11:0] bit, and the lower limit is set by DATA1.LWL[11:0];
- Step3.** Set the step values for triangle wave increment and decrement through DATA2.STEP[11:0];
- Step4.** Select an appropriate timer event or software trigger as the trigger source through the trigger source selection register (DCU\_TRGSEL);
- Step5.** Configure the selected timer to generate trigger events at set time intervals;
- Step6.** Start the timer. On the first trigger event, the value of DATA1.LWL is loaded into DATA0.DO and output to DAC. Each time a trigger event occurs in the future, DATA0.DO will increment according to the step value set in DATA2.STEP;
- Step7.** When DATA0.DO increments to the upper limit set by DATA1.UPL, it will automatically start to decrement according to the step value set in DATA2.STEP. When it decrements to the lower limit set by DATA1.LWL, it starts to increment again;
- Step8.** If an interrupt is required when the triangle wave reaches its peak or valley, first set the interrupt enable bit (INTEN) in the DCU control register (DCU\_CTL) to inactive, then set the values of DATA0-DATA2. Clear the DCU flag bit once before starting the timer to generate trigger events, then set the interrupt enable bit (INTEN) in the DCU control register to active before starting the timer;
- Step9.** Set the MODE in DCU control register (DCU\_CTL) to 0b0000 to exit triangle wave output mode.

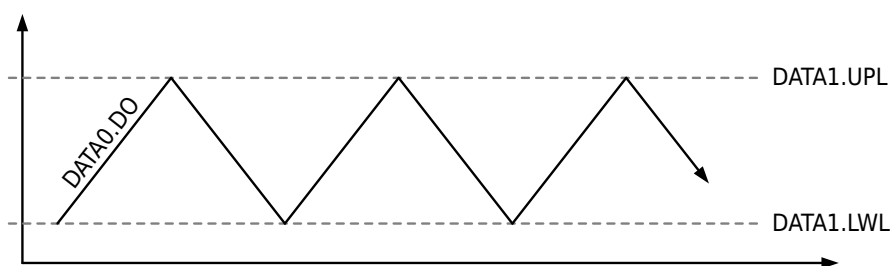


Figure 44-2 Triangular Wave Output Mode Waveform

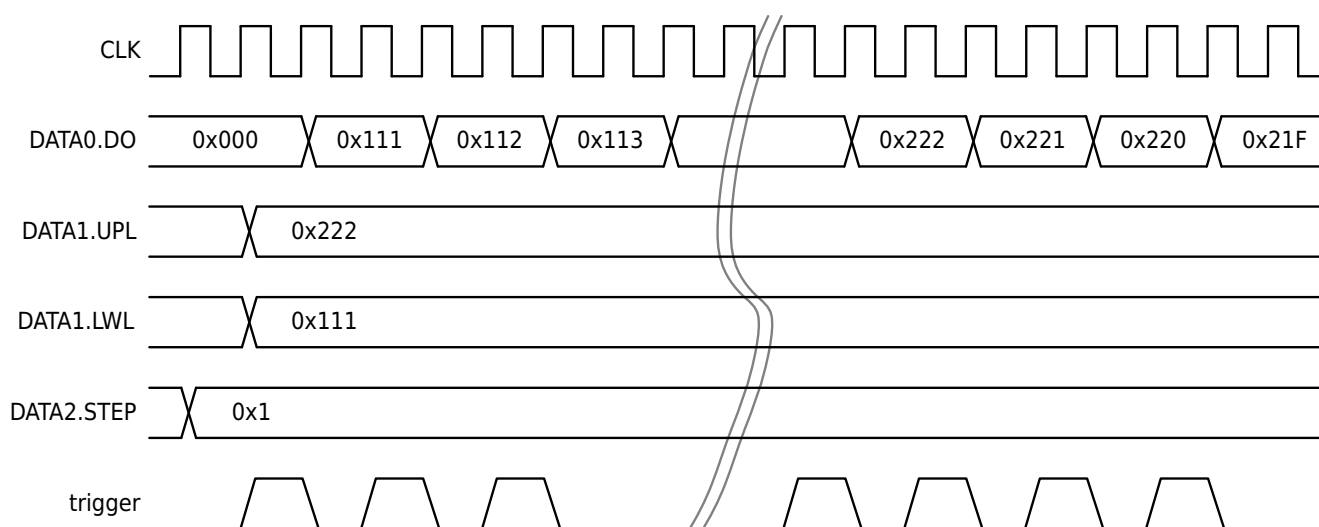


Figure 44-3 Triangular Wave Output Mode

### 44.3.6 Increment Sawtooth Wave Output Mode

Increment sawtooth wave output mode is where DCU uses a timer as the trigger source to provide a continuously changing digital value for the Digital-to-Analog Converter (DAC), which converts these digital values into a sawtooth wave output. Increment sawtooth wave output mode is set through the DCU control register (DCU\_CTL). After setting to increment sawtooth wave output mode, the functions of data registers DATA0-2 will change. The data register DATA0 will serve as the data register for the digital values output to the DAC and be reset. DATA0 becomes the output data register, with DATA.DO (Bit0-Bit11) as active bits. DATA1 becomes the amplitude setting register, with DATA1.UPL (Bit16-Bit27) and DATA1.LWL (Bit0-Bit11) as active bits. DATA2 becomes the step setting register, with DATA2.STEP (Bit0-Bit11) as active bits.



#### Note

Only DCU1-4 support increment sawtooth wave output mode.

Steps for using increment sawtooth wave output mode:

- Step1.** Set the MODE in DCU control register (DCU\_CTL) to 0b1001;
- Step2.** Set the upper and lower limits of the sawtooth wave amplitude. The upper limit is set by DATA1.UPL[11:0] bit, and the lower limit is set by DATA1.LWL[11:0];
- Step3.** Set the increment step value of the sawtooth wave through DATA2.STEP[11:0];
- Step4.** Select an appropriate timer event or software trigger as the trigger source through the trigger source selection register (DCU\_TRGSEL);
- Step5.** Configure the selected timer to generate trigger events at set time intervals;
- Step6.** Start the timer. On the first trigger event, the value of DATA1.LWL is loaded into DATA0.DO and output to DAC. Each time a trigger event occurs in the future, DATA0.DO will increment according to the step value set in DATA2.STEP;
- Step7.** When DATA0.DO increments to the upper limit set by DATA1.UPL, it will automatically reload DATA1.LWL to DATA0.DO, making it start incrementing again from the lower limit set by DATA1.LWL;
- Step8.** If an interrupt is required when the sawtooth wave reloads, first set the interrupt enable bit (INTEN) in the DCU control register (DCU\_CTL) to inactive, then set the values of DATA0-DATA2. Clear the DCU flag bit once before starting the timer to generate trigger events, then set the interrupt enable bit (INTEN) in the DCU control register to active before starting the timer;
- Step9.** Set the MODE in DCU control register (DCU\_CTL) to 0b0000 to exit sawtooth wave output mode.

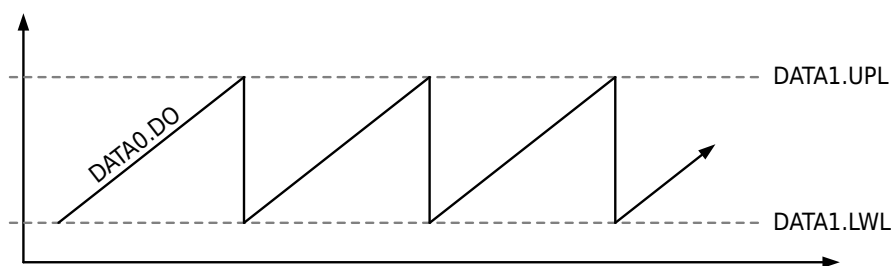


Figure 44-4 Incremental Sawtooth Wave Output Mode Waveform

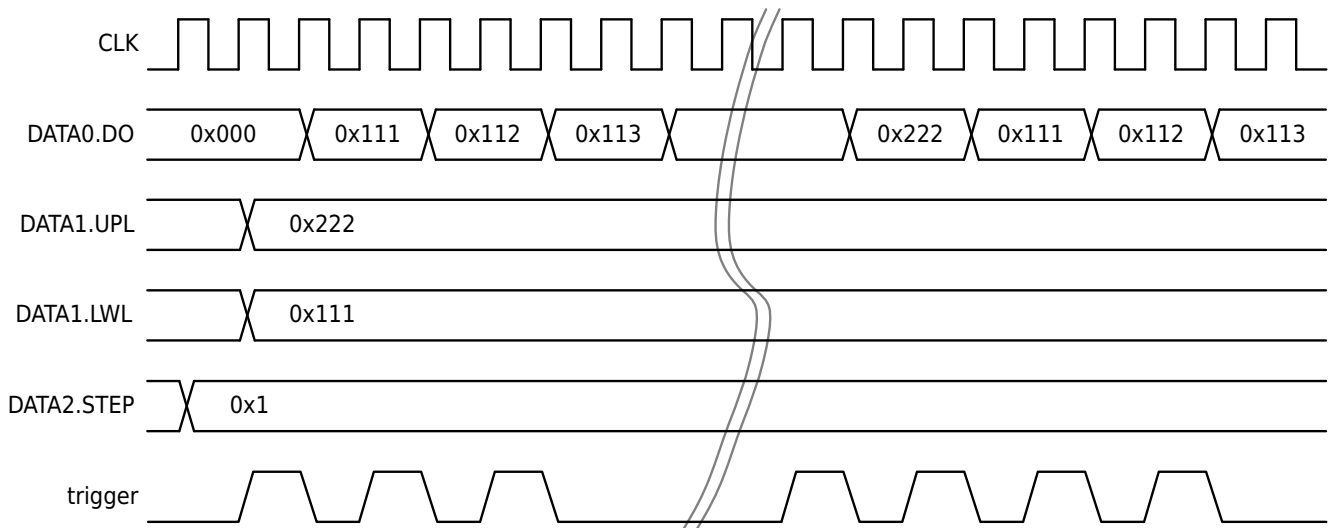


Figure 44-5 Incremental Sawtooth Wave Output Mode

### 44.3.7 Decrement Sawtooth Wave Output Mode

Decrement sawtooth wave output mode is where DCU uses a timer as the trigger source to provide a continuously changing digital value for the Digital-to-Analog Converter (DAC), which converts these digital values into a sawtooth wave output. Decrement sawtooth wave output mode is set through the DCU control register (DCU\_CTL). After setting to decrement sawtooth wave output mode, the functions of data registers DATA0-2 will change. The data register DATA0 will serve as the data register for the digital values output to the DAC and be reset. DATA0 becomes the output data register, with DATA.DO (Bit0-Bit11) as active bits. DATA1 becomes the amplitude setting register, with DATA1.UPL (Bit16-Bit27) and DATA1.LWL (Bit0-Bit11) as active bits. DATA2 becomes the step setting register, with DATA2.STEP (Bit0-Bit11) as active bits.



#### Note

Only DCU1-4 support decrement sawtooth wave output mode.

Steps for using decrement sawtooth wave output mode:

- Step1.** Set the MODE in DCU control register (DCU\_CTL) to 0b1010;
- Step2.** Set the upper and lower limits of the sawtooth wave amplitude. The upper limit is configured via DATA1.UPL[11:0], and the lower limit is configured via DATA1.LWL[11:0];
- Step3.** Set the decrement step value of the sawtooth wave through DATA2.STEP[11:0];
- Step4.** Select an appropriate timer event or software trigger as the trigger source through the trigger source selection register (DCU\_TRGSEL);
- Step5.** Configure the selected timer to generate trigger events at set time intervals;
- Step6.** Start the timer. On the first trigger event, the value of DATA1.UPL is loaded into DATA0.DO and output to DAC. Each time a trigger event occurs in the future, DATA0.DO will decrement according to the step value set in DATA2.STEP;
- Step7.** When DATA0.DO decrements to the lower limit set by DATA1.LWL, it will automatically reload DATA1.UPL to DATA0.DO, making it start decrementing again from the upper limit set by DATA1.UPL;
- Step8.** If an interrupt is required when the sawtooth wave reloads, first set the interrupt enable bit (INTEN) in the DCU control register (DCU\_CTL) to inactive, then set the values of DATA0-DATA2.

Clear the DCU flag bit once before starting the timer to generate trigger events, then set the interrupt enable bit (INTEN) in the DCU control register to active before starting the timer;

- Step9.** Set the MODE in DCU control register (DCU\_CTL) to 0b0000 to exit sawtooth wave output mode.

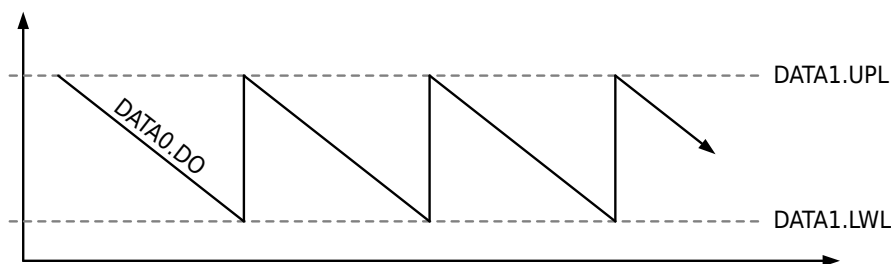


Figure 44-6 Decremental Sawtooth Wave Output Mode Waveform

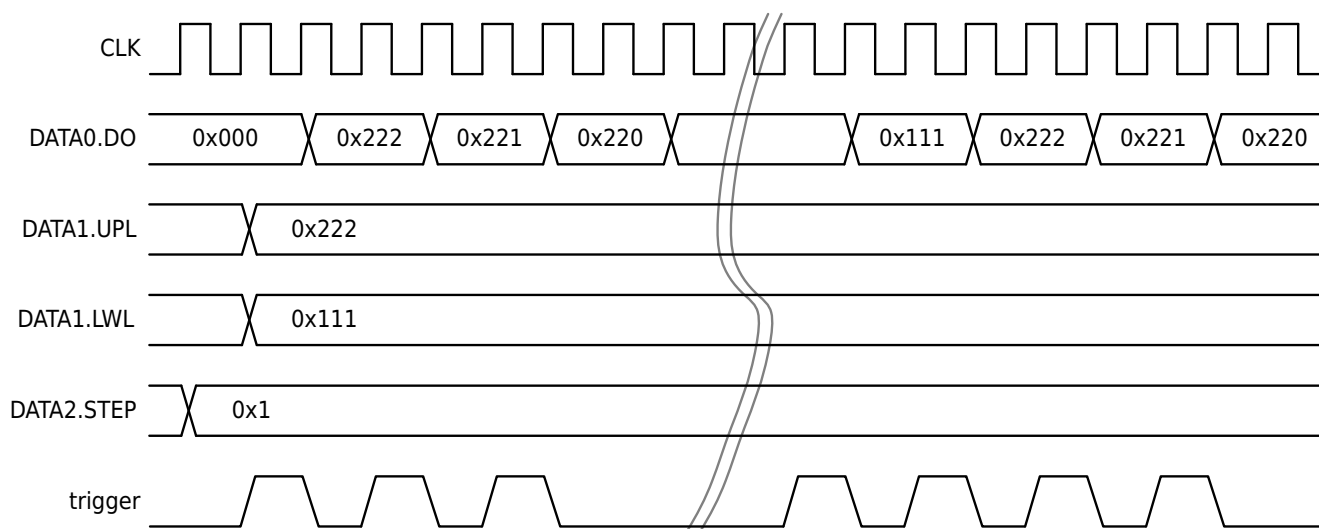


Figure 44-7 Decrement Sawtooth Wave Output Mode

### 44.3.8 Interrupt and Event

DCU has various interrupts and event outputs for triggering other peripheral circuits for user selection. The control of interrupts and event outputs is managed by the interrupt and event register (DCU\_INTEVTSEL). When an event signal needs to be output, users need to set the corresponding control bit in the interrupt and event register (DCU\_INTEVTSEL) to active. Each DCU unit outputs a DCU event signal, respectively DCU1-DCU8 in the event list. When a corresponding event needs to generate an interrupt, users need to set the corresponding control bit in the interrupt and event register (DCU\_INTEVTSEL) to active, and simultaneously set the DCU\_CTL.INTEN bit in the control register to 1. Each DCU unit outputs a DCU interrupt signal, respectively DCU1-DCU8 in the interrupt list.

## 44.4 Register Description

### 44.4.1 Register List

- DCU1 base address: 0x40056000
- DCU2 base address: 0x40056400
- DCU3 base address: 0x40056800
- DCU4 base address: 0x40056C00
- DCU5 base address: 0x40057000
- DCU6 base address: 0x40057400
- DCU7 base address: 0x40057800
- DCU8 base address: 0x40057C00

Table 44-1 Data Computing Unit (DCU) Register Offset

| Offset      | Register      | Description                                  | Width |
|-------------|---------------|----------------------------------------------|-------|
| 0x00        | DCU_CTL       | DCU Control Register                         | 32    |
| 0x04        | DCU_FLAG      | DCU Flag Register (DCU1-DCU4)                | 32    |
| 0x04        | DCU_FLAG      | DCU Flag Register (DCU5-DCU8)                | 32    |
| 0x08+0x04*x | DCU_DATAx     | DCU Data Register x (x=0-2)                  | 32    |
| 0x14        | DCU_FLAGCLR   | DCU Flag Reset Register (DCU1-DCU4)          | 32    |
| 0x14        | DCU_FLAGCLR   | DCU Flag Reset Register (DCU5-DCU8)          | 32    |
| 0x18        | DCU_INTEVTSEL | DCU Interrupt and Event Register (DCU1-DCU4) | 32    |
| 0x18        | DCU_INTEVTSEL | DCU Interrupt and Event Register (DCU5-DCU8) | 32    |



#### 44.4.2 DCU Control Register (DCU\_CTL)

This register is used to configure the action mode of DCU.

| Offset | Bit Position |          |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |         |          |   |          |      |   |   |   |   |   |
|--------|--------------|----------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---------|----------|---|----------|------|---|---|---|---|---|
| 0x00   | 31           | 30       | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9       | 8        | 7 | 6        | 5    | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x80000000   |          |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |         |          |   |          |      |   |   |   |   |   |
| Name   | INTEN        | Reserved |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | COMPTRG | Reserved |   | DATASIZE | MODE |   |   |   |   |   |
| Access | RW           |          |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RW      |          |   | RW       | RW   |   |   |   |   |   |

| Bit  | Name     | Function                                              | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | Access |
|------|----------|-------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31   | INTEN    | Interrupt enable                                      | <ul style="list-style-type: none"> <li>0b0: Disable</li> <li>0b1: Enable</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | RW     |
| 30:9 | Reserved | Reserved                                              | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | -      |
| 8    | COMPTRG  | Comparative mode trigger timing                       | <ul style="list-style-type: none"> <li>0b0: Compare after writing to DATA0</li> <li>0b1: Compare after writing to DATA0, DATA1, or DATA2</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | RW     |
| 7:6  | Reserved | Reserved                                              | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | -      |
| 5:4  | DATASIZE | Addition/subtraction mode, comparative mode data size | <ul style="list-style-type: none"> <li>0b00: 8-bit</li> <li>0b01: 16-bit</li> <li>0b10: 32-bit</li> <li>0b11: Setting prohibited</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | RW     |
| 3:0  | MODE     | Action mode                                           | <ul style="list-style-type: none"> <li>0b0000: DCU inactive</li> <li>0b0001: Addition mode, operation performed after data is written to the DATA1 register</li> <li>0b0010: Subtraction mode, operation performed after data is written to the DATA1 register</li> <li>0b0011: Hardware trigger addition mode, addition operation triggered by other peripheral circuits</li> <li>0b0100: Hardware trigger subtraction mode, subtraction operation triggered by other peripheral circuits</li> <li>0b0101: Comparative mode</li> <li>0b1000*: Triangle wave output mode</li> <li>0b1001*: Increment sawtooth wave output mode</li> <li>0b1010*: Decrement sawtooth wave output mode</li> <li>Others: Setting prohibited</li> </ul> | RW     |

| Bit | Name | Function    | Description                                                                                                                                                                          | Access |
|-----|------|-------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 3:0 | MODE | Action mode | <div><div></div><div><b>Note</b><br/>*:Only DCU1/DCU2/DCU3/DCU4 supports this function.</div></div> | RW     |

### 44.4.3 DCU Flag Register (DCU\_FLAG) (DCU1-DCU4)

This register generates the DCU result identifier.

