

HC32K11x Series

32-bit ARM[®] Cortex[®]-M0+ Microcontroller

Reference Manual

Rev1.00
November 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:	https://www.xhsc.com.cn
E-mail:	XHSC_MCU@xhsc.com.cn (Business Consulting) XHSC_FAE@xhsc.com.cn (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 HC32K11x Series 32-bit ARM® Cortex®-M0+ 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

Preface	iii
1 Product Introduction	27
2 Memory Mapping	28
2.1 Memory Mapping.....	28
2.2 Bit Segment Space.....	30
3 Bus Architecture (BUS)	32
3.1 Introduction.....	32
3.2 Bus Architecture.....	33
3.3 Bus Function.....	33
4 Reset Module Unit (RMU)	34
4.1 Introduction.....	34
4.2 Functional Description.....	34
4.2.1 Reset Mode and Reset Flag Bit.....	34
4.2.2 Reset Timing.....	35
4.2.3 Reset Mode Judgment.....	40
4.3 Register Description.....	41
4.3.1 Register List.....	41
4.3.2 Reset Flag Register 0 (RMU_RSTF0).....	42
5 Clock Module Unit (CMU)	44
5.1 Introduction.....	44
5.2 Features.....	44
5.3 Functional Description.....	45
5.3.1 System Block Diagram.....	45
5.3.2 Working Clock Specifications.....	46
5.3.3 Crystal Oscillator Circuit.....	46
5.3.4 Internal RC Clock.....	49
5.3.5 Clock Switching Steps.....	50
5.3.6 Clock Output Function.....	51
5.3.7 Clock Frequency Measurement.....	52
5.4 Register Description.....	53
5.4.1 Register List.....	53
5.4.2 CMU XTAL Config Register (CMU_XTALCFGR).....	55
5.4.3 CMU XTAL Stable Control Register (CMU_XTALSTBCR).....	56
5.4.4 CMU XTAL Control Register (CMU_XTALCR).....	57
5.4.5 CMU XTAL Clock Stop Detect Control Register (CMU_XTALSTDCR).....	58
5.4.6 CMU XTAL Clock Stop Detect Status Register (CMU_XTALSTDSR).....	59
5.4.7 CMU XTAL32 Config Register (CMU_XTAL32CFGR).....	60
5.4.8 CMU XTAL32 Filter Control Register (CMU_XTAL32NFR).....	61
5.4.9 CMU XTAL32 Control Register (CMU_XTAL32CR).....	61
5.4.10 CMU HRC Config Register (CMU_HRCCFGR).....	62
5.4.11 CMU HRC Control Register (CMU_HRCCR).....	63
5.4.12 CMU LRC Control Register (CMU_LRCCR).....	64