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |          |          |          |          |          |          |          |          |          |          |         |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|---------|---|
| 0x04   | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11       | 10       | 9        | 8        | 7        | 6        | 5        | 4        | 3        | 2        | 1       | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |          |          |          |          |          |          |          |          |          |          |         |   |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | FLAG_TOP | FLAG_BTM | FLAG_RLD | Reserved | FLAG_GT1 | FLAG_EQ1 | FLAG_LS1 | FLAG_GT2 | FLAG_EQ2 | FLAG_LS2 | FLAG_OP |   |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RO       | RO       | RO       |          | RO       | RO       | RO       | RO       | RO       | RO       | RO      |   |

| Bit   | Name     | Function                    | Description                                                                                                                                                                                                                                                                    | Access |
|-------|----------|-----------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:12 | Reserved | Reserved                    | -                                                                                                                                                                                                                                                                              | -      |
| 11    | FLAG_TOP | Triangle wave peak points   | In triangle wave output mode, when the value of DATA0[11:0] reaches the maximum, the bit is set. It is cleared when the DCU_FLAGCLR.CLR_TOP bit is written as 1.                                                                                                               | RO     |
| 10    | FLAG_BTM | Triangle wave valley points | In triangle wave output mode, when the value of DATA0[11:0] reaches the minimum, the bit is set. It is cleared when the DCU_FLAGCLR.CLR_TOP bit is written as 1.                                                                                                               | RO     |
| 9     | FLAG_RLD | Sawtooth wave reload        | In increment sawtooth wave output mode, when the value of DATA0[11:0] reaches the maximum, the bit is set; in decrement sawtooth wave output mode, when the value of DATA0[11:0] reaches the minimum, the bit is set. It cleared when DCU_FLAGCLR.CLR_RLD bit is written as 1. | RO     |
| 8:7   | Reserved | Reserved                    | -                                                                                                                                                                                                                                                                              | -      |
| 6     | FLAG_GT1 | Greater than flag bit 1     | In comparison mode, the flag is set when DATA0 > DATA1, and the flag is cleared when the DCU_FLAGCLR.CLR_GT1 bit is written as 1.                                                                                                                                              | RO     |
| 5     | FLAG_EQ1 | Equal flag bit 1            | In comparative mode, when DATA0=DATA1, the bit is set. It is cleared when the DCU_FLAGCLR.CLR_EQ1 bit is written as 1.                                                                                                                                                         | RO     |
| 4     | FLAG_LS1 | Less than flag bit 1        | In comparative mode, when DATA0<DATA1, the bit is set. It is cleared when the DCU_FLAGCLR.CLR_LS1 bit is written as 1.                                                                                                                                                         | RO     |
| 3     | FLAG_GT2 | Greater than flag bit 2     | In comparison mode, the flag is set when DATA0 > DATA2, and the flag is cleared when the DCU_FLAGCLR.CLR_GT2 bit is written as 1.                                                                                                                                              | RO     |
| 2     | FLAG_EQ2 | Equal flag bit 2            | In comparative mode, when DATA0=DATA2, the bit is set. It is cleared when the DCU_FLAGCLR.CLR_EQ2 bit is written as 1.                                                                                                                                                         | RO     |

| Bit | Name     | Function             | Description                                                                                                                                                                                                                     | Access |
|-----|----------|----------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 1   | FLAG_LS2 | Less than flag bit 2 | In comparative mode, when $DATA0 < DATA2$ , the bit is set. It is cleared when the DCU_FLAGCLR.CLR_LS2 bit is written as 1.                                                                                                     | RO     |
| 0   | FLAG_OP  | Operation flag bit   | In addition/subtraction and trigger addition/trigger subtraction modes, when addition produces an overflow or subtraction produces an underflow, the bit is set. It is cleared when the DCU_FLAGCLR.CLR_OP bit is written as 1. | RO     |

#### 44.4.4 DCU Flag Register (DCU\_FLAG) (DCU5-DCU8)

This register generates the DCU result identifier.

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |          |    |          |    |          |    |          |    |          |  |          |  |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|----------|----|----------|----|----------|----|----------|----|----------|--|----------|--|
| 0x04   | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7        | 6  | 5        | 4  | 3        | 2  | 1        | 0  |          |  |          |  |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |          |    |          |    |          |    |          |    |          |  |          |  |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   | FLAG_GT1 |    | FLAG_EQ1 |    | FLAG_LS1 |    | FLAG_GT2 |    | FLAG_EQ2 |  | FLAG_LS2 |  |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   | RO       | RO | RO       | RO | RO       | RO | RO       | RO |          |  |          |  |

| Bit  | Name     | Function                | Description                                                                                                                                                                                                                     | Access |
|------|----------|-------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:7 | Reserved | Reserved                | -                                                                                                                                                                                                                               | -      |
| 6    | FLAG_GT1 | Greater than flag bit 1 | In comparison mode, the flag is set when DATA0 > DATA1, and the flag is cleared when the DCU_FLAGCLR.CLR_GT1 bit is written with 1.                                                                                             | RO     |
| 5    | FLAG_EQ1 | Equal flag bit 1        | In comparative mode, when DATA0=DATA1, the bit is set. It is cleared when the DCU_FLAGCLR.CLR_EQ1 bit is written as 1.                                                                                                          | RO     |
| 4    | FLAG_LS1 | Less than flag bit 1    | In comparative mode, when DATA0<DATA1, the bit is set. It is cleared when the DCU_FLAGCLR.CLR_LS1 bit is written as 1.                                                                                                          | RO     |
| 3    | FLAG_GT2 | Greater than flag bit 2 | In comparison mode, the flag is set when DATA0 > DATA2, and the flag is cleared when the DCU_FLAGCLR.CLR_GT2 bit is written with 1.                                                                                             | RO     |
| 2    | FLAG_EQ2 | Equal flag bit 2        | In comparative mode, when DATA0=DATA2, the bit is set. It is cleared when the DCU_FLAGCLR.CLR_EQ2 bit is written as 1.                                                                                                          | RO     |
| 1    | FLAG_LS2 | Less than flag bit 2    | In comparative mode, when DATA0<DATA2, the bit is set. It is cleared when the DCU_FLAGCLR.CLR_LS2 bit is written as 1.                                                                                                          | RO     |
| 0    | FLAG_OP  | Operation flag bit      | In addition/subtraction and trigger addition/trigger subtraction modes, when addition produces an overflow or subtraction produces an underflow, the bit is set. It is cleared when the DCU_FLAGCLR.CLR_OP bit is written as 1. | RO     |

#### 44.4.5 DCU Data Register (DCU\_DATAx) (x=0-2)

The functions of each data register in different modes are as follows:

| Mode Functions                         | DATA0           | DATA1              | DATA2              |
|----------------------------------------|-----------------|--------------------|--------------------|
| Addition mode                          | Summand/Result  | Addend             | Halving result     |
| Trigger addition mode                  | Summand/Result  | Addend             | Halving result     |
| Subtraction mode                       | Minuend/Result  | Subtrahend         | Halving result     |
| Trigger subtraction mode               | Minuend/Result  | Subtrahend         | Halving result     |
| Compare mode                           | Compared object | Compare object 1   | Compare object 2   |
| Compare mode (window compare)          | Compared object | Window upper limit | Window lower limit |
| Triangular wave output mode*           | Output data     | Amplitude setting  | Step size          |
| Incremental sawtooth wave output mode* | Output data     | Amplitude setting  | Step size          |
| Decrement sawtooth wave output mode*   | Output data     | Amplitude setting  | Step size          |



#### Note

\*: Only DCU1/DCU2/DCU3/DCU4 supports this function.

Offset: 0x08+0x04\*x

#### Addition/Subtraction Mode, Trigger Addition/Subtraction Mode, Comparative Mode

This register is used to store operation data.

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| '      | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | DAT          |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access | RW           |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit  | Name | Function       | Description                                                                                                                                                                                                                                                                                                                       | Access |
|------|------|----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:0 | DAT  | Operation data | Store operation data, the actual number of bits used depends on DCU_CTL.DATASIZE setting: <ul style="list-style-type: none"> <li>When DCU_CTL.DATASIZE=0b00, DATA[7:0] are valid data;</li> <li>When DCU_CTL.DATASIZE=0b01, DATA[15:0] are valid data;</li> <li>When DCU_CTL.DATASIZE=0b10, DATA[31:0] are valid data.</li> </ul> | RW     |

### Triangular Wave Output Mode

The register DCU\_DATA0 serves as the output data register, with Bit0-Bit11 as active bits.

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| -      | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | DO |    |   |   |   |   |   |   |   |   |   |   |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RW |    |   |   |   |   |   |   |   |   |   |   |

| Bit   | Name     | Function    | Description        | Access |
|-------|----------|-------------|--------------------|--------|
| 31:12 | Reserved | Reserved    | -                  | -      |
| 11:0  | DO       | Output data | Data output to DAC | RW     |

The register DCU\_DATA1 serves as the amplitude setting register, with Bit16-Bit27 and Bit0-Bit11 as active bits.

| Offset | Bit Position |    |    |    |     |    |    |    |    |    |    |    |    |    |    |    |          |    |    |    |     |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|-----|----|----|----|----|----|----|----|----|----|----|----|----------|----|----|----|-----|----|---|---|---|---|---|---|---|---|---|---|
| -      | 31           | 30 | 29 | 28 | 27  | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15       | 14 | 13 | 12 | 11  | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |     |    |    |    |    |    |    |    |    |    |    |    |          |    |    |    |     |    |   |   |   |   |   |   |   |   |   |   |
| Name   | Reserved     |    |    |    | UPL |    |    |    |    |    |    |    |    |    |    |    | Reserved |    |    |    | LWL |    |   |   |   |   |   |   |   |   |   |   |
| Access |              |    |    |    | RW  |    |    |    |    |    |    |    |    |    |    |    |          |    |    |    | RW  |    |   |   |   |   |   |   |   |   |   |   |

| Bit   | Name     | Function              | Description                                    | Access |
|-------|----------|-----------------------|------------------------------------------------|--------|
| 31:28 | Reserved | Reserved              | -                                              | -      |
| 27:16 | UPL      | Amplitude Upper Limit | Set the upper limit of triangle wave amplitude | RW     |
| 15:12 | Reserved | Reserved              | -                                              | -      |
| 11:0  | LWL      | Amplitude Lower Limit | Set the lower limit of triangle wave amplitude | RW     |

The register DCU\_DATA2 serves as the step register, with Bit0-Bit11 as active bits.

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |      |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|------|----|---|---|---|---|---|---|---|---|---|---|
| '      | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11   | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |      |    |   |   |   |   |   |   |   |   |   |   |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | STEP |    |   |   |   |   |   |   |   |   |   |   |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RW   |    |   |   |   |   |   |   |   |   |   |   |

| Bit   | Name     | Function     | Description                                                      | Access |
|-------|----------|--------------|------------------------------------------------------------------|--------|
| 31:12 | Reserved | Reserved     | -                                                                | -      |
| 11:0  | STEP     | Step setting | Set the increment and decrement step values of the triangle wave | RW     |

### Incremental/Decremental Sawtooth Wave Output Mode

The register DCU\_DATA0 serves as the output data register, with Bit0-Bit11 as active bits.

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| -      | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | DO |    |   |   |   |   |   |   |   |   |   |   |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit   | Name     | Function    | Description        | Access |
|-------|----------|-------------|--------------------|--------|
| 31:12 | Reserved | Reserved    | -                  | -      |
| 11:0  | DO       | Output data | Data output to DAC | RW     |



The register DCU\_DATA1 serves as the amplitude setting register, with Bit16-Bit27 and Bit0-Bit11 as active bits.

| Offset | Bit Position |    |    |    |     |    |    |    |    |    |    |    |    |    |    |    |          |    |    |    |     |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|-----|----|----|----|----|----|----|----|----|----|----|----|----------|----|----|----|-----|----|---|---|---|---|---|---|---|---|---|---|
| -      | 31           | 30 | 29 | 28 | 27  | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15       | 14 | 13 | 12 | 11  | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |     |    |    |    |    |    |    |    |    |    |    |    |          |    |    |    |     |    |   |   |   |   |   |   |   |   |   |   |
| Name   | Reserved     |    |    |    | UPL |    |    |    |    |    |    |    |    |    |    |    | Reserved |    |    |    | LWL |    |   |   |   |   |   |   |   |   |   |   |
| Access |              |    |    |    | RW  |    |    |    |    |    |    |    |    |    |    |    |          |    |    |    | RW  |    |   |   |   |   |   |   |   |   |   |   |

| Bit   | Name     | Function              | Description                                    | Access |
|-------|----------|-----------------------|------------------------------------------------|--------|
| 31:28 | Reserved | Reserved              | -                                              | -      |
| 27:16 | UPL      | Amplitude Upper Limit | Set the upper limit of sawtooth wave amplitude | RW     |
| 15:12 | Reserved | Reserved              | -                                              | -      |
| 11:0  | LWL      | Amplitude Lower Limit | Set the lower limit of sawtooth wave amplitude | RW     |

The register DCU\_DATA2 serves as the step register, with Bit0-Bit11 as active bits.

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |      |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| -      | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15   | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |      |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | STEP |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |      |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit   | Name     | Function     | Description                                                    | Access |
|-------|----------|--------------|----------------------------------------------------------------|--------|
| 31:12 | Reserved | Reserved     | -                                                              | -      |
| 11:0  | STEP     | Step setting | Set the increment or decrement step value of the sawtooth wave | RW     |

#### 44.4.6 DCU Flag Reset Register (DCU\_FLAGCLR) (DCU1-DCU4)

This register is used to clear the DCU result identifier.

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |         |         |         |          |         |         |         |         |         |         |        |    |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---------|---------|---------|----------|---------|---------|---------|---------|---------|---------|--------|----|
| 0x14   | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11      | 10      | 9       | 8        | 7       | 6       | 5       | 4       | 3       | 2       | 1      | 0  |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |         |         |         |          |         |         |         |         |         |         |        |    |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | CLR_TOP | CLR_BTM | CLR_RLD | Reserved | CLR_GT1 | CLR_EQ1 | CLR_LS1 | CLR_GT2 | CLR_EQ2 | CLR_LS2 | CLR_OP |    |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | WO      | WO      | WO      |          | WO      | WO      | WO      | WO      | WO      | WO      | WO     | WO |

| Bit   | Name     | Function                            | Description                                                        | Access |
|-------|----------|-------------------------------------|--------------------------------------------------------------------|--------|
| 31:12 | Reserved | Reserved                            | -                                                                  | -      |
| 11    | CLR_TOP  | Clear Triangle Wave Peak Flag Bit   | Write 1 to clear the DCU_FLAG.FLAG_TOP bit, write 0 has no effect. | WO     |
| 10    | CLR_BTM  | Clear Triangle Wave Valley Flag Bit | Write 1 to clear the DCU_FLAG.FLAG_BTM bit, write 0 has no effect. | WO     |
| 9     | CLR_RLD  | Clear Sawtooth Wave Reload Flag Bit | Write 1 to clear the DCU_FLAG.FLAG_RLD bit, write 0 has no effect. | WO     |
| 8:7   | Reserved | Reserved                            | -                                                                  | -      |
| 6     | CLR_GT1  | Clear Greater Than Flag Bit 1       | Write 1 to clear the DCU_FLAG.FLAG_GT1 bit, write 0 has no effect. | WO     |
| 5     | CLR_EQ1  | Clear Equals Flag Bit 1             | Write 1 to clear the DCU_FLAG.FLAG_EQ1 bit, write 0 has no effect. | WO     |
| 4     | CLR_LS1  | Clear Less Than Flag Bit 1          | Write 1 to clear the DCU_FLAG.FLAG_LS1 bit, write 0 has no effect. | WO     |
| 3     | CLR_GT2  | Clear Greater Than Flag Bit 2       | Write 1 to clear the DCU_FLAG.FLAG_GT2 bit, write 0 has no effect. | WO     |
| 2     | CLR_EQ2  | Clear Equals Flag Bit 2             | Write 1 to clear the DCU_FLAG.FLAG_EQ2 bit, write 0 has no effect. | WO     |
| 1     | CLR_LS2  | Clear Less Than Flag Bit 2          | Write 1 to clear the DCU_FLAG.FLAG_LS2 bit, write 0 has no effect. | WO     |
| 0     | CLR_OP   | Clear Operation Flag Bit            | Write 1 to clear the DCU_FLAG.FLAG_OP bit, write 0 has no effect.  | WO     |

#### 44.4.7 DCU Flag Reset Register (DCU\_FLAGCLR) (DCU5-DCU8)

This register is used to clear the DCU result identifier.

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |         |         |         |         |         |         |        |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---------|---------|---------|---------|---------|---------|--------|---|
| 0x14   | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7       | 6       | 5       | 4       | 3       | 2       | 1      | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |         |         |         |         |         |         |        |   |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   | CLR_GT1 | CLR_EQ1 | CLR_LS1 | CLR_GT2 | CLR_EQ2 | CLR_LS2 | CLR_OP |   |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   | WO      | WO      | WO      | WO      | WO      | WO      | WO     |   |

| Bit  | Name     | Function                      | Description                                                        | Access |
|------|----------|-------------------------------|--------------------------------------------------------------------|--------|
| 31:7 | Reserved | Reserved                      | -                                                                  | -      |
| 6    | CLR_GT1  | Clear Greater Than Flag Bit 1 | Write 1 to clear the DCU_FLAG.FLAG_GT1 bit, write 0 has no effect. | WO     |
| 5    | CLR_EQ1  | Clear Equals Flag Bit 1       | Write 1 to clear the DCU_FLAG.FLAG_EQ1 bit, write 0 has no effect. | WO     |
| 4    | CLR_LS1  | Clear Less Than Flag Bit 1    | Write 1 to clear the DCU_FLAG.FLAG_LS1 bit, write 0 has no effect. | WO     |
| 3    | CLR_GT2  | Clear Greater Than Flag Bit 2 | Write 1 to clear the DCU_FLAG.FLAG_GT2 bit, write 0 has no effect. | WO     |
| 2    | CLR_EQ2  | Clear Equals Flag Bit 2       | Write 1 to clear the DCU_FLAG.FLAG_EQ2 bit, write 0 has no effect. | WO     |
| 1    | CLR_LS2  | Clear Less Than Flag Bit 2    | Write 1 to clear the DCU_FLAG.FLAG_LS2 bit, write 0 has no effect. | WO     |
| 0    | CLR_OP   | Clear Operation Flag Bit      | Write 1 to clear the DCU_FLAG.FLAG_OP bit, write 0 has no effect.  | WO     |

#### 44.4.8 DCU Interrupt and Event Register (DCU\_INTEVTSEL) (DCU1-DCU4)

This register can select under what conditions DCU generates interrupts and outputs event signals.

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |         |         |         |         |         |         |         |         |         |         |        |    |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|--------|----|---|---|---|---|---|---|
| 0x18   | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17      | 16      | 15      | 14      | 13      | 12      | 11      | 10      | 9       | 8       | 7      | 6  | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |         |         |         |         |         |         |         |         |         |         |        |    |   |   |   |   |   |   |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    | SEL_TOP | SEL_BTM | SEL_RLD | SEL_WIN | SEL_GT1 | SEL_EQ1 | SEL_LS1 | SEL_GT2 | SEL_EQ2 | SEL_LS2 | SEL_OP |    |   |   |   |   |   |   |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    | RW      | RW      | RW      | RW      | RW      | RW      | RW      | RW      | RW      | RW      | RW     | RW |   |   |   |   |   |   |

| Bit   | Name     | Function                                       | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                 | Access |
|-------|----------|------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:12 | Reserved | Reserved                                       | -                                                                                                                                                                                                                                                                                                                                                                                                                                                           | -      |
| 11    | SEL_TOP  | Triangle Wave Peak Point Condition Selection   | <ul style="list-style-type: none"> <li>0b0: In triangle wave output mode, when DATA0[11:0] reaches the maximum, no interrupt is generated and no event signal is output</li> <li>0b1: In triangle wave output mode, when DATA0[11:0] reaches the maximum, an interrupt is generated and an event signal is output</li> </ul>                                                                                                                                | RW     |
| 10    | SEL_BTM  | Triangle Wave Valley Point Condition Selection | <ul style="list-style-type: none"> <li>0b0: In triangle wave output mode, when DATA0[11:0] reaches the minimum, no interrupt is generated and no event signal is output</li> <li>0b1: In triangle wave output mode, when DATA0[11:0] reaches the minimum, an interrupt is generated and an event signal is output</li> </ul>                                                                                                                                | RW     |
| 9     | SEL_RLD  | Sawtooth Wave Reload Condition Selection       | <ul style="list-style-type: none"> <li>0b0: In increment/decrement sawtooth wave output mode, no interrupt is generated and no event signal is output</li> <li>0b1: In increment sawtooth wave output mode, when DATA0[11:0] reaches the maximum, an interrupt is generated and an event signal is output; in decrement sawtooth wave output mode, when DATA0[11:0] reaches the minimum, an interrupt is generated and an event signal is output</li> </ul> | RW     |

| Bit | Name    | Function                           | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | Access |
|-----|---------|------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 8:7 | SEL_WIN | Window Compare Condition Selection | <p>In comparative mode, when the window comparison condition set by SEL_WIN is met, an interrupt is generated and an event signal is output. When SEL_WIN is set active, no interrupt or event signal will be generated when other comparison conditions are met:</p> <ul style="list-style-type: none"> <li>● 0b00: No window comparison interrupt is generated and no event signal is output. Under this setting, other interrupt and event signal generation conditions are selected by Bit1-Bit6 of this register</li> <li>● 0b01: When DATA0 data is within the window, an interrupt is generated and an event signal is output, i.e., <math>DATA2 \leq DATA0 \leq DATA1</math></li> <li>● 0b10: When DATA0 data is outside the window, an interrupt is generated and an event signal is output, i.e., <math>DATA0 &gt; DATA1</math> or <math>DATA0 &lt; DATA2</math></li> <li>● 0b11: In comparative mode, no interrupt and event signal is generated</li> </ul> | RW     |
| 6   | SEL_GT1 | Greater Than Condition Selection 1 | <ul style="list-style-type: none"> <li>● 0b0: In comparative mode and when SEL_WIN=0b00, when <math>DATA0 &gt; DATA1</math>, no interrupt is generated and no event signal is output</li> <li>● 0b1: In compare Mode with SEL_WIN=0b00, an interrupt and output event signal will be generated when <math>DATA0 &gt; DATA1</math>.</li> </ul> <div>  <b>Note</b><br/>           When SEL_WIN≠0b00, this bit setting has no effect.         </div>                                                                                                                                                                                                                                                                                                                                                                                                                                   | RW     |
| 5   | SEL_EQ1 | Equal Condition Selection 1        | <ul style="list-style-type: none"> <li>● 0b0: In comparative mode with SEL_WIN=0b00, no interrupt and no event output is generated when <math>DATA0 = DATA1</math></li> <li>● 0b1: In comparative mode with SEL_WIN=0b00, an interrupt and output event signal will be generated when <math>DATA0 = DATA1</math>.</li> </ul> <div>  <b>Note</b><br/>           When SEL_WIN≠0b00, this bit setting has no effect.         </div>                                                                                                                                                                                                                                                                                                                                                                                                                                                    | RW     |
| 4   | SEL_LS1 | Less Than Condition Selection 1    | <ul style="list-style-type: none"> <li>● 0b0: In comparative mode with SEL_WIN=0b00, no interrupt and no event output is generated when <math>DATA0 &lt; DATA1</math></li> <li>● 0b1: In comparative mode with SEL_WIN=0b00, an interrupt and event output are generated when <math>DATA0 &lt; DATA1</math></li> </ul> <div>  <b>Note</b><br/>           When SEL_WIN≠0b00, this bit setting has no effect.         </div>                                                                                                                                                                                                                                                                                                                                                                                                                                                          | RW     |

| Bit | Name    | Function                           | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | Access |
|-----|---------|------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 3   | SEL_GT2 | Greater Than Condition Selection 2 | <ul style="list-style-type: none"> <li>0b0: In comparative mode and when SEL_WIN=0b00, when DATA0&gt;DATA2, no interrupt is generated and no event signal is output</li> <li>0b1: In comparative mode with SEL_WIN=0b00, an interrupt and output event signal will be generated when DATA0 &gt; DATA2.</li> </ul> <div>  <b>Note</b><br/>           When SEL_WIN≠0b00, this bit setting has no effect.         </div> | RW     |
| 2   | SEL_EQ2 | Equal Condition Selection 2        | <ul style="list-style-type: none"> <li>0b0: In comparative mode with SEL_WIN=0b00, no interrupt and no event output is generated when DATA0=DATA2</li> <li>0b1: In comparative mode with SEL_WIN=0b00, an interrupt and output event signal will be generated when DATA0=DATA2.</li> </ul> <div>  <b>Note</b><br/>           When SEL_WIN≠0b00, this bit setting has no effect.         </div>                        | RW     |
| 1   | SEL_LS2 | Less Than Condition Selection 2    | <ul style="list-style-type: none"> <li>0b0: In comparative mode with SEL_WIN=0b00, no interrupt and no event output is generated when DATA0&lt;DATA2</li> <li>0b1: In comparative mode with SEL_WIN=0b00, an interrupt and event output are generated when DATA0&lt;DATA2</li> </ul> <div>  <b>Note</b><br/>           When SEL_WIN≠0b00, this bit setting has no effect.         </div>                            | RW     |
| 0   | SEL_OP  | Operation Condition Selection      | <ul style="list-style-type: none"> <li>0b0: In addition/subtraction modes, no interrupt and no event output is generated on arithmetic overflow/underflow</li> <li>0b1: In addition/subtraction modes, an interrupt and event output are generated on arithmetic overflow/underflow</li> </ul>                                                                                                                                                                                                         | RW     |

#### 44.4.9 DCU Interrupt and Event Register (DCU\_INTEVTSEL) (DCU5-DCU8)

This register can select under what conditions DCU generates interrupts and outputs event signals.

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |         |         |         |         |         |         |         |        |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---------|---------|---------|---------|---------|---------|---------|--------|
| 0x18   | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7       | 6       | 5       | 4       | 3       | 2       | 1       | 0      |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |         |         |         |         |         |         |         |        |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   | SEL_WIN | SEL_GT1 | SEL_EQ1 | SEL_LS1 | SEL_GT2 | SEL_EQ2 | SEL_LS2 | SEL_OP |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   | RW      | RW      | RW      | RW      | RW      | RW      | RW      | RW     |

| Bit  | Name     | Function                           | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | Access |
|------|----------|------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:9 | Reserved | Reserved                           | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | -      |
| 8:7  | SEL_WIN  | Window compare condition selection | <p>In comparative mode, when the window comparison condition set by SEL_WIN is met, an interrupt is generated and an event output signal is output. When SEL_WIN is set active, no interrupt and no event output signal will be generated when other comparison conditions are met:</p> <ul style="list-style-type: none"> <li>0b00: No window comparison interrupt is generated and no event output signal is output. Under this setting, other interrupt and event signal generation conditions are selected by Bit1-Bit6 of this register</li> <li>0b01: When DATA0 data is within the window, an interrupt is generated and an event signal is output, i.e., <math>DATA2 \leq DATA0 \leq DATA1</math></li> <li>0b10: When DATA0 data is outside the window, an interrupt is generated and an event signal is output, i.e., <math>DATA0 &gt; DATA1</math> or <math>DATA0 &lt; DATA2</math></li> <li>0b11: In comparative mode, no interrupt and no event signal is generated</li> </ul> | RW     |
| 6    | SEL_GT1  | Greater than condition selection 1 | <ul style="list-style-type: none"> <li>0b0: In comparative mode with SEL_WIN=0b00, when <math>DATA0 &gt; DATA1</math>, no interrupt is generated and no event signal is output</li> <li>0b1: In comparative mode with SEL_WIN=0b00, an interrupt and output event signal will be generated when <math>DATA0 &gt; DATA1</math>.</li> </ul> <div>  <b>Note</b><br/>           When SEL_WIN≠0b00, this bit setting has no effect.         </div>                                                                                                                                                                                                                                                                                                                                                                                                                                                           | RW     |

| Bit | Name    | Function                        | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | Access |
|-----|---------|---------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 5   | SEL_EQ1 | Equal condition selection 1     | <ul style="list-style-type: none"> <li>0b0: In comparative mode with SEL_WIN=0b00, no interrupt and no event output is generated when DATA0=DATA1</li> <li>0b1: In comparative mode with SEL_WIN=0b00, an interrupt and output event signal will be generated when DATA0=DATA1.</li> </ul> <div>  <b>Note</b><br/>When SEL_WIN≠0b00, this bit setting has no effect.         </div>                      | RW     |
| 4   | SEL_LS1 | Less than condition selection 1 | <ul style="list-style-type: none"> <li>0b0: In comparative mode with SEL_WIN=0b00, no interrupt and no event output is generated when DATA0&lt;DATA1</li> <li>0b1: In comparative mode with SEL_WIN=0b00, an interrupt and event output are generated when DATA0&lt;DATA1</li> </ul> <div>  <b>Note</b><br/>When SEL_WIN≠0b00, this bit setting has no effect.         </div>                            | RW     |
| 3   | SEL_GT2 | Greater than condition select 2 | <ul style="list-style-type: none"> <li>0b0: In comparative mode with SEL_WIN=0b00, when DATA0&gt;DATA2, no interrupt is generated and no event signal is output</li> <li>0b1: In comparative mode with SEL_WIN=0b00, an interrupt and output event signal will be generated when DATA0 &gt; DATA2.</li> </ul> <div>  <b>Note</b><br/>When SEL_WIN≠0b00, this bit setting has no effect.         </div> | RW     |
| 2   | SEL_EQ2 | Equal condition selection 2     | <ul style="list-style-type: none"> <li>0b0: In comparative mode with SEL_WIN=0b00, no interrupt and no event output is generated when DATA0=DATA2</li> <li>0b1: In comparative mode with SEL_WIN=0b00, an interrupt and output event signal will be generated when DATA0=DATA2.</li> </ul> <div>  <b>Note</b><br/>When SEL_WIN≠0b00, this bit setting has no effect.         </div>                    | RW     |
| 1   | SEL_LS2 | Less Than condition Selection 2 | <ul style="list-style-type: none"> <li>0b0: In comparative mode with SEL_WIN=0b00, no interrupt and no event output is generated when DATA0&lt;DATA2</li> <li>0b1: In comparative mode with SEL_WIN=0b00, an interrupt and event output are generated when DATA0&lt;DATA2</li> </ul> <div>  <b>Note</b><br/>When SEL_WIN≠0b00, this bit setting has no effect.         </div>                          | RW     |



| Bit | Name   | Function                      | Description                                                                                                                                                                                                                                                                                     | Access |
|-----|--------|-------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 0   | SEL_OP | Operation condition selection | <ul style="list-style-type: none"><li>● 0b0: In addition/subtraction modes, no interrupt and no event output is generated on arithmetic overflow/underflow</li><li>● 0b1: In addition/subtraction modes, an interrupt and event output are generated on arithmetic overflow/underflow</li></ul> | RW     |

## 45 Math Arithmetical Unit (MAU)

### 45.1 Introduction

Math Arithmetical Unit (MAU) is a hardware accelerated operation module which contains two types of operations: root operation and sine operation, and supports root operation and sine operation of fixed points. The sine function supports  $360^\circ/2^{12}$  calculation accuracy.

### 45.2 Functional Description

#### 45.2.1 Square Root Operation

The square root operation supports 32-bit fixed-point input and outputs a 32-bit fixed-point result. The execution period varies depending on the input data value.

##### Operating Procedures

When the interrupt is disabled (MAU\_CSR.INTEN=0), the square root operation follows the steps below. When the interrupt is enabled, users may customize the process by adding interrupt handling:

- Step1.** Write the radicand (number to be square-rooted) into MAU\_DTR0 register;
- Step2.** Set the MAU\_CSR.START bit to 1 to trigger the operation;
- Step3.** The MAU\_CSR.BUSY bit will be set during computation. When the calculation is complete, this bit is automatically cleared; Wait until the BUSY bit is cleared, indicating the computation has completed;
- Step4.** Read the result from MAU\_RTR0 register;
- Step5.** Repeat steps 1-4 as needed.

Note: In step 3, after setting the MAU\_CSR.START bit, the BUSY bit rises and remains high for 8 to 16 calculation clock periods, then automatically goes to 0, with the calculation time varying depending on the input data size. Users can monitor this bit to track computation status. When using the square root operation module, users can flexibly judge the current operation status according to the changes of this bit.

##### Result Handling

The result of the square root operation is a 32-bit value. The least significant bit is rounded to the nearest integer using the "round half up" method.

For example: Input=0x0000000C → Output=0x00000003; Input=0x0000000D → Output=0x00000004.

##### Fixed-Point Number Processing

When using fixed-point decimal input and needing consistent formatting between input and output, users can configure the MAU\_CSR.SHIFT bit to adjust the output accordingly.

For example, when using the Q14.2 format for fixed-point arithmetic, you can set MAU\_CSR.SHIFT=1 so that the output results also follow the Q14.2 format. The fixed-point number 6.25 (0x00000019 in Q14.2 format) has an output result of 5 (0x00000005) before shifting, and a result of 2.5 (0x0000000A in Q14.2 format) after shifting 1 bit.

**Note**

When the exponent data written to MAU\_DTR0 is greater than 0xFFFF0000, the maximum value of MAU\_CSR.SHIFT can be set to 15; when the exponent data written to MAU\_DTR0 is less than or equal to 0xFFFF0000, the maximum value of MAU\_CSR.SHIFT can be set to 16.

### 45.2.2 Sine Operation

The sine operation supports 12-bit data input and the operation result is a 16-bit signed number output. The 12-bit data input represents the rectangular coordinate system angle to be calculated; in the 16-bit data output, the highest bit represents the positive or negative sign of the result.

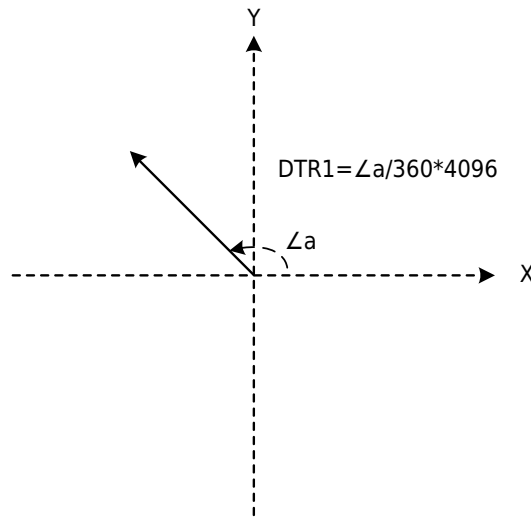


Figure 45-1 Schematic Diagram Of Sine Operation Angle Value

### Operating Procedures

The operation flow of sine operation is as follows:

- Step1.** Write the angle value in MAU\_DTR1 register;
- Step2.** After 1 period, read the operation result from MAU\_RTR1 register;
- Step3.** Repeat the above steps 1-2.

### 45.2.3 Interrupt and Event Output

#### Interrupt Output

After each square root operation is completed, a corresponding operation completion event will be generated. If the interrupt control bit is enabled (MAU\_CSR.INTEN=1), the corresponding interrupt request signal (MAU\_SQRT) will also be generated.

#### Event Output

After each square root operation is completed, a corresponding operation completion event will be generated, and the corresponding event request signal (MAU\_SQRT) will also be output, which can be used to select and trigger other modules.

## 45.3 Register Description

### 45.3.1 Register List

Base address: 0x40055000

Table 45-1 Math Unit (MAU) Register Offset

| Offset | Register | Description              | Width |
|--------|----------|--------------------------|-------|
| 0x0000 | MAU_CSR  | Control Status Register  | 32    |
| 0x0004 | MAU_DTR0 | Data Input Register 0    | 32    |
| 0x000C | MAU_RTR0 | Result Output Register 0 | 32    |
| 0x0010 | MAU_DTR1 | Data Input Register 1    | 32    |
| 0x0014 | MAU_RTR1 | Result Output Register 1 | 32    |

## 45.3.2 Control State Register (MAU\_CSR)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |       |    |    |    |          |    |   |   |      |   |          |   |       |   |       |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-------|----|----|----|----------|----|---|---|------|---|----------|---|-------|---|-------|---|
| 0x0000 | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15    | 14 | 13 | 12 | 11       | 10 | 9 | 8 | 7    | 6 | 5        | 4 | 3     | 2 | 1     | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |       |    |    |    |          |    |   |   |      |   |          |   |       |   |       |   |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | SHIFT |    |    |    | Reserved |    |   |   | BUSY |   | Reserved |   | INTEN |   | START |   |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RW    |    |    |    |          |    |   |   | RO   |   |          |   | RW    |   | RW    |   |

| Bit   | Name     | Function                     | Description                                                                                                                                                                                                                                                                                                                                               | Access |
|-------|----------|------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:13 | Reserved | Reserved                     | -                                                                                                                                                                                                                                                                                                                                                         | -      |
| 12:8  | SHIFT    | Shift the square root result | <ul style="list-style-type: none"> <li>0b00000: The result is output without shifted</li> <li>0b00001: The result is output with shifted left by 1 bit</li> <li>0b00010: The result is output with shifted left by 2 bit</li> <li>.....</li> <li>0b10000: The result is output with shifted left by 16 bit</li> <li>Others: Setting prohibited</li> </ul> | RW     |
| 7:4   | Reserved | Reserved                     | -                                                                                                                                                                                                                                                                                                                                                         | -      |
| 3     | BUSY     | Square root operation status | <ul style="list-style-type: none"> <li>0b0: The square root operation has not started or is completed</li> <li>0b1: Square root operation in progress</li> </ul>                                                                                                                                                                                          | RO     |
| 2     | Reserved | Reserved                     | -                                                                                                                                                                                                                                                                                                                                                         | -      |
| 1     | INTEN    | Square root interrupt enable | <ul style="list-style-type: none"> <li>0b0: No interrupt will be generated after the square root operation is completed</li> <li>0b1: An interrupt is generated after the square root operation is completed</li> </ul>                                                                                                                                   | RW     |
| 0     | START    | Square root operation starts | Write 1 starts the square root operation (generating a START request internally), write 0 has no effect; read always returns 0.                                                                                                                                                                                                                           | RW     |

45.3.3 Data Input Register 0 (MAU\_DTR0)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0x0004 | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | SQRT_DIN     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access | RW           |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit  | Name     | Function | Description                       | Access |
|------|----------|----------|-----------------------------------|--------|
| 31:0 | SQRT_DIN | Radicand | Input of the number to be squared | RW     |

## 45.3.4 Result Output Register 0 (MAU\_RTR0)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0x000C | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | SQRT_DOUT    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access | RW           |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit  | Name      | Function           | Description               | Access |
|------|-----------|--------------------|---------------------------|--------|
| 31:0 | SQRT_DOUT | Square Root Result | Square root result output | RW     |

## 45.3.5 Data Input Register 1 (MAU\_DTR1)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |         |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0x0010 | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15      | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |         |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | SIN_DIN |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RW      |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit   | Name     | Function      | Description                                | Access |
|-------|----------|---------------|--------------------------------------------|--------|
| 31:12 | Reserved | Reserved      | -                                          | -      |
| 11:0  | SIN_DIN  | Angle Setting | Angle input in Cartesian coordinate system | RW     |

45.3.6 Result Output Register 1 (MAU\_RTR1)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |          |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0x0014 | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15       | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |          |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | SIN_DOUT |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RW       |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit   | Name     | Function              | Description                  | Access |
|-------|----------|-----------------------|------------------------------|--------|
| 31:16 | Reserved | Reserved              | -                            | -      |
| 15:0  | SIN_DOUT | Sine operation result | Sine operation result output | RW     |



## 46 Filter Math Accelerator (FMAC)

### 46.1 Introduction

Filter Math Accelerator (FMAC) is a hardware acceleration module for FIR filter calculation. The module can carry out FIR digital filtering with a maximum of 16 orders and configurable orders. Built-in 16x16bits multiplier and 32+5bits adder, users can customize the output data precision. This series of products is equipped with 4 FMAC modules.