5.4.13 CMU Clock Source Stable Status Register (CMU_OSCSTBSR).....	65
5.4.14 CMU System Clock Source Switch Register (CMU_CKSWR).....	66
5.4.15 CMU Clock Divider Config Register (CMU_SCKDIVR).....	67
5.4.16 CMU MCAN Clock Config Register (CMU_CANCKCFGR).....	68
5.4.17 CMU ADC Clock Config Register (CMU_PERICKSEL).....	69
5.4.18 CMU Function Clock Control Register (CMU_FCG).....	70
5.4.19 CMU MCO1 Clock Output Config Register (CMU_MCO1CFGR).....	72
5.4.20 FCM Lower Limit Control Register (FCM_LVR)	73
5.4.21 FCM Upper Limit Control Register (FCM_UVR)	73
5.4.22 FCM Counter Register (FCM_CNTR)	74
5.4.23 FCM Start Control Register (FCM_STR).....	74
5.4.24 FCM Measurement Control Register (FCM_MCCR)	75
5.4.25 FCM Reference Control Register (FCM_RCCR)	76
5.4.26 FCM Reset Interrupt Control Register (FCM_RIER)	78
5.4.27 FCM Flag Register (FCM_SR).....	79
5.4.28 FCM Flag Clear Register (FCM_CLR)	79
6 Power Control (PWC).....	80
6.1 Introduction.....	80
6.2 Functional Description.....	80
6.2.1 Power Distribution.....	80
6.2.2 Power Voltage Detection Unit (LVD) Description.....	81
6.2.3 Action Mode and Low Power Mode.....	85
6.2.4 Methods to Reduce Power Consumption.....	88
6.3 Register Description.....	89
6.3.1 Register List.....	89
6.3.2 Stop Mode Control Register (PWC_STPMCR).....	90
6.3.3 Power Mode Control Register (PWC_PWRC).....	90
6.3.4 Power Monitoring Register (PWC_PWRMON).....	92
6.3.5 RAM Control Register (PWC_RAMCR).....	92
6.3.6 Function Protection Control Register (PWC_FPRC).....	93
6.3.7 LVD1 ICG Control Register (PWC_LVD1ICGCR).....	94
6.3.8 LVD2 Control Register (PWC_LVD2CR).....	95
6.3.9 LVD Control and Status Register (PWC_LVDCSR).....	97
6.3.10 Low Power Mode Status Register (PWC_LPMCSR).....	99
6.3.11 Low Power Mode Status Clear Register (PWC_LPMCSCR).....	99
6.3.12 Debug Mode Control Register (PWC_DBGCR).....	100
7 Initialization Configuration (ICG).....	101
7.1 Introduction.....	101
7.2 Register Description.....	101
7.2.1 Register List.....	101
7.2.2 Initialization Configuration Register A0 (ICGA0x) (x=A/B/C).....	104
7.2.3 Initialization Configuration Register A1 (ICGA1x) (x=A/B/C).....	106
7.2.4 Initialization Configuration Register A2 (ICGA2x) (x=A/B/C).....	108
7.2.5 Initialization Configuration Register A3 (ICGA3x) (x=A/B/C).....	109
7.2.6 Initialization Configuration Register A4 (ICGA4x) (x=A/B/C).....	109

7.2.7 Initialization Configuration Register B0 (ICGB0).....	111
7.2.8 Initialization Configuration Register B1 (ICGB1).....	111
8 Embedded Flash Memory (EFM).....	113
8.1 Introduction.....	113
8.2 Features.....	113
8.3 Functional Description.....	114
8.3.1 Embedded Flash.....	114
8.3.2 Read Interface.....	114
8.3.3 Flash Read Prefetch.....	116
8.3.4 Flash Programming and Erasing Operations.....	116
8.3.5 One-Time Programmable Byte (OTP).....	121
8.3.6 Boot Swap.....	123
8.3.7 Data Reliability and Integrity.....	125
8.3.8 Interrupts and Reset.....	126
8.4 Register Description.....	129
8.4.1 Register List.....	129
8.4.2 Flash Access Protect Register (EFM_(D)FAPRT).....	131
8.4.3 Flash Key 1 Register (EFM_(D)KEY1).....	131
8.4.4 Flash Key 2 Register (EFM_(D)KEY2).....	132
8.4.5 Flash Stop Register (EFM_(D)FSTP).....	132
8.4.6 Flash Read Mode Register (EFM_(D)FRMC).....	133
8.4.7 Flash Write Mode Register (EFM_(D)FWMC).....	134
8.4.8 Flash State Register (EFM_(D)FSR).....	136
8.4.9 Flash State Clear Register (EFM_(D)FSCLR).....	139
8.4.10 Flash Interrupt Enable Register (EFM_(D)FITE).....	141
8.4.11 Flash Boot Swap Status Register (EFM_FSWP).....	142
8.4.12 Chip Identification Register (EFM_CHIPID).....	143
8.4.13 UNIQUE ID Register (EFM_UQID0).....	143
8.4.14 UNIQUE ID Register (EFM_UQID1).....	144
8.4.15 UNIQUE ID Register (EFM_UQID2).....	144
8.4.16 ECC Control Register (EFM_(D)FEC).....	145
8.4.17 ECC Error Address Register (EFM_(D)FECERA).....	146
8.4.18 Error Injection Register 0 (EFM_(D)ERI0).....	146
8.4.19 Error Injection Register 1 (EFM_ERI1).....	147
8.4.20 Error Injection Register 2 (EFM_(D)ERI2).....	147
8.4.21 Flash Write Protection Register 0 (EFM_(D)FNWPRT0).....	148
8.4.22 ProgramFLASH Write Protection Register 1 (EFM_FNWPRT1).....	150
8.4.23 EDataFLASH Write Protection Register 1 (EFM_DFNWPRT1).....	151
8.5 Precautions.....	152
9 Built-in SRAM (SRAM).....	153
9.1 Introduction.....	153
9.2 Functional Description.....	153
9.2.1 SRAM Space Allocation.....	153
9.2.2 ECC Check Error Injection.....	153
9.3 Register Description.....	155