### 46.2 Features

- Maximum 16-order FIR filter, configurable order
- Configurable filter coefficients
- 16 -bit signed input and output data, 16 -bit signed filter coefficients
- 16x16bits multiplier
- 32+5bits adder
- Provide interrupt signal and status query
- AHB slave interface configuration parameters and input and output data

### 46.3 Functional Description

#### 46.3.1 Basic Block Diagram

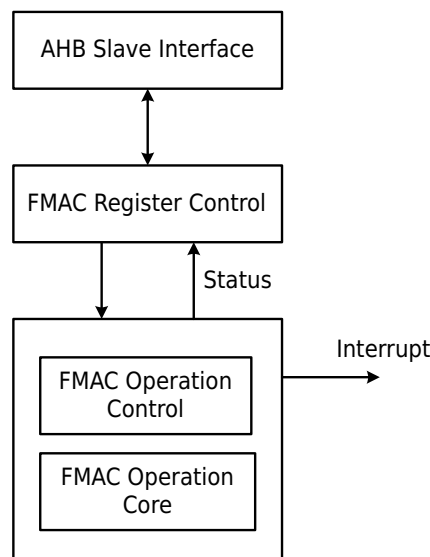


Figure 46-1 FMAC Basic Block Diagram

When the 16th-order FIR filter is selected, the FMAC operation control and operation core requires 17 clock periods to process one data. At an operating clock frequency of 240MHz, it can process a maximum input data stream of 14.1MHz. Different orders require different processing times. The smaller the order, the faster the processing speed. The relationship between the maximum input data frequency, the order and the operating clock frequency is as follows:

Maximum input data frequency (MHz)=operating clock frequency (MHz) ÷(order + 1)

### 46.3.2 Operating Procedures

This module needs to be used according to the following steps:

- Step1.** Set the enable bit (FMAC\_ENR.FMACEN) to 1;
- Step2.** Configure the filtering order (FMAC\_CTR.STAGE\_NUM) and the number of shifts to output active results (FMAC\_CTR.SHIFT);
- Step3.** If you need to use interrupts, configure the interrupt enable bit (FMAC\_IER.INTEN) to 1;
- Step4.** Write filter coefficients (if there are N orders, just configure FMAC\_COR0-FMAC\_CORN);
- Step5.** Write data to be processed (FMAC\_DTR);
- Step6.** Wait for the completion interrupt or query the completion status (FMAC\_STR.READY) to be 1;
- Step7.** Read result (FAMC\_RTR0 or FAMC\_RTR1);
- Step8.** Repeat steps 5 to 7 until all data has been processed.

### 46.3.3 Module Enable

The module enable bit (FMAC\_ENR.FMACEN) has two functions. The first function is to enable the entire module. Only when it is 1, the module is enabled to work. The second function is the soft reset of the module. When a certain order and coefficient configuration is used up, it is necessary to change to another order and coefficient configuration. In this case, FMACEN needs to be set to 0, the residual information of the previous calculation is cleared, and then the operation is carried out according to the operation process.

FMACEN is reset to 0, the configuration parameters are not cleared, only the internal registers and intermediate results are cleared.

### 46.3.4 Coefficient Normalization

If the output result is only 16 bits, then the total number of bits truncated during the calculation process and the calculation result is as high as 20 bits. Therefore, if the input data involved in the calculation is too small, the output result may be 0 or a very small number (with a large loss of precision). Therefore, it is necessary to normalize the filter coefficients, that is, normalize the maximum value (if it is a positive number) to 32767 (if it is a negative number, normalize it to -32768 ), and multiply the other coefficients by the same ratio.

For example, there are 5 coefficients 10, 13, 35, 96, 42, and they need to be normalized. First find the maximum value, which is 96, and then divide 32767 by the maximum value,  $32767/96=341.3$ , and take the value 341, which is the normalized multiple. All other coefficients must be multiplied by this multiple to get the normalized number.

So the normalized coefficients are: 3410, 4433, 11935, 32736, 14322.

### 46.3.5 Interrupt and Events

#### 46.3.5.1 Interrupt Output

After each operation is completed, a corresponding operation completion event will be generated. If the interrupt control bit is enabled ( FMAC\_IER.INTEN=1 ), the corresponding interrupt request signal ( FMAC\_m\_FIR, m=1-4 ) will also be generated.

#### 46.3.5.2 Event Output

After each operation is completed, a corresponding operation completion event will be generated, and the corresponding event request signal (FMAC\_m\_FIR, m=1-4) will also be output, which can be used to select and trigger other modules.

## 46.4 Register Description

All registers cannot be read immediately after writing. When writing to a register, two HCLK periods are required to load it into the register.

### 46.4.1 Register List

- FMAC1 base address: 0x40058000
- FMAC2 base address: 0x40058400
- FMAC3 base address: 0x40058800
- FMAC4 base address: 0x40058C00

Table 46-1 Filter Math Accelerator (FMAC) Register Offset

| Offset        | Register  | Description                            | Width |
|---------------|-----------|----------------------------------------|-------|
| 0x0000        | FMAC_ENR  | Module Enable Register                 | 32    |
| 0x0004        | FMAC_CTR  | Basic Control Register                 | 32    |
| 0x0008        | FMAC_IER  | Interrupt Control Register             | 32    |
| 0x000C        | FMAC_DTR  | Data Input Register                    | 32    |
| 0x0020+0x04*n | FMAC_CORn | Filter Coefficient Register n (n=0-16) | 32    |
| 0x0010        | FMAC_RTR0 | Result Output Register 0               | 32    |
| 0x0014        | FMAC_RTR1 | Result Output Register 1               | 32    |
| 0x0018        | FMAC_STR  | Operation Status Register              | 32    |

## 46.4.2 Module Enable Register (FMAC\_ENR)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |        |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|--------|
| 0x0000 | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0      |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |        |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   | FMACEN |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   | RW     |

| Bit  | Name     | Function      | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                | Access |
|------|----------|---------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:1 | Reserved | Reserved      | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | -      |
| 0    | FMACEN   | Module Enable | <ul style="list-style-type: none"> <li>0b0: FMAC disabled</li> <li>0b1: FMAC enabled</li> </ul> <div>  <b>Note</b><br/>           This signal is used as the data input reset signal of this module. Even if you need to change the filter order FMAC_DTR or filter coefficient FMAC_COR0-16 later, you need to set this bit signal to 0 first and then set this bit to 1.         </div> | RW     |

## 46.4.3 Basic Control Register (FMAC\_CTR)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |       |    |    |    |          |    |   |   |           |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-------|----|----|----|----------|----|---|---|-----------|---|---|---|---|---|---|---|
| 0x0004 | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15    | 14 | 13 | 12 | 11       | 10 | 9 | 8 | 7         | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000010   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |       |    |    |    |          |    |   |   |           |   |   |   |   |   |   |   |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | SHIFT |    |    |    | Reserved |    |   |   | STAGE_NUM |   |   |   |   |   |   |   |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RW    |    |    |    |          |    |   |   | RW        |   |   |   |   |   |   |   |

| Bit   | Name      | Function                | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | Access |
|-------|-----------|-------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:13 | Reserved  | Reserved                | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | -      |
| 12:8  | SHIFT     | Filtering result shift  | <p>The number of right shifts of the filtering result controls the output of MAU_RTR1:</p> <ul style="list-style-type: none"> <li>0b00000: No shift, that is, select [31:0] of the operation result to MAU_RTR1</li> <li>0b00001: Right shift 1-bit, that is, select [32:1] of the operation result to MAU_RTR1</li> <li>0b00010: Right shift 2-bit, that is, select [33:2] of the operation result to MAU_RTR1</li> <li>.....</li> <li>0b00101: Right shift 5-bit, that is, select [36:5] of the operation result to MAU_RTR1</li> <li>0b00110: Right shift 6-bit, that is, select [36:6] of the operation result to MAU_RTR1, and move the high bit into the flag bit</li> <li>0b00111: Right shift 7-bit, that is, select [36:7] of the operation result to MAU_RTR1, and move the high bit into the flag bit</li> <li>.....</li> <li>0b10100: Right shift 20-bit, that is, select [36:20] of the operation result to MAU_RTR1, and move the high bit into the flag bit</li> <li>0b10101: Right shift 21-bit, that is, select [36:21] of the operation result to MAU_RTR1, and move the high bit into the flag bit</li> <li>Others: No displacement</li> </ul> | RW     |
| 7:5   | Reserved  | Reserved                | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | -      |
| 4:0   | STAGE_NUM | Filtering order setting | Set the filtering order                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | RW     |

#### 46.4.4 Interrupt Control Register (FMAC\_IER)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |       |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|-------|
| 0x0008 | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0     |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |       |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   | INTEN |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   | RW    |

| Bit  | Name     | Function         | Description                                                                                                                                                                                        | Access |
|------|----------|------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:1 | Reserved | Reserved         | -                                                                                                                                                                                                  | -      |
| 0    | INTEN    | Interrupt enable | <ul style="list-style-type: none"> <li>0b0: No interrupt will be generated after the operation is completed</li> <li>0b1: Interrupt occurs after the square root operation is completed</li> </ul> | RW     |

#### 46.4.5 Data Input Register (FMAC\_DTR)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |     |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0x000C | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15  | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |     |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | DIN |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RW  |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit   | Name     | Function   | Description     | Access |
|-------|----------|------------|-----------------|--------|
| 31:16 | Reserved | Reserved   | -               | -      |
| 15:0  | DIN      | Data input | FMAC data input | RW     |

#### 46.4.6 Filter Coefficient Register n (FMAC\_CORn) (n=0-16)

Offset: 0x0020+0x04\*n

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |     |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| -      | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15  | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |     |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | CIN |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RW  |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit   | Name     | Function           | Description              | Access |
|-------|----------|--------------------|--------------------------|--------|
| 31:16 | Reserved | Reserved           | -                        | -      |
| 15:0  | CIN      | Filter coefficient | Filter coefficient input | RW     |

#### 46.4.7 Result Output Register 0 (FMAC\_RTR0)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0x0010 | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | DOUT0        |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access | RO           |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit  | Name  | Function      | Description                                       | Access |
|------|-------|---------------|---------------------------------------------------|--------|
| 31:0 | DOUT0 | Result Output | High 32 bits of the accumulator result are output | RO     |

## 46.4.8 Result Output Register 1 (FMAC\_RTR1)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |  |  |  |  |  |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|--|--|--|--|--|--|
| 0x0014 | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |  |  |  |  |  |  |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |  |  |  |  |  |
| Name   | DOUT1        |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |  |  |  |  |  |
| Access | RO           |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |  |  |  |  |  |

| Bit  | Name  | Function      | Description                                      | Access |
|------|-------|---------------|--------------------------------------------------|--------|
| 31:0 | DOUT1 | Result Output | Low 32 bits of the accumulator result are output | RO     |

## 46.4.9 Operation Status Register (FMAC\_STR)

| Offset | Bit Position |          |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |  |  |  |  |  |
|--------|--------------|----------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|--|--|--|--|--|--|
| 0x0018 | 31           | 30       | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |  |  |  |  |  |  |
| Reset  | 0x00000000   |          |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |  |  |  |  |  |
| Name   | READY        | Reserved |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |  |  |  |  |  |
| Access | RC           |          |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |  |  |  |  |  |

| Bit  | Name     | Function                | Description                                                                                                                                                  | Access |
|------|----------|-------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31   | READY    | Operation completed bit | <ul style="list-style-type: none"> <li>0b0: Calculation not completed, result not available</li> <li>0b1: Calculation completed, result available</li> </ul> | RC     |
| 30:0 | Reserved | Reserved                | -                                                                                                                                                            | -      |



## 47 Error Management Unit (ERMU)

### 47.1 Introduction

Error Management Unit (ERMU) collects error notifications from the safety mechanisms of this product. Each error notification has three response behavior configurations: system interrupt, system reset, and output to the error output port to notify the system for further error handling operations.

### 47.2 Features

- Each error notification has three response behavior configurations: system interrupt, system reset, and output to the error output port
- Error output can be configured from the following three types
  - ▶ Single pin output
  - ▶ Single pin flip output
  - ▶ Dual pins complementary output
- Up to 4 groups of error output are supported
- Support pseudo-error source triggering for diagnosis and debugging
- Support response timeout function for error interrupt handling

## 47.3 Functional Description

### 47.3.1 Function Block Diagram

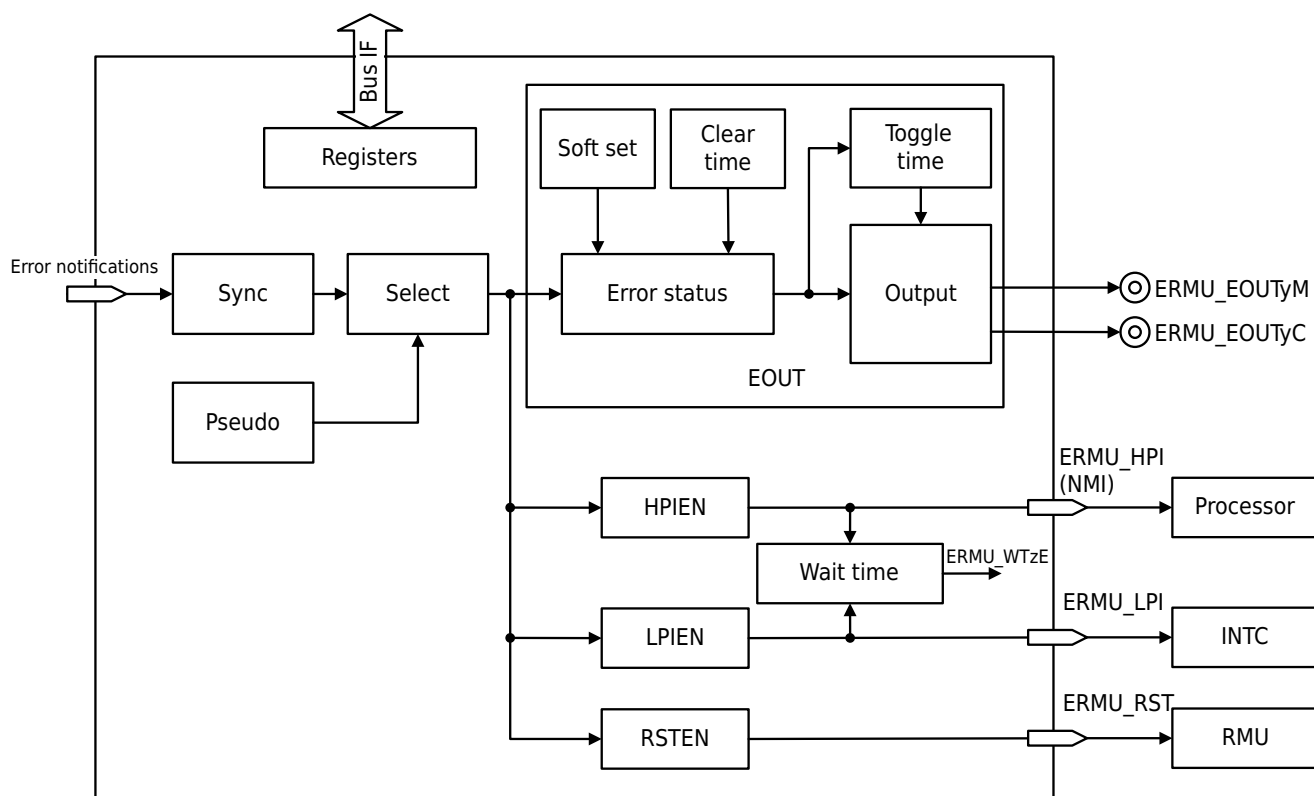


Figure 47-1 ERMU Block Diagram

Table 47-1 ERMU Parameter Definition Table

| Parameter Name | Value | Description                                                       |
|----------------|-------|-------------------------------------------------------------------|
| y              | 0-3   | Error output channel groups are identified by the number 'y'      |
| z              | 0-3   | Single wait timer is identified by the number 'z'                 |
| j              | 0-1   | Error source groups are identified by the number 'j'              |
| k              | 0-31  | Single error source in each group is identified by the number 'k' |
| i              | 0-63  | Single error source ID is identified by the number 'i'            |



#### Note

The content of this table applies to all sections of ERMU.

Table 47-2 ERMU Input/Output Ports

| Port Name                   | Port Type | Description                                                                                 |
|-----------------------------|-----------|---------------------------------------------------------------------------------------------|
| ERMU_EOU-TyM <sup>(1)</sup> | Output    | Error output forward port                                                                   |
| ERMU_EOUTyC                 | Output    | Error output reverse port, the output is reversed compared to the error output forward port |

**Note**

1. Among them, ERMU\_EOUT0M can be configured by ICG8 as the ERMU\_EOUT0M default output pin.
2. Except for 1, other error output ports can be selected for the corresponding functions through the port configuration register.

## 47.3.2 Error Output Channel

### 47.3.2.1 Error Output Operation

After reset release, the default ERMU\_EOyS.EOS is 1. Before using ERMU, ERMU\_EOyC.CLR needs to be written to clear ERMU\_EOyS.EOS. For the output level behavior of the output ports ERMU\_EOUTyM and ERMU\_EOUTyC during reset and after reset, refer to [Error Output Description](#).

In addition to the output ERMU\_EOUT0M of ERMU, which can be configured as the default output pin by ICG8, the other outputs ERMU\_EOUTyM and ERMU\_EOUTyC of ERMU need to be set to the corresponding port output functions to ERMU\_EOUTyM and ERMU\_EOUTyC before using ERMU. For specific configurations, see the "General-Purpose IO (GPIO) " chapter.

When the ERMU\_EOyS.EOS status changes, the levels of the outputs ERMU\_EOUTyM and ERMU\_EOUTyC of ERMU change accordingly. For details, see the following table.

Table 47-3 Error Output Operation

| Error Source Status ER-MU_ESSj.ESSk | Toggle Timer Enable | ERMU_EOUTyM Output Level <sup>(1)</sup> | ERMU_EOUTyC Output Level <sup>(1)</sup> | Error Output Status ER-MU_EOyS.EOS |
|-------------------------------------|---------------------|-----------------------------------------|-----------------------------------------|------------------------------------|
| 0                                   | 0(Disable)          | High                                    | Low                                     | 0 (No Error)                       |
|                                     | 1(Enable)           | Flip                                    | Flip                                    | 0 (No Error)                       |
| 1                                   | 0(Disable)          | Low                                     | High                                    | 1(Error Detected)                  |
|                                     | 1(Enable)           | Low                                     | High <sup>(2)</sup>                     | 1(Error Detected)                  |

**Note**

1. The output from the error source to the error port can be masked by ERMU\_EOyOMj. ERMU\_EOUT0M can be configured as the default output pin by ICG8. For the other output ports ERMU\_EOUTyM and ERMU\_EOUTyC, the corresponding port configuration register needs to be set by software to select the corresponding error output function of ERMU to act as the error output port.
2. Assuming that when the flip timer is enabled, ERMU\_EOUTyC is configured as the port output function.

### 47.3.2.2 Error Output Description

For the behavior of ERMU\_EOUTyM and ERMU\_EOUTyC during reset and after reset, see the following table.