9.3.1 Register List.....	155
9.3.2 SRAM Check Control Register (SRAMC_CKCR).....	156
9.3.3 SRAM Check Protect Register (SRAMC_CKPR).....	157
9.3.4 SRAM Check Status Register (SRAMC_CKSR).....	158
9.3.5 SRAM1 Error Injection Enable Register (SRAM1_EIEN).....	159
9.3.6 SRAM1 Error Injection Bit Enable Register 0 (SRAM1_EIBIT0).....	159
9.3.7 SRAM1 Error Injection Bit Enable Register 1 (SRAM1_EIBIT1).....	160
9.3.8 SRAM1 Error Address Register (SRAM1_ECCERRADDR).....	160
9.4 Precautions.....	161
10 General-Purpose I/O (GPIO).....	162
10.1 Introduction.....	162
10.2 Features.....	162
10.3 Functional Description.....	163
10.3.1 Function Block Diagram.....	163
10.3.2 General Input/Output GPIO Function.....	163
10.3.3 Peripheral Function.....	164
10.3.4 Dual Peripheral Function.....	164
10.3.5 Event Port Input and Output Functions.....	164
10.3.6 External Interrupt EIRQ Input Function.....	165
10.3.7 Analog Function.....	165
10.3.8 Reset Terminal Multiplexing Configuration.....	165
10.3.9 General Control.....	165
10.4 Register Description.....	167
10.4.1 Register List.....	167
10.4.2 General Input Data Register (PIDRx).....	169
10.4.3 General Output Data Register (PODRx).....	169
10.4.4 General Output Enable Register (POERx).....	171
10.4.5 General Output Set Register (POSRx).....	171
10.4.6 General Output Reset Register (PORRx).....	173
10.4.7 General Output Toggle Register (POTRx).....	173
10.4.8 Special Control Register (PSPCR).....	175
10.4.9 Common Control Register (PCCR).....	176
10.4.10 Input Control Register (PINAER).....	177
10.4.11 Write Protection Register (PWPR).....	178
10.4.12 General Control Register (PCRxy).....	179
10.4.13 Function Select Register (PFSRxy).....	181
10.4.14 Event Port Direction Selection Register (PEVNTDIRRm).....	182
10.4.15 Event Port Input Data Register (PEVNTIDRm).....	182
10.4.16 Event Port Output Data Register (PEVNTODRm).....	183
10.4.17 Event Port Output Data Reset Register (PEVNTORRm).....	184
10.4.18 Event Port Output Data Set Register (PEVNTOSRm).....	185
10.4.19 Event Port Rising Edge Input Enable Register (PEVNTRISRm).....	186
10.4.20 Event Port Falling Edge Input Enable Register (PEVNTFALRm).....	187
10.4.21 Event Port Input Filter Control Register (PEVNTNFCR).....	188
10.5 Precautions.....	189