Table 47-4 Description of ERMU\_EOUT0M Output Situation

| Categories                | Reset Category            |              |                            |                            |
|---------------------------|---------------------------|--------------|----------------------------|----------------------------|
|                           | Power-on/Power-down Reset | System Reset | Module Reset               | JTAG Reset                 |
| Output level during reset | Hi-Z                      | Hi-Z         | Retain output before reset | Retain output before reset |

| Categories               | Reset Category            |              |                            |                            |
|--------------------------|---------------------------|--------------|----------------------------|----------------------------|
|                          | Power-on/Power-down Reset | System Reset | Module Reset               | JTAG Reset                 |
| Output level after reset | Low level                 | Low level    | Retain output before reset | Retain output before reset |

Table 47-5 ERMU\_EOUTyM (y=1-3) and ERMU\_EOUTyC (y=0-3) Output Port Description

| Categories                | Reset Category            |                           |                            |                            |
|---------------------------|---------------------------|---------------------------|----------------------------|----------------------------|
|                           | Power-on/Power-down Reset | System Reset              | Module Reset               | JTAG Reset                 |
| Output level during reset | Hi-Z                      | Hi-Z                      | Retain output before reset | Retain output before reset |
| Output level after reset  | General IO <sup>(1)</sup> | General IO <sup>(1)</sup> | Retain output before reset | Retain output before reset |

**Note**

1. When the reset of the output ports ERMU\_EOUTyM (y=1-3) and ERMU\_EOUTyC (y=0-3) is released and it become general IO function. The corresponding port configuration register needs to be set by software to select the corresponding error output function of ERMU to act as the error output port.

### 47.3.2.3 Error Output Status is Controlled by Software

#### Set Error Output Status

Follow the steps below to set the error output status through software:

1. Set ERMU\_EOyC.SET to 1
2. Wait until ERMU\_EOyS.EOS becomes 1

#### Clear Error Output Status

Follow the steps below to clear the error output status:

1. Confirm that all ERMU\_ESSj.ESSk not masked by ERMU\_EOyOMj are 0
2. Wait for ERMU\_EOyS.CTS to become 0
3. Set ERMU\_EOyC.CLR to 1
4. Wait until ERMU\_EOyS.EOS becomes 0

### 47.3.2.4 Flip Timer

The flip timer count clock is the clock after the MRC (8MHz) is prescaled. If ERMU\_EOyTTC.TTE=1, the flip timer starts counting from 0, and when the counter value matches ERMU\_EOyTTCMP, it starts counting from 0 again. When the counter value matches, the output signals ERMU\_EOUTyM and ERMU\_EOUTyC are flipped.

### 47.3.2.5 Enable Flip Timer

Follow the steps below to enable the flip timer:

1. Set ERMU\_EOyTTCMP to define the flip period
2. Set ERMU\_EOyC.CLR to 1 to clear the error output status
3. Wait until ERMU\_EOyS.EOS becomes 0
4. Set ERMU\_EOyTTC.TTE to 1

#### 47.3.2.6 Disable Flip Timer

Follow the steps below to disable the flip timer:

1. Set `ERMU_EOyC.SET` to 1 to set the error output status
2. Wait until `ERMU_EOyS.EOS` becomes 1
3. Set `ERMU_EOyTTC.TTE` to 0

#### 47.3.2.7 Clear Timer

Use the clear timer to keep the error status output by the ERMU output port active for a certain time period, which can be configured through `ERMU_EOyCTCMP`.

The clear timer count clock is the clock after the MRC (8MHz) is prescaled. Set `ERMU_EOyCTCMP` to a non-zero value and `ERMU_EOyC.CTE=1` to enable the clear timer. When an active error source occurs, the counter starts counting. Note that even during the clear timer count process, when a new active error source appears, the counter starts counting from 0. When the counter value matches `ERMU_EOyCTCMP`, the clear timer stops counting. When `ERMU_EOyS.CTS=1`, writing 1 to `ERMU_EOyC.CLR` has no effect.

#### 47.3.3 Wait Timer

Use the wait timer to detect the processing time for the CPU to handle ERMU interrupts (high priority interrupts or low priority interrupts). Once a timeout is detected, further requests can be made for the port to output the error status or request a system reset.

The wait timer count clock is the clock after the MRC (8MHz) is prescaled. Set `ERMU_WTzCMP` to a non-zero value and `ERMU_WTzC.WTE=1` to enable the wait timer. When the start interrupt set by `ERMU_WTzSE` is enabled, the counter starts counting from 0. During the wait timer count process, the generation of new interrupts does not affect the wait timer count. When the counter value matches `ERMU_WTzCMP`, the wait timer stops counting and generates the corresponding wait timer timeout error signal `ERMU_WTzE`.

`ERMU_WTzE` can be configured as error output, interrupt, or reset request.



##### Notice

Avoid triggering the timeout event loop by itself for the same wait timer. This situation can be described as when the timeout event of the current stage wait timer occurs (that is, `ERMU_WTzE` is enable), and this signal is also configured as the start interrupt of the ERMU wait timer, triggering the wait timer to start counting for the next stage, and then the timeout event triggers itself in a loop. It is recommended to set `ERMU_WTzE` to different error notification types. For example, when the trigger source of `ERMU_WTzE` is `ERMU_LPI`, set the `ERMU_WTzE` event to the high priority interrupt `ERMU_HPI`; for another example, when the trigger source of `ERMU_WTzE` is `ERMU_HPI`, set the `ERMU_WTzE` event to the reset `ERMU_RST` or `EOUTyM`.

#### 47.3.4 Reset Request

When the enabled error source in `ERMU_REj` has an error, ERMU sends a reset request to the reset control module RMU, which eventually causes a system reset.

For the reset timing and reset flag, see "Reset Control (RMU) " chapter.

#### 47.3.5 Interrupt Request

When the enabled error source in `ERMU_HPIEj` or `ERMU_LPIEj` has an error, ERMU sends an `ERMU_HPI` request or an `ERMU_LPI` request. As a system non-maskable interrupt (NMI), `ERMU_HPI` directly causes a

system interrupt. ERMU\_LPI needs to set the corresponding enable register in INTC at the same time to cause an active system interrupt. For details, see "Interrupt Controller (INTC) " chapter.

### 47.3.6 Pseudo Error Trigger

The pseudo error source triggers ERMU\_PETj.PETk as a test or debugging methods. The error source error behavior can be injected by software. The processing of pseudo-error source error behavior in the ERMU module is basically the same as the processing when the real error source has an error.

### 47.3.7 Interrupt

ERMU supports interrupt requests from different events. The specific information is as follows:

| Interrupt Event              | Interrupt Flag | Interrupt Enable     | Description                                                       |
|------------------------------|----------------|----------------------|-------------------------------------------------------------------|
| ERMU high priority interrupt | ERMU_HPI       | Register ER-MU_HPIEj | Used as an NMI interrupt, directly output to the processor        |
| ERMU low priority interrupt  | ERMU_LPI       | Register ER-MU_LPIEj | Normal interrupt, output to the processor through the INTC module |

### 47.3.8 Error Source

ERMU supports requests from different error sources. The specific information is as follows:

Table 47-7 ERMU Error Source List

| ID    | Module | Name       | Description                   |
|-------|--------|------------|-------------------------------|
| 0     | ERMU   | ERMU_WT0E  | ERMU wait timer 0 error       |
| 1     | ERMU   | ERMU_WT1E  | ERMU wait timer 1 error       |
| 2     | ERMU   | ERMU_WT2E  | ERMU wait timer 2 error       |
| 3     | ERMU   | ERMU_WT3E  | ERMU wait timer 3 error       |
| 4-7   | -      | -          | Reserved                      |
| 8     | SWDT   | SWDT_ERR   | SWDT timeout overflow error   |
| 9     | WDT    | WDT_ERR    | WDT timeout overflow error    |
| 10    | -      | -          | Reserved                      |
| 11    | EFM    | FLASH_CERR | FLASH ECC correctable error   |
| 12    | EFM    | FLASH_FERR | FLASH ECC uncorrectable error |
| 13    | EFM    | FLASH_OERR | FLASH ECC overflow error      |
| 14    | SRAMH  | SRAMH_CERR | SRAMH ECC correctable error   |
| 15    | SRAM1  | SRAM1_CERR | SRAM1 ECC correctable error   |
| 16    | SRAM2  | SRAM2_CERR | SRAM2 ECC correctable error   |
| 17    | SRAM3  | SRAM3_CERR | SRAM3 ECC correctable error   |
| 18    | SRAM4  | SRAM4_CERR | SRAM4 ECC correctable error   |
| 19    | SRAMB  | SRAMB_CERR | SRAMB ECC correctable error   |
| 20-22 | -      | -          | Reserved                      |

| ID    | Module | Name                   | Description                       |
|-------|--------|------------------------|-----------------------------------|
| 23    | SRAMH  | SRAMH_FERR             | SRAMH ECC uncorrectable error     |
| 24    | SRAM1  | SRAM1_FERR             | SRAM1 ECC uncorrectable error     |
| 25    | SRAM2  | SRAM2_FERR             | SRAM2 ECC uncorrectable error     |
| 26    | SRAM3  | SRAM3_FERR             | SRAM3 ECC uncorrectable error     |
| 27    | SRAM4  | SRAM4_FERR             | SRAM4 ECC uncorrectable error     |
| 28    | SRAMB  | SRAMB_FERR             | SRAMB ECC uncorrectable error     |
| 29-31 | -      | -                      | Reserved                          |
| 32    | CACHE  | CACHE_CERR             | Cache RAM ECC correctable error   |
| 33    | CACHE  | CACHE_FERR             | Cache RAM ECC uncorrectable error |
| 34-35 | -      | -                      | Reserved                          |
| 36    | CAN1   | CREC1_CERR             | CAN1 RAM ECC correctable error    |
| 37    | CAN2   | CREC2_CERR             | CAN2 RAM ECC correctable error    |
| 38    | MCAN   | MCREC_CERR             | MCAN RAM ECC correctable error    |
| 39-41 | -      | -                      | Reserved                          |
| 42    | CAN1   | CREC1_FERR             | CAN1 RAM ECC uncorrectable error  |
| 43    | CAN2   | CREC2_FERR             | CAN2 RAM ECC uncorrectable error  |
| 44    | MCAN   | MCREC_FERR             | MCAN RAM ECC uncorrectable error  |
| 45-47 | -      | -                      | Reserved                          |
| 48    | PKE_LP | PKERAM_CERR            | PKE RAM ECC correctable error     |
| 49    | PKE_LP | PKERAM_FERR            | PKE RAM ECC uncorrectable error   |
| 50    | -      | -                      | Reserved                          |
| 51    | MPU    | MPU_SMPU1ERR           | MPU SMPU 1 bus error              |
| 52    | MPU    | MPU_SMPU2ERR           | MPU SMPU 2 bus error              |
| 53    | MPU    | MPU_FMPUERR            | MPU FMPU bus error                |
| 54    | MPU    | MPU_HMPUERR            | MPU HMPU bus error                |
| 55    | MPU    | MPU_EMPUERR            | MPU EMPU bus error                |
| 56    | -      | -                      | Reserved                          |
| 57    | DMA    | DMA1_ERR               | DMA1 transfer error               |
| 58    | DMA    | DMA2_ERR               | DMA2 transfer error               |
| 59    | -      | -                      | Reserved                          |
| 60    | INTC   | FPU_ERR <sup>(1)</sup> | FPU calculation error             |
| 61    | FCM    | FCM_ERR                | clock monitor error               |
| 62-63 | -      | -                      | Reserved                          |

**Note**

1. The FPU-related error interrupt needs to be enabled first, and the corresponding error can be output to ERMU when it occurs.



## 47.4 Operation Example

### 47.4.1 ERMU Error Management Example

#### Procedure

- Step1.** Read and check each bit in ERMU\_ESSj to confirm whether there is an error;
- Step2.** If there is an error, directly enter the system error handling program. If there is no error, enter the following process;
- Step3.** Write 1 to ERMU\_EOyC.CLR;
- Step4.** Read and check ERMU\_EOyS.EOS until it is 0. If it is not 0 all the time, go to Step1;
- Step5.** Set the port function register as needed to select the output functions of ERMU\_EOUTyM and ERMU\_EOUTyC;
- Step6.** Configure ERMU\_EOyOMj, ERMU\_HPIEj, ERMU\_LPIEj, and ERMU\_REj as required.
- Step7.** Set ERMU\_EOyCTCMP and ERMU\_EOyC.CTE as needed;
- Step8.** Configure ERMU\_EOyTTCMP and ERMU\_EOyTTC.TTE as required.
- Step9.** Configure ERMU\_WTzCMP, ERMU\_WTzSE, and ERMU\_WTzC.WTE as required.
- Step10.** Set the access permission of the ERMU bus, cancel the write permission, and only keep the read permission;
- Step11.** Configure the interrupt handling functions for ERMU\_HPI and ERMU\_LPI when interrupts occur as needed.

## 47.5 Register Description

### 47.5.1 Register List

Base address: 0x40048800

Table 47-8 Error Management Unit (ERMU) Register Offset

| Offset          | Register      | Description                                         | Width |
|-----------------|---------------|-----------------------------------------------------|-------|
| 0x0080*y        | ERMU_EOyC     | Error Output y Control Register (y=0-3)             | 32    |
| 0x0004+0x0080*y | ERMU_EOyS     | Error Output y Status Register (y=0-3)              | 32    |
| 0x0008+0x0080*y | ERMU_EOyCTCNT | Error Output y Clear Timer Count Register (y=0-3)   | 32    |
| 0x000C+0x0080*y | ERMU_EOyCTCMP | Error Output y Clear Timer Compare Register (y=0-3) | 32    |
| 0x0010+0x0080*y | ERMU_EOyTTC   | Error Output y Flip Timer Control Register (y=0-3)  | 32    |
| 0x0018+0x0080*y | ERMU_EOyTTCMP | Error output y Flip Timer Compare register (y=0-3)  | 32    |
| 0x0040+0x0080*y | ERMU_EOyOM0   | Error Output y Output Mask 0 Register (y=0-3)       | 32    |
| 0x0044+0x0080*y | ERMU_EOyOM1   | Error Output y Output Mask 1 Register (y=0-3)       | 32    |
| 0x0200+0x0020*z | ERMU_WTzC     | Wait Timer z Control Register (z=0-3)               | 32    |
| 0x0204+0x0020*z | ERMU_WTzS     | Wait Timer z Status Register (z=0-3)                | 32    |
| 0x0208+0x0020*z | ERMU_WTzCNT   | Wait Timer z Count Register (z=0-3)                 | 32    |
| 0x020C+0x0020*z | ERMU_WTzCMP   | Wait Timer z Compare Register (z=0-3)               | 32    |
| 0x0210+0x0020*z | ERMU_WTzSE    | Wait Timer z Start Enable Register (z=0-3)          | 32    |
| 0x0300+0x0004*j | ERMU_ESSj     | Error Source Status j Register (j=0-1)              | 32    |
| 0x0340+0x0004*j | ERMU_ESSCj    | Error Source Status Clear j Register (j=0-1)        | 32    |
| 0x0380+0x0004*j | ERMU_PETj     | Pseudo-error Source Trigger j Register (j=0-1)      | 32    |
| 0x03C0+0x0004*j | ERMU_REj      | Reset Enable j Register (j=0-1)                     | 32    |
| 0x0400+0x0004*j | ERMU_LPIEj    | Low-Priority Interrupt Enable j Register (j=0-1)    | 32    |
| 0x0500+0x0004*j | ERMU_HPIEj    | High-Priority Interrupt Enable j Register (j=0-1)   | 32    |
| 0x0600          | ERMU_CCPS     | Count Clock Prescaler Register                      | 32    |



#### Note

Write access to the ERMU register only supports word write mode.

## 47.5.2 Error Output y Control Register (ERMU\_EOyC) (y=0-3)

| Offset   | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |     |          |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |     |     |
|----------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-----|----------|----|----|----|----|---|---|---|---|---|---|---|---|---|---|--|-----|-----|
| 0x0080*y | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15  | 14       | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |  |     |     |
| Reset    | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |     |          |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |     |     |
| Name     | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | CTE | Reserved |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  | CLR |     |
| Access   |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RW  |          |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  | WO  | SET |

| Bit   | Name     | Function                  | Description                                                                                                                                                                                                                                                                                                                                                                                                                                 | Access |
|-------|----------|---------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:17 | Reserved | Reserved                  | -                                                                                                                                                                                                                                                                                                                                                                                                                                           | -      |
| 16    | CTE      | Clear timer enable        | <ul style="list-style-type: none"> <li>0b0: Disable clear timer</li> <li>0b1: Enable clear timer</li> </ul>                                                                                                                                                                                                                                                                                                                                 | RW     |
| 15:2  | Reserved | Reserved                  | -                                                                                                                                                                                                                                                                                                                                                                                                                                           | -      |
| 1     | CLR      | Error output status clear | <p>Write 1 to clear the error output status ERMU_EoyS.EOS, write 0 has no effect; read always returns 0</p> <div>  <b>Note</b> <ul style="list-style-type: none"> <li>When ERMU_ESSj.ESSk not masked by ERMU_EOyOMj is not 0, writing 1 to this bit has no effect.</li> <li>When ERMU_EOyS.CTS=1, writing 1 to this bit has no effect.</li> </ul> </div> | WO     |
| 0     | SET      | Set error output status   | Write 1 to set the error output status ERMU_EoyS.EOS, write 0 has no effect; read always returns 0                                                                                                                                                                                                                                                                                                                                          | WO     |

### 47.5.3 Error Output y Status Register (ERMU\_EOyS) (y=0-3)

Offset: 0x0004+0x0080\*y

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |     |    |          |    |    |    |   |   |   |   |   |   |   |   |   |   |     |  |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-----|----|----------|----|----|----|---|---|---|---|---|---|---|---|---|---|-----|--|
| -      | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15  | 14 | 13       | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |     |  |
| Reset  | 0x00000001   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |     |    |          |    |    |    |   |   |   |   |   |   |   |   |   |   |     |  |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | CTS |    | Reserved |    |    |    |   |   |   |   |   |   |   |   |   |   | EOS |  |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RO  |    |          |    |    |    |   |   |   |   |   |   |   |   |   |   | RO  |  |

| Bit   | Name     | Function                | Description                                                                                                         | Access |
|-------|----------|-------------------------|---------------------------------------------------------------------------------------------------------------------|--------|
| 31:17 | Reserved | Reserved                | -                                                                                                                   | -      |
| 16    | CTS      | Clear timer status      | <ul style="list-style-type: none"> <li>0b0: Clear timer count stop</li> <li>0b1: Clear timer is counting</li> </ul> | RO     |
| 15:1  | Reserved | Reserved                | -                                                                                                                   | -      |
| 0     | EOS      | Set error output status | <ul style="list-style-type: none"> <li>0b0: No error output</li> <li>0b1: Error output exists</li> </ul>            | RO     |

### 47.5.4 Error Output y Clear Timer Count Register (ERMU\_EOyCTCNT) (y=0-3)

Offset: 0x0008+0x0080\*y

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |     |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| -      | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15  | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |     |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | CNT |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |     |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit   | Name     | Function                | Description                                                  | Access |
|-------|----------|-------------------------|--------------------------------------------------------------|--------|
| 31:16 | Reserved | Reserved                | -                                                            | -      |
| 15:0  | CNT      | Clear timer count value | When the core stops, the clear timer does not stop counting. | RO     |

### 47.5.5 Error Output y Clear Timer Compare Register (ERMU\_EOyCTCMP) (y=0-3)

Offset: 0x000C+0x0080\*y

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |     |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| -      | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15  | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |     |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | CMP |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RW  |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit   | Name     | Function                  | Description                                                                                                                                                                                          | Access |
|-------|----------|---------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:16 | Reserved | Reserved                  | -                                                                                                                                                                                                    | -      |
| 15:0  | CMP      | Clear timer compare value | This parameter can only be set when the value of ERMU_EOyC.CTE is 0. When the count value of the clear timer reaches the comparison value, the counting stops and automatically clears ERMU_EOyS.CTS | RW     |

### 47.5.6 Error Output y Flip Timer Control Register (ERMU\_EOyTTC) (y=0-3)

Offset: 0x0010+0x0080\*y

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |     |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|-----|
| '      | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0   |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |     |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   | TTE |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   | RW  |

| Bit  | Name     | Function          | Description                                                                                               | Access |
|------|----------|-------------------|-----------------------------------------------------------------------------------------------------------|--------|
| 31:1 | Reserved | Reserved          | -                                                                                                         | -      |
| 0    | TTE      | Flip timer enable | <ul style="list-style-type: none"> <li>0b0: Disable flip timer</li> <li>0b1: Enable flip timer</li> </ul> | RW     |