11 Interrupt Controller (INTC).....	190
11.1 Introduction.....	190
11.2 Vector Digraph.....	191
11.2.1 Interrupt Vector Digraph.....	191
11.2.2 Interrupt Event List.....	192
11.3 Functional Description.....	196
11.3.1 Function Block Diagram.....	196
11.3.2 Non-maskable Interrupt.....	196
11.3.3 EIRQ Pin Interrupt.....	197
11.3.4 Software Trigger Event.....	197
11.3.5 Interrupt Source Selection.....	197
11.3.6 WFE Event Selection.....	197
11.3.7 WFE Wake-up Event Management.....	197
11.3.8 Digital Filter.....	198
11.3.9 Low Power Mode Return.....	199
11.4 Register Description.....	200
11.4.1 Register List.....	200
11.4.2 Function Protection Control Register (INTC_FPRCR).....	201
11.4.3 NMI Pin Interrupt Control Register (INTC_NMICR).....	202
11.4.4 Non-maskable Interrupt Enable Register (INTC_NMIER).....	203
11.4.5 Non-maskable Interrupt Flag Register (INTC_NMIFR).....	205
11.4.6 Non-maskable Interrupt Flag Clear Register (INTC_NMIFCR).....	206
11.4.7 EIRQ Pin Interrupt Control Register (INTC_EIRQCRx) (x=0-7).....	207
11.4.8 EIRQ Pin Interrupt Flag Register (INTC_EIFR).....	208
11.4.9 EIRQ Pin Interrupt Flag Clear Register (INTC_EIFCR).....	208
11.4.10 Interrupt Event Enable Register (INTC_INTENx) (x=0-3).....	209
11.4.11 Wake-up Event Enable Register (INTC_WKEN).....	210
11.4.12 WFE Event Enable Register (INTC_EVTER).....	211
11.4.13 USART/SPI Wake-up Event Flag Register (INTC_WKFR).....	212
11.4.14 USART/SPI Wake-up Event Flag Clear Register (INTC_WKFCR).....	213
11.5 Precautions.....	213
12 Automatic Operation System (AOS).....	214
12.1 Introduction.....	214
12.2 Features.....	214
12.3 Functional Description.....	215
12.3.1 Function Block Diagram.....	215
12.3.2 AOS Source Events and Targets.....	215
12.3.3 Dedicated Trigger Source.....	216
12.3.4 Common Trigger Source.....	216
12.4 Register Description.....	217
12.4.1 Register List.....	217
12.4.2 Software Trigger Event Register (INTSFTTRG).....	218
12.4.3 DMA Transfer Start Trigger Source Selection Register (DMA_TRGSELx) (x=0-3).....	219
12.4.4 Event Port Trigger Source Selection Register (PEVNT_TRGSEL12/PEVNT_TRGSEL34).....	220
12.4.5 Timer0 Hardware Trigger Event Selection Register (TMR0_TRGSEL).....	221

12.4.6 Timer2 Hardware Trigger Event Selection Register (TMR2_TRGSELx) (x=0-1).....	222
12.4.7 Timer4 Hardware Trigger Event Selection Register (TMR4_TRGSEL).....	223
12.4.8 TimerA Hardware Trigger Event Selection Register (TMRA_TRGSELx) (x=0-1).....	224
12.4.9 ADC Conversion Start On-Chip Trigger Source Selection Register (ADC_TRGSELx) (x=0-1).....	225
12.4.10 Common Trigger Source Selection Register (AOS_COMTRGSELx) (x=1-2).....	226
13 Clock Trimming Controller (CTC).....	227
13.1 Introduction.....	227
13.2 Features.....	227
13.3 Functional Description.....	227
13.3.1 Function Block Diagram.....	227
13.3.2 Reference Clock.....	227
13.3.3 Frequency Calibration.....	235
13.3.4 Interrupt.....	236
13.4 Operation Example.....	236
13.4.1 Use XTAL32 as the Reference Clock to Calibrate HRC to 48MHz with a Calibration Target Deviation of Less Than 0.5%.....	236
13.5 Register Description.....	237
13.5.1 Register List.....	237
13.5.2 CTC Control Register 1 (CTC_CR1).....	238
13.5.3 CTC Control Register 2 (CTC_CR2).....	241
13.5.4 CTC Status Register (CTC_STR).....	243
13.5.5 CTC Counter Register (CTC_CNT).....	244
14 DMA Controller (DMA).....	245
14.1 Introduction.....	245
14.2 Features.....	245
14.3 Functional Description.....	246
14.3.1 DMA Block Diagram.....	246
14.3.2 Enabling the DMA Controller.....	246
14.3.3 Channel Selection and Channel Priority.....	246
14.3.4 Start DMA.....	247
14.3.5 Data Block.....	247
14.3.6 Transfer Address Control.....	247
14.3.7 Transfer Count.....	247
14.3.8 Chain Transfer.....	248
14.3.9 NonSequence Address Transfer.....	249
14.3.10 Early Termination of Transfer.....	250
14.3.11 Interrupt.....	250
14.4 Operation Example.....	251
14.4.1 Memory-to-Memory Transfer.....	251
14.4.2 Memory-to-Peripheral Transfer.....	252
14.4.3 Memory-to-Memory Chain Transfer.....	253
14.5 Register Description.....	255
14.5.1 Register List.....	255
14.5.2 DMA Enable Register (DMA_EN).....	256
14.5.3 Interrupt Status Register 0 (DMA_INTSTAT0).....	257

14.5.4 Interrupt Status Register 1 (DMA_INTSTAT1).....	258
14.5.5 Interrupt Mask Register 0 (DMA_INTMASK0).....	259
14.5.6 Interrupt Mask Register 1(DMA_INTMASK1).....	260
14.5.7 Interrupt Status Clear Register 0 (DMA_INTCLR0).....	261
14.5.8 Interrupt Status Clear Register 1 (DMA_INTCLR1).....	262
14.5.9 Channel Enable Register (DMA_CHEN).....	263
14.5.10 Channel Status Register (DMA_CHSTAT).....	264
14.5.11 Chain Descriptor Base Address Register 1 (DMA_BASE1_LL).....	265
14.5.12 Chain Descriptor Base Address Register 2 (DMA_BASE2_LL).....	265
14.5.13 Channel Enable Reset Register (DMA_CHENCLR).....	266
14.5.14 Transmission Source Address Register (DMA_SARx) (x=0-3).....	266
14.5.15 Transmission Destination Address Register (DMA_DARx) (x=0-3).....	267
14.5.16 Transfer Data Control Register (DMA_CHxCTL0) (x=0-3).....	268
14.5.17 Channel Control Register (DMA_CHxCTL1) (x=0-3).....	269
15 Voltage Comparator (CMP).....	271
15.1 Introduction.....	271
15.2 Features.....	271
15.3 Functional Description.....	271
15.3.1 Function Block Diagram.....	271
15.3.2 Input Selection.....	272
15.3.3 Normal Compare.....	272
15.3.4 Window Output.....	273
15.3.5 Noise Filter.....	275
15.3.6 External Pin Output.....	275
15.3.7 Built-in DAC.....	275
15.3.8 Interrupt and Event.....	276
15.4 Register Description.....	277
15.4.1 Register List.....	277
15.4.2 Comparator Working Mode Register (CMP_MDR).....	278
15.4.3 Comparator Filter and Interrupt Register (CMP_FIR).....	279
15.4.4 Comparator Output Control Register (CMP_OCR).....	281
15.4.5 Comparator Output Monitoring Register (CMP_OMR).....	282
15.4.6 Comparator Positive and Negative Input Selection Register (CMP_PMSR).....	283
15.4.7 Comparator DAC Data Register (CMP_DADR).....	285
15.4.8 Comparator Digital to Analog Conversion Control Register (CMP_DACR).....	285
15.4.9 Comparator Blank Window Output Source Register 1 (CMP_BWSR1).....	286
15.4.10 Comparator Blank Window Output Source Register 2 (CMP_BWSR2).....	287
15.5 Precautions.....	288
15.5.1 Module Stop Function.....	288
15.5.2 Power Consumption During Module Stop.....	288
15.5.3 Power Consumption During Stop Low Power Mode.....	288
16 Analog-to-Digital Converter (ADC).....	289
16.1 Introduction.....	289
16.2 Features.....	289
16.3 Functional Description.....	290