### 47.5.7 Error Output y Flip Timer Compare Register (ERMU\_EOyTTCMP) (y=0-3)

Offset: 0x0018+0x0080\*y

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |     |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| -      | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15  | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |     |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | CMP |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RW  |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit   | Name     | Function                 | Description                                                                                                                                                                | Access |
|-------|----------|--------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:16 | Reserved | Reserved                 | -                                                                                                                                                                          | -      |
| 15:0  | CMP      | Flip timer compare value | The parameter can only be set when ERMU_EOyC.TTE is 0. When the flip timer count value reaches this comparison value, the counter automatically starts counting from zero. | RW     |

### 47.5.8 Error Output y Mask 0 Register (ERMU\_EOyOM0) (y=0-3)

Offset: 0x0040+0x0080\*y

| Offset | Bit Position |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |     |     |     |     |     |     |     |     |     |     |    |    |    |    |    |    |
|--------|--------------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|----|----|----|----|----|----|
| -      | 31           | 30   | 29   | 28   | 27   | 26   | 25   | 24   | 23   | 22   | 21   | 20   | 19   | 18   | 17   | 16   | 15   | 14   | 13   | 12   | 11   | 10   | 9   | 8   | 7   | 6   | 5   | 4   | 3   | 2   | 1   | 0   |    |    |    |    |    |    |
| Reset  | 0x00000000   |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |     |     |     |     |     |     |     |     |     |     |    |    |    |    |    |    |
| Name   | OM31         | OM30 | OM29 | OM28 | OM27 | OM26 | OM25 | OM24 | OM23 | OM22 | OM21 | OM20 | OM19 | OM18 | OM17 | OM16 | OM15 | OM14 | OM13 | OM12 | OM11 | OM10 | OM9 | OM8 | OM7 | OM6 | OM5 | OM4 | OM3 | OM2 | OM1 | OM0 |    |    |    |    |    |    |
| Access | RW           | RW   | RW   | RW   | RW   | RW   | RW   | RW   | RW   | RW   | RW   | RW   | RW   | RW   | RW   | RW   | RW   | RW   | RW   | RW   | RW   | RW   | RW  | RW  | RW  | RW  | RW  | RW  | RW  | RW  | RW  | RW  | RW | RW | RW | RW | RW | RW |

| Bit  | Name            | Function                | Description                                                                                                                                                                                             | Access |
|------|-----------------|-------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:0 | OMk<br>(k=31-0) | Output mask control bit | Error output y output mask control bit, used to mask error source number i (i=k) <ul style="list-style-type: none"> <li>0b0: Error output is not masked</li> <li>0b1: Error output is masked</li> </ul> | RW     |

### 47.5.9 Error Output y Mask 1 Register (ERMU\_EOyOM1) (y=0-3)

Offset: 0x0044+0x0080\*y

| Offset | Bit Position |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |     |     |     |     |     |     |     |     |     |     |
|--------|--------------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| -      | 31           | 30   | 29   | 28   | 27   | 26   | 25   | 24   | 23   | 22   | 21   | 20   | 19   | 18   | 17   | 16   | 15   | 14   | 13   | 12   | 11   | 10   | 9   | 8   | 7   | 6   | 5   | 4   | 3   | 2   | 1   | 0   |
| Reset  | 0x00000000   |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |     |     |     |     |     |     |     |     |     |     |
| Name   | OM31         | OM30 | OM29 | OM28 | OM27 | OM26 | OM25 | OM24 | OM23 | OM22 | OM21 | OM20 | OM19 | OM18 | OM17 | OM16 | OM15 | OM14 | OM13 | OM12 | OM11 | OM10 | OM9 | OM8 | OM7 | OM6 | OM5 | OM4 | OM3 | OM2 | OM1 | OM0 |
| Access | RW           | RW   | RW   | RW   | RW   | RW   | RW   | RW   | RW   | RW   | RW   | RW   | RW   | RW   | RW   | RW   | RW   | RW   | RW   | RW   | RW   | RW   | RW  | RW  | RW  | RW  | RW  | RW  | RW  | RW  | RW  | RW  |

| Bit  | Name            | Function                | Description                                                                                                                                                                                                   | Access |
|------|-----------------|-------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:0 | OMk<br>(k=31-0) | Output mask control bit | Error output y output mask control bit, used to mask error source number i (i=32+k)<br><ul style="list-style-type: none"> <li>0b0: Error output is not masked</li> <li>0b1: Error output is masked</li> </ul> | RW     |

### 47.5.10 Wait Timer z Control Register (ERMU\_WTzC) (z= 0-3)

Offset: 0x0200+0x0020\*z

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |          |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |    |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----------|----|----|----|----|---|---|---|---|---|---|---|---|---|---|--|----|
| -      | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14       | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |  |    |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |          |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |    |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | Reserved |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |    |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | WO |          |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  | RW |

| Bit   | Name     | Function                      | Description                                                                                                                      | Access |
|-------|----------|-------------------------------|----------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:15 | Reserved | Reserved                      | -                                                                                                                                | -      |
| 16    | STP      | Wait timer stop count control | Writing 1 stops the wait timer counting and waits for the next active start enable, write 0 has no effect; read always returns 0 | WO     |
| 15:1  | Reserved | Reserved                      | -                                                                                                                                | -      |
| 0     | WTE      | Wait timer enable             | <ul style="list-style-type: none"> <li>0b0: Disable wait timer</li> <li>0b1: Enable wait timer</li> </ul>                        | RW     |

### 47.5.11 Wait Timer z Status Register (ERMU\_WTzS) (z= 0-3)

Offset: 0x0204+0x0020\*z

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |     |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|-----|
| -      | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0   |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |     |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   | WTS |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   | RO  |

| Bit  | Name     | Function                | Description                                                                                                       | Access |
|------|----------|-------------------------|-------------------------------------------------------------------------------------------------------------------|--------|
| 31:1 | Reserved | Reserved                | -                                                                                                                 | -      |
| 0    | WTS      | Wait timer count status | <ul style="list-style-type: none"> <li>0b0: Wait timer count stop</li> <li>0b1: Wait timer is counting</li> </ul> | RO     |

### 47.5.12 Wait Timer z Count Register (ERMU\_WTzCNT) (z= 0-3)

Offset: 0x0208+0x0020\*z

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |     |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| -      | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15  | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |     |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | CNT |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |     |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit   | Name     | Function               | Description                                                 | Access |
|-------|----------|------------------------|-------------------------------------------------------------|--------|
| 31:16 | Reserved | Reserved               | -                                                           | -      |
| 15:0  | CNT      | Wait timer count value | When the core stops, the wait timer does not stop counting. | RO     |



### 47.5.13 Wait Timer z Compare Register (ERMU\_WTzCMP) (z=0-3)

Offset: 0x020C+0x0020\*z

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |     |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| -      | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15  | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |     |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | CMP |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RW  |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit   | Name     | Function                 | Description                                                                                                                                                                      | Access |
|-------|----------|--------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:16 | Reserved | Reserved                 | -                                                                                                                                                                                | -      |
| 15:0  | CMP      | Wait timer compare value | The parameter can only be set when ERMU_WTzC.WTE is 0. When the wait timer count value reaches this comparison value, counting stops and ERMU_WTzS.WTS is automatically cleared. | RW     |

### 47.5.14 Wait Timer z Start Enable Register (ERMU\_WTzSE) (z=0-3)

Offset: 0x0210+0x0020\*z

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |       |          |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |       |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-------|----------|----|----|----|----|---|---|---|---|---|---|---|---|---|---|--|-------|
| -      | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15    | 14       | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |  |       |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |       |          |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |       |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | HPISE | Reserved |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  | LPISE |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RW    |          |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  | RW    |

| Bit   | Name     | Function               | Description                                                                                                                                                                                                   | Access |
|-------|----------|------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:15 | Reserved | Reserved               | -                                                                                                                                                                                                             | -      |
| 16    | HPISE    | HPI start timer enable | <ul style="list-style-type: none"> <li>0b0: HPI start timer enable is off</li> <li>0b1: HPI start timer enable is on, and the start wait timer starts counting when ERMU triggers an HPI interrupt</li> </ul> | RW     |
| 15:1  | Reserved | Reserved               | -                                                                                                                                                                                                             | -      |
| 0     | LPISE    | LPI start timer enable | <ul style="list-style-type: none"> <li>0b0: LPI start timer enable is off</li> <li>0b1: LPI start timer enable is on, and the start wait timer starts counting when ERMU triggers an LPI interrupt</li> </ul> | RW     |

### 47.5.15 Error Source Status j Register (ERMU\_ESSj) (j=0-1)

Offset: 0x0300+0x0004\*j

| Offset | Bit Position |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |      |      |      |      |      |      |      |      |      |      |
|--------|--------------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|------|------|------|------|------|------|------|------|------|------|
| -      | 31           | 30    | 29    | 28    | 27    | 26    | 25    | 24    | 23    | 22    | 21    | 20    | 19    | 18    | 17    | 16    | 15    | 14    | 13    | 12    | 11    | 10    | 9    | 8    | 7    | 6    | 5    | 4    | 3    | 2    | 1    | 0    |
| Reset  | 0x00000000   |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |      |      |      |      |      |      |      |      |      |      |
| Name   | ESS31        | ESS30 | ESS29 | ESS28 | ESS27 | ESS26 | ESS25 | ESS24 | ESS23 | ESS22 | ESS21 | ESS20 | ESS19 | ESS18 | ESS17 | ESS16 | ESS15 | ESS14 | ESS13 | ESS12 | ESS11 | ESS10 | ESS9 | ESS8 | ESS7 | ESS6 | ESS5 | ESS4 | ESS3 | ESS2 | ESS1 | ESS0 |
| Access | RO           | RO    | RO    | RO    | RO    | RO    | RO    | RO    | RO    | RO    | RO    | RO    | RO    | RO    | RO    | RO    | RO    | RO    | RO    | RO    | RO    | RO    | RO   | RO   | RO   | RO   | RO   | RO   | RO   | RO   | RO   | RO   |

| Bit  | Name             | Function                | Description                                                                                                                                                                                                                                                                                  | Access |
|------|------------------|-------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:0 | ESSk<br>(k=31-0) | Error source status bit | Error source status bit i (i=j * 32 + k) <ul style="list-style-type: none"> <li>0b0: Error source has no error</li> <li>0b1: Error source has an error</li> </ul> For the correspondence between error source status bit i and error source ID, see <a href="#">ERMU Error Source List</a> . | RO     |



#### Notice

When SWDT is configured for overflow reset, after the system is reset by SWDT, ERMU\_ESS0 corresponding to SWDT\_ERR is not set.

### 47.5.16 Error Source Status Clear j Register (ERMU\_ESSCj) (j=0-1)

Offset: 0x0340+0x0004\*j

| Offset | Bit Position |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |       |       |       |       |       |       |       |       |       |       |    |
|--------|--------------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|----|
| -      | 31           | 30     | 29     | 28     | 27     | 26     | 25     | 24     | 23     | 22     | 21     | 20     | 19     | 18     | 17     | 16     | 15     | 14     | 13     | 12     | 11     | 10     | 9     | 8     | 7     | 6     | 5     | 4     | 3     | 2     | 1     | 0     |    |
| Reset  | 0x00000000   |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |       |       |       |       |       |       |       |       |       |       |    |
| Name   | ESSC31       | ESSC30 | ESSC29 | ESSC28 | ESSC27 | ESSC26 | ESSC25 | ESSC24 | ESSC23 | ESSC22 | ESSC21 | ESSC20 | ESSC19 | ESSC18 | ESSC17 | ESSC16 | ESSC15 | ESSC14 | ESSC13 | ESSC12 | ESSC11 | ESSC10 | ESSC9 | ESSC8 | ESSC7 | ESSC6 | ESSC5 | ESSC4 | ESSC3 | ESSC2 | ESSC1 | ESSC0 |    |
| Access | WO           | WO     | WO     | WO     | WO     | WO     | WO     | WO     | WO     | WO     | WO     | WO     | WO     | WO     | WO     | WO     | WO     | WO     | WO     | WO     | WO     | WO     | WO    | WO    | WO    | WO    | WO    | WO    | WO    | WO    | WO    | WO    | WO |

| Bit  | Name              | Function                      | Description                                                                                                                                                                                                                                                                                                                            | Access |
|------|-------------------|-------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:0 | ESSCk<br>(k=31-0) | Error source status clear bit | Error source status clear bit i (i=j * 32 + k) <p>Writing 1 clears the corresponding error source status bit ERMU_ESSj.ESSk, write 0 has no effect; read always returns 0</p> For the correspondence between the error source status clear bit i and the error source ID, please refer to the <a href="#">ERMU Error Source List</a> . | WO     |

### 47.5.17 Pseudo-error Source Trigger Register (ERMU\_PETj) (j=0-1)

Offset: 0x0380+0x0004\*j

| Offset | Bit Position |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |      |      |      |      |      |      |      |      |      |      |    |
|--------|--------------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|------|------|------|------|------|------|------|------|------|------|----|
| -      | 31           | 30    | 29    | 28    | 27    | 26    | 25    | 24    | 23    | 22    | 21    | 20    | 19    | 18    | 17    | 16    | 15    | 14    | 13    | 12    | 11    | 10    | 9    | 8    | 7    | 6    | 5    | 4    | 3    | 2    | 1    | 0    |    |
| Reset  | 0x00000000   |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |      |      |      |      |      |      |      |      |      |      |    |
| Name   | PET31        | PET30 | PET29 | PET28 | PET27 | PET26 | PET25 | PET24 | PET23 | PET22 | PET21 | PET20 | PET19 | PET18 | PET17 | PET16 | PET15 | PET14 | PET13 | PET12 | PET11 | PET10 | PET9 | PET8 | PET7 | PET6 | PET5 | PET4 | PET3 | PET2 | PET1 | PET0 |    |
| Access | WO           | WO    | WO    | WO    | WO    | WO    | WO    | WO    | WO    | WO    | WO    | WO    | WO    | WO    | WO    | WO    | WO    | WO    | WO    | WO    | WO    | WO    | WO   | WO   | WO   | WO   | WO   | WO   | WO   | WO   | WO   | WO   | WO |

| Bit  | Name             | Function                        | Description                                                                                                                                                                                                                                                                                                                                       | Access |
|------|------------------|---------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:0 | PETk<br>(k=31-0) | Pseudo-error source trigger bit | Pseudo-error source trigger bit i (i=j * 32 + k)<br>Writing 1 to a pseudo-error source triggers the corresponding ERMU_ESSj.ESSk to 1, write 0 has no effect; read always returns 0<br>For the correspondence between the pseudo-error source trigger bit i and the error source ID, please refer to the <a href="#">ERMU Error Source List</a> . | WO     |

### 47.5.18 Reset Enable j Register (ERMU\_REj) (j=0-1)

Offset: 0x03C0+0x0004\*j

| Offset | Bit Position |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |     |     |     |     |     |     |     |     |     |     |    |
|--------|--------------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|----|
| -      | 31           | 30   | 29   | 28   | 27   | 26   | 25   | 24   | 23   | 22   | 21   | 20   | 19   | 18   | 17   | 16   | 15   | 14   | 13   | 12   | 11   | 10   | 9   | 8   | 7   | 6   | 5   | 4   | 3   | 2   | 1   | 0   |    |
| Reset  | 0x00000000   |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |     |     |     |     |     |     |     |     |     |     |    |
| Name   | RE31         | RE30 | RE29 | RE28 | RE27 | RE26 | RE25 | RE24 | RE23 | RE22 | RE21 | RE20 | RE19 | RE18 | RE17 | RE16 | RE15 | RE14 | RE13 | RE12 | RE11 | RE10 | RE9 | RE8 | RE7 | RE6 | RE5 | RE4 | RE3 | RE2 | RE1 | RE0 |    |
| Access | RW           | RW   | RW   | RW   | RW   | RW   | RW   | RW   | RW   | RW   | RW   | RW   | RW   | RW   | RW   | RW   | RW   | RW   | RW   | RW   | RW   | RW   | RW  | RW  | RW  | RW  | RW  | RW  | RW  | RW  | RW  | RW  | RW |

| Bit  | Name            | Function         | Description                                                                                                                                                                                                                                                                                                                           | Access |
|------|-----------------|------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:0 | REk<br>(k=31-0) | Reset enable bit | Reset enable bit i (i=j * 32 + k)<br><ul style="list-style-type: none"> <li>0b0: Reset function disable</li> <li>0b1: Enable reset function, the active bit of ERMU_ESSj.ESSk generates a system reset</li> </ul> For the correspondence between reset enable bit i and error source ID, see <a href="#">ERMU Error Source List</a> . | RW     |

### 47.5.19 Low Priority Interrupt Enable j Register (ERMU\_LPIEj) (j=0-1)

Offset: 0x0400+0x0004\*j

| Offset | Bit Position |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |       |       |       |       |       |       |       |       |       |       |    |
|--------|--------------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|----|
| -      | 31           | 30     | 29     | 28     | 27     | 26     | 25     | 24     | 23     | 22     | 21     | 20     | 19     | 18     | 17     | 16     | 15     | 14     | 13     | 12     | 11     | 10     | 9     | 8     | 7     | 6     | 5     | 4     | 3     | 2     | 1     | 0     |    |
| Reset  | 0x00000000   |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |       |       |       |       |       |       |       |       |       |       |    |
| Name   | LPIE31       | LPIE30 | LPIE29 | LPIE28 | LPIE27 | LPIE26 | LPIE25 | LPIE24 | LPIE23 | LPIE22 | LPIE21 | LPIE20 | LPIE19 | LPIE18 | LPIE17 | LPIE16 | LPIE15 | LPIE14 | LPIE13 | LPIE12 | LPIE11 | LPIE10 | LPIE9 | LPIE8 | LPIE7 | LPIE6 | LPIE5 | LPIE4 | LPIE3 | LPIE2 | LPIE1 | LPIE0 |    |
| Access | RW           | RW     | RW     | RW     | RW     | RW     | RW     | RW     | RW     | RW     | RW     | RW     | RW     | RW     | RW     | RW     | RW     | RW     | RW     | RW     | RW     | RW     | RW    | RW    | RW    | RW    | RW    | RW    | RW    | RW    | RW    | RW    | RW |

| Bit  | Name              | Function                | Description                                                                                                                                                                                                                                                                                                                                                                                                                   | Access |
|------|-------------------|-------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:0 | LPIEk<br>(k=31-0) | Low priority enable bit | <p>Low priority interrupt enable bit i (i=j * 32 + k)</p> <ul style="list-style-type: none"> <li>0b0: Low-priority interrupt function disable</li> <li>0b1: Enable low priority interrupt function, the active bit of ERMU_ESSj.ESSk generates a low priority interrupt</li> </ul> <p>For the correspondence between low priority interrupt enable bit i and error source ID, see <a href="#">ERMU Error Source List</a>.</p> | RW     |

### 47.5.20 High Priority Interrupt Enable j Register (ERMU\_HPIEj) (j=0-1)

Offset: 0x0500+0x0004\*j

| Offset | Bit Position |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |       |       |       |       |       |       |       |       |       |       |    |
|--------|--------------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|----|
| -      | 31           | 30     | 29     | 28     | 27     | 26     | 25     | 24     | 23     | 22     | 21     | 20     | 19     | 18     | 17     | 16     | 15     | 14     | 13     | 12     | 11     | 10     | 9     | 8     | 7     | 6     | 5     | 4     | 3     | 2     | 1     | 0     |    |
| Reset  | 0x00000000   |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |       |       |       |       |       |       |       |       |       |       |    |
| Name   | HPIE31       | HPIE30 | HPIE29 | HPIE28 | HPIE27 | HPIE26 | HPIE25 | HPIE24 | HPIE23 | HPIE22 | HPIE21 | HPIE20 | HPIE19 | HPIE18 | HPIE17 | HPIE16 | HPIE15 | HPIE14 | HPIE13 | HPIE12 | HPIE11 | HPIE10 | HPIE9 | HPIE8 | HPIE7 | HPIE6 | HPIE5 | HPIE4 | HPIE3 | HPIE2 | HPIE1 | HPIE0 |    |
| Access | RW           | RW     | RW     | RW     | RW     | RW     | RW     | RW     | RW     | RW     | RW     | RW     | RW     | RW     | RW     | RW     | RW     | RW     | RW     | RW     | RW     | RW     | RW    | RW    | RW    | RW    | RW    | RW    | RW    | RW    | RW    | RW    | RW |

| Bit  | Name              | Function                 | Description                                                                                                                                                                                                                                                                                                                                                                                                                        | Access |
|------|-------------------|--------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:0 | HPIEk<br>(k=31-0) | High priority enable bit | <p>High priority interrupt enable bit i (i=j * 32 + k)</p> <ul style="list-style-type: none"> <li>0b0: High priority interrupt function disable</li> <li>0b1: Enable high priority interrupt function, the active bit of ERMU_ESSj.ESSk generates a high priority interrupt</li> </ul> <p>For the correspondence between high priority interrupt enable bit i and error source ID, see <a href="#">ERMU Error Source List</a>.</p> | RW     |

## 47.5.21 Count Clock Prescaler Register (ERMU\_CCPS)

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |          |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |     |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|-----|
| 0x0600 | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15       | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |     |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |          |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |     |
| Name   | PSS          |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | Reserved |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   | PSE |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RW       |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |     |

| Bit   | Name     | Function                       | Description                                                                                                                                                                                                                                                                                                                                                                                                | Access |
|-------|----------|--------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:16 | PSS      | Prescaler division coefficient | <p>The count clock prescaler division coefficient. The clock after prescaling is used as the count clock for the clear timer, flip timer, and wait timer. The clock frequency after prescaling is <math>f = f_{MRC} / (PSS + 1)</math>.</p> <div>  <b>Note</b><br/> <math>f_{MRC}</math> the MRC frequency (8MHz). </div> | RW     |
| 15:1  | Reserved | Reserved                       | -                                                                                                                                                                                                                                                                                                                                                                                                          | -      |
| 0     | PSE      | Prescaler enable               | <ul style="list-style-type: none"> <li>0b0: Prescaler function disable</li> <li>0b1: Enable prescaler function</li> </ul>                                                                                                                                                                                                                                                                                  | RW     |

## 48 Debug Controller (DBGC)

### 48.1 Introduction

This MCU's core is Cortex-M4, which includes hardware for advanced debugging functions and supports the Embedded Trace Macrocell (ETM). Using these debugging functions, the core can be stopped when fetching instructions (instruction breakpoint) or accessing data (data breakpoint). When the core is stopped, the internal state of the core and the external state of the system can be queried. After the query completes, the kernel and system are restored and program execution is restored.

**Two debugging interfaces are provided:**

- Serial debug trace interface SWD
- Parallel debug trace interface JTAG

### 48.2 Functional Description

#### 48.2.1 Function Block Diagram

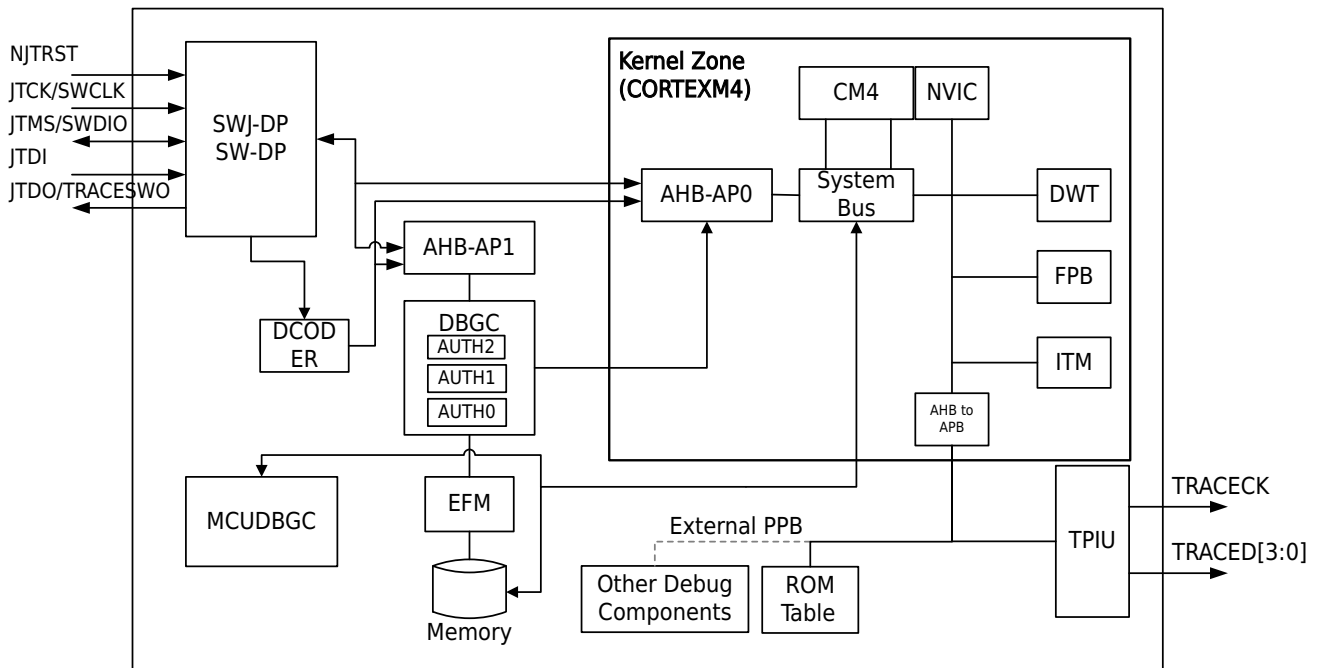


Figure 48-1 Debug Control System Block Diagram

The ARM Cortex-M4 core provides integrated on-chip debugging support. It includes:

- SWJ-DP: SWD/ JTAG debug port
- AHB-AP: AHB access port
- ITM: Instruction tracking unit
- FPB: Flash instruction breakpoint
- DWT: Data breakpoint trigger
- TPIU: Trace port unit interface (available on larger packages where corresponding pins are mapped)
- Flexible debugging pin assignment

**Note**

For more information on the debugging functions supported by the ARM Cortex-M4 core, see the "Cortex-M4F-r0p1 Technical Reference Manual" and "CoreSight Design Kit r0p1 Technical Reference Manual".

## 48.2.2 Reference Documentation

This product references the following ARM technical documentation:

- Cortex M4 Technical Reference Manual r0p1
- Arm Debug Interface Architecture Specification version 5
- CoreSight Components Technical Reference Manual version 1.0

## 48.2.3 SWJ-DP Debug Port (SWD and JTAG)

The MCU core integrates an SWD/JTAG debug port (SWJ-DP). This port is an ARM-standard CoreSight debug port, featuring both a JTAG-DP (5-pin) interface and an SW-DP (2-pin) interface.

- The JTAG debug port (JTAG-DP) provides a 5-pin standard JTAG interface for connecting to the AHB-AP port.
- The Serial Wire Debug port (SW-DP) provides a 2-pin (Clock + Data) interface for connecting to the AHB-AP port.

In SWJ-DP, the 2 JTAG pins of SW-DP are multiplexed with some of the 5 JTAG pins of JTAG-DP.

In the figure below, JTDO reuses TRACESWO and TDO. This means that asynchronous tracing can only be implemented on SW-DP, not on JTAG-DP.

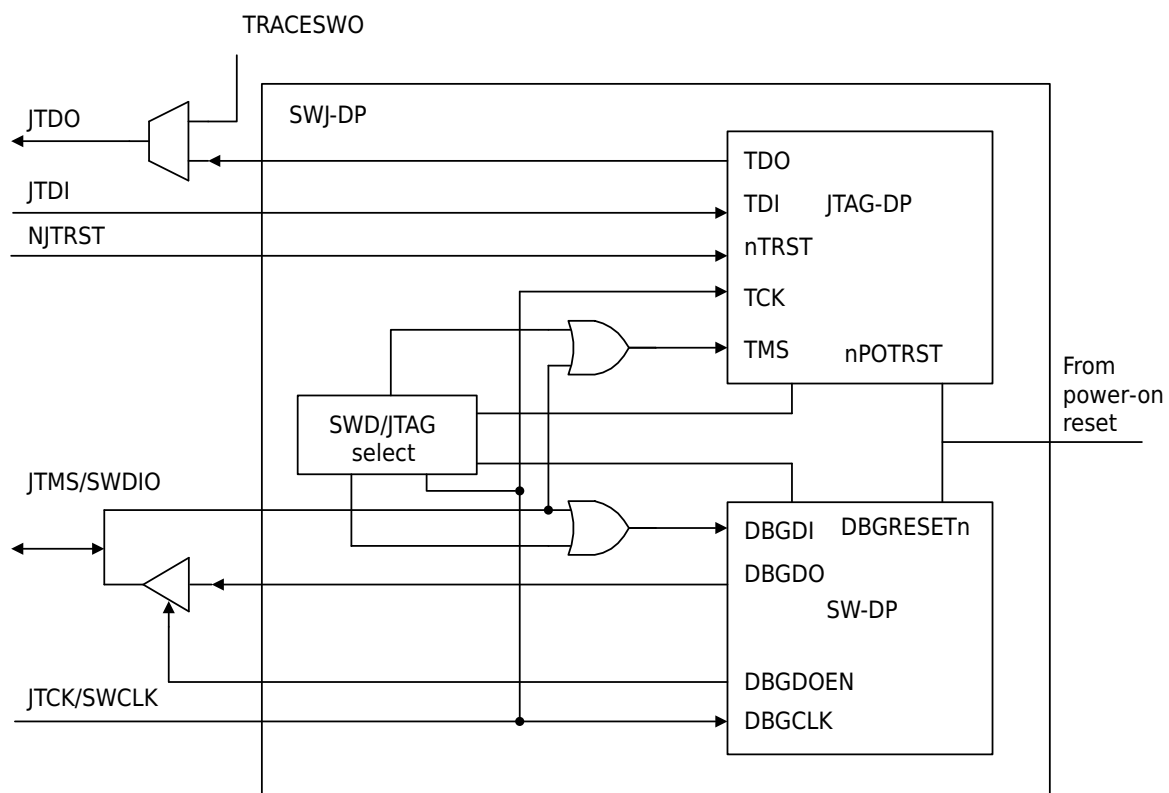


Figure 48-2 Debug Control System

### 48.2.3.1 JTAG-DP or SW-DP Switching Mechanism

The default debug interface is the JTAG-DP interface.

If the debug tool wants to switch to SW-DP, it must provide a dedicated JTAG sequence on JTMS (SWDIO) /JTCK (SWCLK) to disable JTAG-DP and enable SW-DP. This allows access to SW-DP using only the SWCLK and SWDIO pins.

The sequence is:

1. Output a JTMS (SWDIO)=1 signal for more than 50 JTCK periods
2. Output 16 JTMS (SWDIO) signals 0111\_1001\_1110\_0111 (MSB)
3. Output a JTMS (SWDIO)=1 signal for more than 50 JTCK periods

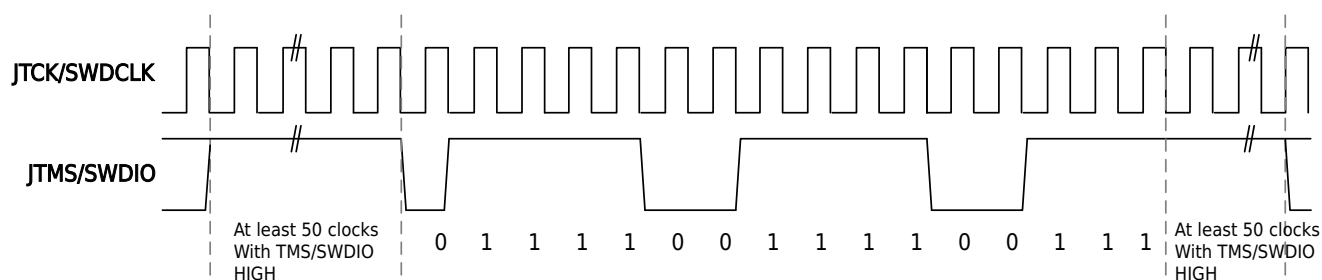


Figure 48-3 JTAG-DP to SW-DP Switching Timing

#### 48.2.4 Pin Arrangement and Debug Port Pins

Depending on the MCU package, there are different active pin counts. Therefore, some pin-related functions may vary from package to package.

##### SWJ Debug Port Pin

5 common I/O ports of MCU can be used as SWJ-DP interface pins.

Table 48-1 SWJ Debug Port Pin

| SWJ-DP Pin Name | JTAG Debug Port |                     | SW Debug Port |                                               |
|-----------------|-----------------|---------------------|---------------|-----------------------------------------------|
|                 | Types           | Description         | Type          | Debug Allocation                              |
| JTMS/SWDIO      | I               | JTAG Mode Selection | I/O           | Serial data input/output                      |
| JTCK/SWCLK      | I               | JTAG Clock          | I             | Serial clock                                  |
| JTDI            | I               | JTAG Data Input     | -             | -                                             |
| JTDO/TRACESWO   | O               | JTAG Data Output    | -             | TRACESWO (if asynchronous tracing is enabled) |
| NJTRST          | I               | JTAG Reset          | -             | -                                             |

##### Flexible SWJ-DP Pin Assignment

After reset (Power On or Pin Reset), all five pins used for SWJ-DP are designated as dedicated pins, immediately available for use by debug tools (Note: to use trace output, configure the register MCUTRA-CECTL). However, the MCU can disable some or all SWJ-DP ports, thereby releasing the relevant pins for use as GPIOs. For more details on how to disable SWJ-DP port pins, refer to the "Special Control Register (PSPCR) " in the "General-Purpose I/O (GPIO) " chapter.



Table 48-2 Flexible SWJ-DP Pin Assignment

| Available Debug Ports                  | Assigned SWJ IO Pins |                |            |            |            |
|----------------------------------------|----------------------|----------------|------------|------------|------------|
|                                        | JTMS/<br>SWDIO       | JTCK/<br>SWCLK | JTDI       | JTDO       | NJTRST     |
| All SWJ (JTAG-DP + SW-DP) -reset state | ✓                    | ✓              | ✓          | ✓          | ✓          |
| Disable JTAG-DP and enable SW-DP       | ✓                    | ✓              | Releasable | Releasable | Releasable |
| Disable JTAG-DP and disable SW-DP      | Releasable           | Releasable     | Releasable | Releasable | Releasable |

### Internal Pull-ups on JTAG Pins

According to the JTAG IEEE standard, it must be ensured that the JTAG input pins are not floating because these pins are directly connected to the MCU internals for controlling the debug functions. Special attention must also be paid to the JTCK/SWCLK pin, which is directly used for debug control clock functions. To avoid floating IO levels, the MCU has built-in internal pull-up resistors on the JTAG pins:

- NJTRST: Internal pull-up
- JTDI: Internal pull-up
- JTMS/SWDIO: Internal pull-up
- JTCK/SWCLK: Internal pull-up
- JTDO: Internal pull-up

In the absence of a connected debugger, user software can release JTAG IO for use as general-purpose I/O ports by setting the GPIO special control register. Since the internal pull-up is a weak pull-up of <100kΩ, it is recommended to use an external pull-up of 10kΩ.

### Using Serial Interfaces and Releasing Unused Debug Pins for GPIO

Some GPIOs can be released when using SWD, user software must change the GPIO configuration in the GPIO control register, so that the corresponding pins can be released as GPIO.

When debugging, the host does the following:

- Under system reset, allocate all SWJ pins (JTAG-DP + SW-DP).
- Under system reset, the debug host sends a JTAG sequence to switch from JTAG-DP to SW-DP.
- Still in the system reset state, the debug host sets a breakpoint at the reset address.
- The reset signal is released and the core stops at the reset address.
- From this debug port switch to SW-DP. Then, other JTAG pins are reallocated as GPIOs through user software.



#### Notice

For user software design, when debug pins need to be released, during the period from reset until user software releases the pins, these pins remain in input pull-up mode (NJTRST, JTMS, JTDI, JTCK, and JTDO).

### 48.2.5 SW Debug Port-SW Protocol Overview

The synchronous serial protocol uses two pins:

- SWCLK: Clock from master to slave
- SWDIO: Bidirectional

LSB is ahead when transferring data.

For SWCLK and SWDIO, it is recommended to use a 10kΩ pull-up on the circuit board.

## 48.2.6 TPIU (Trace Port Interface Unit)

### 48.2.6.1 Overview and Block Diagram

TPIU is a bridge between ITM and on-chip trace data.

Output data streams are packaged with trace source IDs and then captured by the Trace Port Analyzer (TPA).

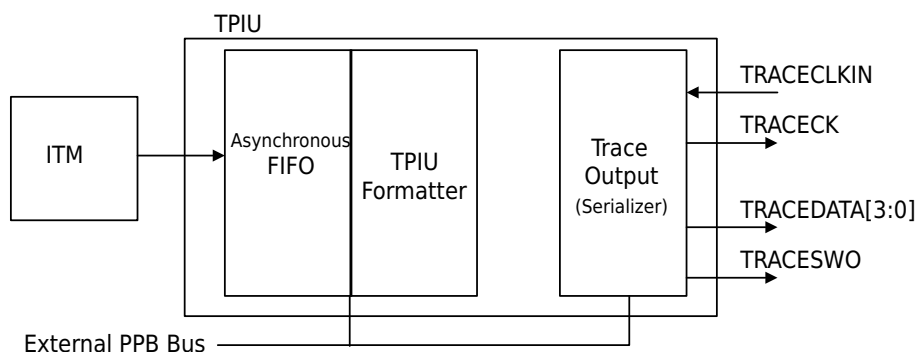


Figure 48-4 TPIU Block Diagram

### 48.2.6.2 TRACE Pin Allocation

- Asynchronous Mode

Asynchronous mode requires 1 extra pin and is available in all packages. Asynchronous mode is only available when using serial line mode (not available in JTAG mode).

| TPIU pin name | Track Asynchronous Mode |                                |
|---------------|-------------------------|--------------------------------|
|               | Type                    | Description                    |
| TRACESWO      | O                       | TRACE asynchronous data output |

- Synchronous Mode

Synchronous mode requires 2 to 5 extra pins, depending on the length of the data being traced, and is only available on larger packages. In addition, synchronous mode is available in both JTAG mode and serial mode, and can provide higher bandwidth output capability than asynchronous trace.

| TPIU Pin Name | Track Sync Mode |                                                       |
|---------------|-----------------|-------------------------------------------------------|
|               | Type            | Description                                           |
| TRACECK       | O               | TRACE Clock                                           |
| TRACED[3:0]   | O               | TRACE synchronous data output can be 1, 2, or 4 bits. |

### TPIU TRACE Pin Allocation Rules

By default, these pins are not allocated. These pins can be configured by setting the TRACE\_IOEN and TRACE\_MODE bits in the MCU Debug Component Configuration Register (MCUTRACECTL). This configuration must be done by the debug host or CPU.

Also, the number of pins to assign depends on the trace configuration (asynchronous trace or synchronous trace).

- Asynchronous mode: 1 extra pin required
- Synchronous mode: 5 extra pins required
  - ▶ TRACECK
  - ▶ TRACED[0] (if port data length is configured as 1, 2, or 4)
  - ▶ TRACED[1] (if port data length is configured as 2 or 4)
  - ▶ TRACED[2] (if port data length is configured as 4)
  - ▶ TRACED[3] (if port data length is configured as 4)

To allocate TRACE pins, the debug host must configure the TRACE\_IOEN and TRACE\_MODE[1:0] bits in the MCU debug configuration register (MCUTRACECTL). By default, TRACE pins are not allocated.