16.3.1 Function Block Diagram.....	290
16.3.2 ADC Clock.....	290
16.3.3 Channel Selection.....	290
16.3.4 Trigger Source Selection.....	291
16.3.5 Sequence A Single Scan Mode.....	291
16.3.6 Sequence A Continuous Scan Mode.....	292
16.3.7 Dual Sequence Scan Mode.....	293
16.3.8 Analog Watchdog Function.....	295
16.3.9 Sampling Time and Conversion Time for Analog Inputs.....	295
16.3.10 ADC Data Register Automatic Clear Function.....	296
16.3.11 Interrupt and Events.....	297
16.4 Register Description.....	298
16.4.1 Register List.....	298
16.4.2 ADC Start Register (ADC_STR).....	299
16.4.3 ADC Control Register 0 (ADC_CR0).....	300
16.4.4 ADC Control Register 1 (ADC_CR1).....	301
16.4.5 ADC Trigger Select Register (ADC_TRGSR).....	302
16.4.6 ADC Sequence A Channel Select Register (ADC_CHSELRA).....	304
16.4.7 ADC Sequence B Channel Select Register (ADC_CHSELRB).....	305
16.4.8 ADC Channel x Sampling Time Register (ADC_SSTR).....	306
16.4.9 ADC Interrupt Flag Register (ADC_ISR).....	307
16.4.10 ADC Interrupt Enable Register (ADC_ICR).....	308
16.4.11 ADC Interrupt Flag Clear Register (ADC_ISCLR).....	309
16.4.12 ADC Channel x Convert Data Register (ADC_DRx) (x=0-15).....	310
16.4.13 ADC Analog Watchdog Control Register (ADC_AWDCR).....	312
16.4.14 ADC Analog Watchdog Status Register (ADC_AWDSR).....	314
16.4.15 ADC Analog Watchdog Status Clear Register (ADC_AWDSCLR).....	315
16.4.16 ADC Analog Watchdog x Compare Data Register y (ADC_AWDxDRy) (x=0-1, y=0-1).....	316
16.4.17 ADC Analog Watchdog x Channel Select Register (ADC_AWDxCHSR) (x=0-1).....	317
16.5 Precautions.....	318
16.5.1 Precautions for Reading Data Registers.....	318
16.5.2 Precautions for Handling Scan Completion Interrupts.....	318
16.5.3 Precautions for Module Stop and Low Power Settings.....	318
16.5.4 ADC Conversion Analog Channel Input Pin Settings.....	318
16.5.5 Noise Control.....	318
17 General Control Timer (Timer4).....	319
17.1 Introduction.....	319
17.2 Features.....	319
17.3 Functional Description.....	320
17.3.1 Timer4 Block Diagram.....	320
17.3.2 Basic Actions.....	321
17.3.3 Buffer Function.....	324
17.3.4 General PWM Output.....	329
17.3.5 Periodic Interval Response.....	335
17.3.6 EMB Control.....	337