This register is mapped on the external PPB bus and is reset by power-on (not pin reset). This register can be written by the debugger in the pin reset state.

| TPIU Pin Usage             |            |            | Assigned TRACE IO Pins |         |           |           |           |           |
|----------------------------|------------|------------|------------------------|---------|-----------|-----------|-----------|-----------|
| Tracking Mode              | TRACE_IOEN | TRACE_MODE | JTDO/TRACE-SWO         | TRACECK | TRACED[0] | TRACED[1] | TRACED[2] | TRACED[3] |
| No tracing (default state) | 0b0        | 0bXX       | Release <sup>(1)</sup> | Release | Release   | Release   | Release   | Release   |
| Asynchronous tracking      | 0b1        | 0b00       | TRACESWO               | Release | Release   | Release   | Release   | Release   |
| Sync tracking 1 bit        | 0b1        | 0b01       | Release <sup>(1)</sup> | TRACECK | TRACED[0] | Release   | Release   | Release   |
| Sync tracking 2 bits       | 0b1        | 0b10       | Release <sup>(1)</sup> | TRACECK | TRACED[0] | TRACED[1] | Release   | Release   |
| Sync tracking 4 bits       | 0b1        | 0b11       | Release <sup>(1)</sup> | TRACECK | TRACED[0] | TRACED[1] | TRACED[2] | TRACED[3] |



#### Note

1. Released when using serial wire mode. But when using JTAG, this pin is assigned to TDO.

### 48.2.6.3 MCU Internal TRACECLKIN Connection

In this MCU, the clock TRACECLKIN of TPIU is connected to the internal clock. The default clock of the MCU is the internal MRC oscillator. The frequency in reset state is different from the frequency after reset release. The reason is that since the default MRC calibration value is adopted in the system reset state, the MRC calibration value will be updated every time the system reset is released. Therefore, the Trace Port Analyzer (TPA) should not enable tracing (using the TRACE\_IOEN bit) during system reset, as the bit width of the synchronous frame packet under reset state differs from the packet after reset.

### 48.2.6.4 TPIU Register Usage Instructions

Reads and writes to TPIU APB registers are only possible when the TRCENA bit in the Debug Exception and Monitor Control Register (DEMCR) is set to 1. Otherwise, these registers will read as 0 (setting this bit enables the TPIU clock).

### 48.2.6.5 TPIU Configuration Instance

1. Setting TRCENA bit in the Debug Exception and Monitor Control Register (DEMCR) to 1.
2. Writing the desired value to the TPIU Current Port Size Register (CSPSR) (default value is 0x1 for 1-bit port size).
3. Writing 0x102 to the TPIU Formatter and Flush Status Register (default value).
4. Write TPIU select pin protocol to select synchronous mode or asynchronous mode. Example: 0x2 means asynchronous NRZ mode (similar to USART).
5. Writing 0x00 to the MCUTRACECTL.TRACE\_IOEN bit to allocate TRACE I/O for asynchronous mode.

6. Sending a TPIU synchronization packet (FF\_FF\_FF\_7F) at this point.
7. Configuring ITM and writing to the ITM Stimulus Port Register to output values.

## 48.3 Register Description

### 48.3.1 Register List

Base address: 0xE0042000

Table 48-6 Debug Controller (DBGC) Register Offset

| Offset | Register    | Description                              | Width | Access Host                  |
|--------|-------------|------------------------------------------|-------|------------------------------|
| 0x000C | CHIPID      | Chip Identification Register             | 32    | Debug IDE                    |
| 0x001C | MCUDBGSTAT  | DBG Status Register                      | 32    | CPU/Debug IDE <sup>(1)</sup> |
| 0x0020 | MCUSTPCTL   | Peripheral Debug Stop Control Register   | 32    | CPU/Debug IDE <sup>(1)</sup> |
| 0x0024 | MCUTRACECTL | Debug Component Trace Control Register   | 32    | CPU/Debug IDE <sup>(1)</sup> |
| 0x0028 | MCUSTPCTL2  | Peripheral Debug Stop Control Register 2 | 32    | CPU/Debug IDE <sup>(1)</sup> |



#### Note

1. Register bits are located in the PPB region and can only be accessed by the CPU in privileged mode.

### 48.3.2 Chip Identification Register (CHIPID)

Used to indicate company information and product model information.

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0x000C | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reset  | 0x584804A8   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Name   | CHIPID       |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Access | RO           |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

| Bit  | Name   | Function            | Description                                                                     | Access |
|------|--------|---------------------|---------------------------------------------------------------------------------|--------|
| 31:0 | CHIPID | Chip Identification | CHIPID[31:16]: ASCII code for XH, 0x5848<br>CHIPID[15:0]: product model, 0x04A8 | RO     |

### 48.3.3 DBG Status Register (MCUDBGSTAT)

DBG debugs power-on status check register.

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |              |              |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|--------------|--------------|
| 0x001C | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1            | 0            |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |              |              |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   | CDBGPWRUPACK | CDBGPWRUPREQ |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   | RW           | RW           |

| Bit  | Name         | Function                   | Description                                                                                               | Access |
|------|--------------|----------------------------|-----------------------------------------------------------------------------------------------------------|--------|
| 31:2 | Reserved     | Reserved                   | -                                                                                                         | -      |
| 1    | CDBGPWRUPACK | Debugger power-on feedback | <ul style="list-style-type: none"> <li>0b0: No feedback</li> <li>0b1: Debug power-on feedback</li> </ul>  | RW     |
| 0    | CDBGPWRUPREQ | Debugger power-on request  | <ul style="list-style-type: none"> <li>0b0: No power-on request</li> <li>0b1: Power-on request</li> </ul> | RW     |

### 48.3.4 Peripheral Debug Stop Control Register (MCUSTPCTL)

The peripheral module pauses control when the CPU is in the debug state.

| Offset | Bit Position |    |    |    |    |    |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |         |         |         |        |        |        |
|--------|--------------|----|----|----|----|----|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|---------|---------|---------|--------|--------|--------|
| 0x0020 | 31           | 30 | 29 | 28 | 27 | 26 | 25     | 24     | 23     | 22     | 21     | 20     | 19     | 18     | 17     | 16     | 15     | 14     | 13     | 12     | 11     | 10     | 9      | 8      | 7      | 6      | 5       | 4       | 3       | 2      | 1      | 0      |
| Reset  | 0x0000003B   |    |    |    |    |    |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |         |         |         |        |        |        |
| Name   | Reserved     |    |    |    |    |    | M25STP | M24STP | M23STP | M22STP | M21STP | M20STP | M19STP | M18STP | M17STP | M16STP | M15STP | M14STP | M13STP | M12STP | M11STP | M10STP | M09STP | M08STP | M07STP | M06STP | PVD2STP | PVD1STP | PVD0STP | RTCSTP | WDTSTP | SWDSTP |
| Access |              |    |    |    |    |    |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |         |         |         |        |        |        |

| Bit   | Name     | Function                    | Description                                                                                                                                                                    | Access |
|-------|----------|-----------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:26 | Reserved | Reserved                    | -                                                                                                                                                                              | -      |
| 25    | M25STP   | Timer0_5 count pause signal | <ul style="list-style-type: none"> <li>0b0: Counter continues counting even when the core is stopped</li> <li>0b1: Counter pauses counting when the core is stopped</li> </ul> | RW     |
| 24    | M24STP   | Timer0_4 count pause signal | <ul style="list-style-type: none"> <li>0b0: Counter continues counting even when the core is stopped</li> <li>0b1: Counter pauses counting when the core is stopped</li> </ul> | RW     |
| 23    | M23STP   | Timer0_3 count pause signal | <ul style="list-style-type: none"> <li>0b0: Counter continues counting even when the core is stopped</li> <li>0b1: Counter pauses counting when the core is stopped</li> </ul> | RW     |
| 22    | M22STP   | Timer6_8 count pause signal | <ul style="list-style-type: none"> <li>0b0: Counter continues counting even when the core is stopped</li> <li>0b1: Counter pauses counting when the core is stopped</li> </ul> | RW     |
| 21    | M21STP   | Timer6_7 count pause signal | <ul style="list-style-type: none"> <li>0b0: Counter continues counting even when the core is stopped</li> <li>0b1: Counter pauses counting when the core is stopped</li> </ul> | RW     |
| 20    | M20STP   | Timer6_6 count pause signal | <ul style="list-style-type: none"> <li>0b0: Counter continues counting even when the core is stopped</li> <li>0b1: Counter pauses counting when the core is stopped</li> </ul> | RW     |
| 19    | M19STP   | Timer6_5 count pause signal | <ul style="list-style-type: none"> <li>0b0: Counter continues counting even when the core is stopped</li> <li>0b1: Counter pauses counting when the core is stopped</li> </ul> | RW     |

| Bit | Name   | Function                    | Description                                                                                                                                                                    | Access |
|-----|--------|-----------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 18  | M18STP | Timer6_4 count pause signal | <ul style="list-style-type: none"> <li>0b0: Counter continues counting even when the core is stopped</li> <li>0b1: Counter pauses counting when the core is stopped</li> </ul> | RW     |
| 17  | M17STP | Timer6_3 count pause signal | <ul style="list-style-type: none"> <li>0b0: Counter continues counting even when the core is stopped</li> <li>0b1: Counter pauses counting when the core is stopped</li> </ul> | RW     |
| 16  | M16STP | Timer6_2 count pause signal | <ul style="list-style-type: none"> <li>0b0: Counter continues counting even when the core is stopped</li> <li>0b1: Counter pauses counting when the core is stopped</li> </ul> | RW     |
| 15  | M15STP | Timer6_1 count pause signal | <ul style="list-style-type: none"> <li>0b0: Counter continues counting even when the core is stopped</li> <li>0b1: Counter pauses counting when the core is stopped</li> </ul> | RW     |
| 14  | M14STP | Timer4_3 count pause signal | <ul style="list-style-type: none"> <li>0b0: Counter continues counting even when the core is stopped</li> <li>0b1: Counter pauses counting when the core is stopped</li> </ul> | RW     |
| 13  | M13STP | Timer4_2 count pause signal | <ul style="list-style-type: none"> <li>0b0: Counter continues counting even when the core is stopped</li> <li>0b1: Counter pauses counting when the core is stopped</li> </ul> | RW     |
| 12  | M12STP | Timer4_1 count pause signal | <ul style="list-style-type: none"> <li>0b0: Counter continues counting even when the core is stopped</li> <li>0b1: Counter pauses counting when the core is stopped</li> </ul> | RW     |
| 11  | M11STP | Timer2_4 count pause signal | <ul style="list-style-type: none"> <li>0b0: Counter continues counting even when the core is stopped</li> <li>0b1: Counter pauses counting when the core is stopped</li> </ul> | RW     |
| 10  | M10STP | Timer2_3 count pause signal | <ul style="list-style-type: none"> <li>0b0: Counter continues counting even when the core is stopped</li> <li>0b1: Counter pauses counting when the core is stopped</li> </ul> | RW     |
| 9   | M09STP | Timer2_2 count pause signal | <ul style="list-style-type: none"> <li>0b0: Counter continues counting even when the core is stopped</li> <li>0b1: Counter pauses counting when the core is stopped</li> </ul> | RW     |
| 8   | M08STP | Timer2_1 count pause signal | <ul style="list-style-type: none"> <li>0b0: Counter continues counting even when the core is stopped</li> <li>0b1: Counter pauses counting when the core is stopped</li> </ul> | RW     |



| Bit | Name    | Function                    | Description                                                                                                                                                                                                                                                                                                                                                     | Access |
|-----|---------|-----------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 7   | M07STP  | Timer0_2 count pause signal | <ul style="list-style-type: none"> <li>0b0: Counter continues counting even when the core is stopped</li> <li>0b1: Counter pauses counting when the core is stopped</li> </ul>                                                                                                                                                                                  | RW     |
| 6   | M06STP  | Timer0_1 count pause signal | <ul style="list-style-type: none"> <li>0b0: Counter continues counting even when the core is stopped</li> <li>0b1: Counter pauses counting when the core is stopped</li> </ul>                                                                                                                                                                                  | RW     |
| 5   | PVD2STP | PVD2 pause signal           | <ul style="list-style-type: none"> <li>0b0: PVD2 interrupt application or reset occurs even when the core is stopped</li> <li>0b1: PVD2 interrupt application or reset is masked when the core is stopped</li> </ul>                                                                                                                                            | RW     |
| 4   | PVD1STP | PVD1 pause signal           | <ul style="list-style-type: none"> <li>0b0: PVD1 interrupt application or reset occurs even when the core is stopped</li> <li>0b1: PVD1 interrupt application or reset is masked when the core is stopped</li> </ul>                                                                                                                                            | RW     |
| 3   | PVD0STP | PVD0 pause signal           | <ul style="list-style-type: none"> <li>0b0: BOR interrupt application or reset occurs even when the core is stopped</li> <li>0b1: BOR interrupt application or reset is masked when the core is stopped</li> </ul> <div>  <b>Note</b><br/> PVD0 is used as BOR reset. </div> | RW     |
| 2   | RTCSTP  | RTC count pause signal      | <ul style="list-style-type: none"> <li>0b0: Counter continues counting even when the core is stopped</li> <li>0b1: Counter pauses counting when the core is stopped</li> </ul>                                                                                                                                                                                  | RW     |
| 1   | WDTSTP  | WDT count pause signal      | <ul style="list-style-type: none"> <li>0b0: Counter continues counting even when the core is stopped</li> <li>0b1: Counter pauses counting when the core is stopped</li> </ul>                                                                                                                                                                                  | RW     |
| 0   | SWDTSTP | SWDT count pause signal     | <ul style="list-style-type: none"> <li>0b0: Counter continues counting even when the core is stopped</li> <li>0b1: Counter pauses counting when the core is stopped</li> </ul>                                                                                                                                                                                  | RW     |

### 48.3.5 Debug Component Trace Control Register (MCUTRACECTL)

Configure the TRACE output pin through this register.

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |            |   |            |  |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|------------|---|------------|--|
| 0x0024 | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1          | 0 |            |  |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |            |   |            |  |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   | TRACE_IOEN |   | TRACE_MODE |  |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   | RW         |   | RW         |  |

| Bit  | Name       | Function                  | Description                                                                                                                                                                                                                                 | Access |
|------|------------|---------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:3 | Reserved   | Reserved                  | -                                                                                                                                                                                                                                           | -      |
| 2    | TRACE_IOEN | TRACE pin output control  | <ul style="list-style-type: none"> <li>0b0: Synchronous trace pin output disabled</li> <li>0b1: Synchronous trace pin output enabled</li> </ul>                                                                                             | RW     |
| 1:0  | TRACE_MODE | TRACED output pin control | <ul style="list-style-type: none"> <li>0b00: Asynchronous tracing</li> <li>0b01: Synchronous tracing 1 bit TRACED[0]</li> <li>0b10: Synchronous tracing 2 bits TRACED[1:0]</li> <li>0b11: Synchronous tracing 4 bits TRACED[3:0]</li> </ul> | RW     |

### 48.3.6 Peripheral Debug Stop Control Register 2 (MCUSTPCTL2)

Controls peripheral module pausing when the CPU is in debug state.

| Offset | Bit Position |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |        |        |        |        |        |        |        |        |        |        |        |        |    |   |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|----|---|
| 0x0028 | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13     | 12     | 11     | 10     | 9      | 8      | 7      | 6      | 5      | 4      | 3      | 2      | 1  | 0 |
| Reset  | 0x00000000   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |        |        |        |        |        |        |        |        |        |        |        |        |    |   |
| Name   | Reserved     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | M43STP | M42STP | M41STP | M40STP | M39STP | M38STP | M37STP | M36STP | M35STP | M34STP | M33STP | M32STP |    |   |
| Access |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RW     | RW     | RW     | RW     | RW     | RW     | RW     | RW     | RW     | RW     | RW     | RW     | RW |   |

| Bit   | Name     | Function                     | Description                                                                                                                                                                    | Access |
|-------|----------|------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 31:12 | Reserved | Reserved                     | -                                                                                                                                                                              | -      |
| 11    | M43STP   | TimerA_12 count pause signal | <ul style="list-style-type: none"> <li>0b0: Counter continues counting even when the core is stopped</li> <li>0b1: Counter pauses counting when the core is stopped</li> </ul> | RW     |
| 10    | M42STP   | TimerA_11 count pause signal | <ul style="list-style-type: none"> <li>0b0: Counter continues counting even when the core is stopped</li> <li>0b1: Counter pauses counting when the core is stopped</li> </ul> | RW     |
| 9     | M41STP   | TimerA_10 count pause signal | <ul style="list-style-type: none"> <li>0b0: Counter continues counting even when the core is stopped</li> <li>0b1: Counter pauses counting when the core is stopped</li> </ul> | RW     |
| 8     | M40STP   | TimerA_9 count pause signal  | <ul style="list-style-type: none"> <li>0b0: Counter continues counting even when the core is stopped</li> <li>0b1: Counter pauses counting when the core is stopped</li> </ul> | RW     |
| 7     | M39STP   | TimerA_8 count pause signal  | <ul style="list-style-type: none"> <li>0b0: Counter continues counting even when the core is stopped</li> <li>0b1: Counter pauses counting when the core is stopped</li> </ul> | RW     |
| 6     | M38STP   | TimerA_7 count pause signal  | <ul style="list-style-type: none"> <li>0b0: Counter continues counting even when the core is stopped</li> <li>0b1: Counter pauses counting when the core is stopped</li> </ul> | RW     |
| 5     | M37STP   | TimerA_6 count pause signal  | <ul style="list-style-type: none"> <li>0b0: Counter continues counting even when the core is stopped</li> <li>0b1: Counter pauses counting when the core is stopped</li> </ul> | RW     |

| Bit | Name   | Function                    | Description                                                                                                                                                                    | Access |
|-----|--------|-----------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 4   | M36STP | TimerA_5 count pause signal | <ul style="list-style-type: none"> <li>0b0: Counter continues counting even when the core is stopped</li> <li>0b1: Counter pauses counting when the core is stopped</li> </ul> | RW     |
| 3   | M35STP | TimerA_4 count pause signal | <ul style="list-style-type: none"> <li>0b0: Counter continues counting even when the core is stopped</li> <li>0b1: Counter pauses counting when the core is stopped</li> </ul> | RW     |
| 2   | M34STP | TimerA_3 count pause signal | <ul style="list-style-type: none"> <li>0b0: Counter continues counting even when the core is stopped</li> <li>0b1: Counter pauses counting when the core is stopped</li> </ul> | RW     |
| 1   | M33STP | TimerA_2 count pause signal | <ul style="list-style-type: none"> <li>0b0: Counter continues counting even when the core is stopped</li> <li>0b1: Counter pauses counting when the core is stopped</li> </ul> | RW     |
| 0   | M32STP | TimerA_1 count pause signal | <ul style="list-style-type: none"> <li>0b0: Counter continues counting even when the core is stopped</li> <li>0b1: Counter pauses counting when the core is stopped</li> </ul> | RW     |

## A Glossary

| Terminology | Full Name                     | description                                                                                                                                                                                                              |
|-------------|-------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Byte        | -                             | Bytes, 8-bit data                                                                                                                                                                                                        |
| Half Word   | -                             | Half word, 16-bit data                                                                                                                                                                                                   |
| Word        | -                             | Words, 32-bit data                                                                                                                                                                                                       |
| Double Word | -                             | Double word, 64 -bit data.                                                                                                                                                                                               |
| IAP         | -                             | It refers to reprogram the MCU's Flash during the running of a user program (programming in the application).                                                                                                            |
| ICP         | -                             | It means that the MCU's Flash of the MCU can be programmed by using JTAG protocol, SWD protocol or bootstrap program when the device is installed on the user application circuit board (in-circuit/online programming). |
| AHB         | Advanced High-performance Bus | Advanced High-Performance Bus                                                                                                                                                                                            |
| APB         | Advanced Peripheral Bus       | Advanced peripheral bus                                                                                                                                                                                                  |
| DMA         | Direct Memory Access          | Direct memory access                                                                                                                                                                                                     |
| TIM/TMR     | -                             | Timer                                                                                                                                                                                                                    |

## Revision History

| Revision/Date            | Changes                 |
|--------------------------|-------------------------|
| Rev1.00<br>Jun. 16, 2025 | First official release. |