17.3.7 Monitoring Output.....	338
17.3.8 Hardware Startup.....	338
17.3.9 Synchronous Starting.....	338
17.3.10 Interrupt and Event.....	338
17.4 Register Description.....	341
17.4.1 Register List.....	341
17.4.2 Count Value Register (TMR4_CNTR).....	344
17.4.3 Period Value Register (TMR4_CPSR).....	344
17.4.4 Control Status Register (TMR4_CCSR).....	345
17.4.5 Valid Period Register (TMR4_CVPR).....	347
17.4.6 General Compare Reference Register (TMR4_OCCRm).....	349
17.4.7 General Control Status Register (TMR4_OCSRn).....	350
17.4.8 General Extended Control Register (TMR4_OCERn).....	352
17.4.9 General Mode Control Register (TMR4_OCMRm).....	355
17.4.10 Special Compare Reference Register (TMR4_SCCRm).....	363
17.4.11 General Control Status Register (TMR4_SCSRm).....	364
17.4.12 Special Extended Control Register (TMR4_SCER).....	366
17.4.13 Special Mode Control Register (TMR4_SCMRm).....	367
17.4.14 PWM Control Register (TMR4_POCRn).....	368
17.4.15 PWM Status Control Register (TMR4_PSCR).....	369
17.4.16 PWM Filter Control Register (TMR4_PFSRn).....	373
17.4.17 PWM Dead Zone Control Register A (TMR4_PDARn).....	374
17.4.18 PWM Dead Zone Control Register B (TMR4_PDBRn).....	374
17.4.19 Reload Control Status Register (TMR4_RCSR).....	375
17.4.20 Special Interrupt Control Register (TMR4_SCIR).....	377
17.4.21 Special Flag Control Register (TMR4_SCFR).....	378
18 Emergency Brake Module (EMB).....	379
18.1 Introduction.....	379
18.2 Functional Description.....	379
18.2.1 EMB Channel Function Block Diagram.....	379
18.2.2 EMB Structure Block Diagram.....	381
18.2.3 Control PWM Signal Output When External Port Input Level Changes.....	381
18.2.4 Stop PWM Signal Output When PWM Output Port Level Occurs In-phase (same high or same low).....	382
18.2.5 Stop PWM Signal Output According to Voltage Comparator Compare Result.....	383
18.2.6 Stop PWM Signal Output when System Error Occurs.....	383
18.2.7 Write Register Software Control PWM Signal Output.....	383
18.3 Register Description.....	384
18.3.1 Register List.....	384
18.3.2 EMB Control Register 1 (EMB_CTL1)	385
18.3.3 EMB Control Register 2 (EMB_CTL2).....	388
18.3.4 EMB Software Output Enable Register (EMB_SOE).....	389
18.3.5 EMB Status Register (EMB_STAT).....	390
18.3.6 EMB Status Clear Register (EMB_STATCLR).....	392
18.3.7 EMB Interrupt Enable Register (EMB_INTEN).....	393
18.3.8 EMB Control PWM Output Release Select Register (EMB_RLSSEL).....	394

19 General Timer (TimerA).....	395
19.1 Introduction.....	395
19.2 Features.....	395
19.3 Functional Description.....	396
19.3.1 TimerA Block Diagram.....	396
19.3.2 Basic Actions.....	396
19.3.3 Clock Source Selection.....	398
19.3.4 Count Direction.....	399
19.3.5 Digital Filtering.....	399
19.3.6 Synchronous Starting.....	399
19.3.7 Buffer Function.....	400
19.3.8 General PWM Output.....	401
19.3.9 Quadrature Encoding Counting.....	402
19.3.10 Interrupt and Events.....	404
19.4 Register Description.....	406
19.4.1 Register List.....	406
19.4.2 General Counter Value Register (TMRA_CNTER).....	407
19.4.3 Period Reference Value Register (TMRA_PERAR).....	408
19.4.4 Compare Reference Value Register (TMRA_CMPARm) (m=1-8).....	409
19.4.5 Basic Control and Status Register L (TMRA_BCSTRL).....	410
19.4.6 Basic Control and Status Register H (TMRA_BCSTRH).....	411
19.4.7 Interrupt Control Register (TMRA_ICONR).....	412
19.4.8 Event Control Register (TMRA_ECONR).....	414
19.4.9 Filter Control Register (TMRA_FCONR).....	416
19.4.10 Status Flag Register (TMRA_STFLR).....	417
19.4.11 Buffer Control Register (TMRA_BCONRm) (m=1-4).....	420
19.4.12 Capture Control Register (TMRA_CCONRm) (m=1-8).....	421
19.4.13 Port Control Register (TMRA_PCONRm) (m=1-8).....	424
19.4.14 Hardware Control Event Select Register (TMRA_HCONR).....	426
19.4.15 Hardware Count Up Event Select Register (TMRA_HCUPR).....	429
19.4.16 Hardware Count Down Event Select Register (TMRA_HCDOR).....	431
19.4.17 Counter Synchronization Enable Register (TMR_SYNENR).....	433
20 General Timer (Timer2).....	434
20.1 Introduction.....	434
20.2 Features.....	434
20.3 Functional Description.....	434
20.3.1 Basic Block Diagram.....	434
20.3.2 Clock Source Selection.....	435
20.3.3 Compare Output.....	436
20.3.4 Hardware Triggering.....	437
20.3.5 Digital Filtering.....	439
20.3.6 Interrupt and Events.....	440
20.4 Register Description.....	441
20.4.1 Register List.....	441
20.4.2 Count Value Register m (TMR2_CNTmR) (m=A/B).....	442

20.4.3 Reference Value Register m (TMR2_CMPmR) (m=A/B).....	443
20.4.4 Basic Control Register (TMR2_BCONR).....	444
20.4.5 Interrupt Control Register (TMR2_ICONR).....	447
20.4.6 Port Control Register (TMR2_PCONR).....	448
20.4.7 Hardware Control Register (TMR2_HCONR).....	451
20.4.8 Status Flag Register (TMR2_STFLR).....	455
20.5 Precautions.....	457
21 General Timer (Timer0).....	458
21.1 Introduction.....	458
21.2 Functional Description.....	458
21.2.1 Timer0 Block Diagram.....	458
21.2.2 Clock Source Selection.....	458
21.2.3 Basic Counting.....	459
21.2.4 Hardware Triggering.....	459
21.2.5 Interrupt and Event.....	459
21.3 Register Description.....	461
21.3.1 Register List.....	461
21.3.2 Counter Value Register (TMR0_CNTAR).....	462
21.3.3 Compare Value Register (TMR0_CMPAR).....	462
21.3.4 Basic Control Register (TMR0_BCONR).....	463
21.3.5 Status Flag Register (TMR0_STFLR).....	465
21.4 Precautions.....	466
22 Real Time Clock (RTC).....	467
22.1 Introduction.....	467
22.2 Features.....	467
22.3 Functional Description.....	468
22.3.1 RTC Block Diagram.....	468
22.3.2 Power-on Setting.....	468
22.3.3 System Low Power Mode Switching.....	468
22.3.4 Clock Error Compensation.....	468
22.3.5 Interrupt.....	469
22.4 Operation Example.....	469
22.4.1 RTC Count Start Setting.....	469
22.4.2 Read Count Register.....	469
22.4.3 Write Count Register.....	469
22.4.4 Alarm Clock Setting.....	470
22.4.5 1Hz Output.....	470
22.5 Register Description.....	472
22.5.1 Register List.....	472
22.5.2 Control Register 0 (RTC_CR0).....	473
22.5.3 Control Register 1 (RTC_CR1).....	474
22.5.4 Control Register 2 (RTC_CR2).....	475
22.5.5 Control Register 3 (RTC_CR3).....	476
22.5.6 Second Count Register (RTC_SEC).....	477
22.5.7 Minute Count Register (RTC_MIN).....	477

22.5.8 Hour Count Register (RTC_HOUR).....	478
22.5.9 Day Count Register (RTC_DAY).....	479
22.5.10 Week Count Register (RTC_WEEK).....	480
22.5.11 Month Count Register (RTC_MON).....	481
22.5.12 Year Count Register (RTC_YEAR).....	482
22.5.13 Minute Alarm Register (RTC_ALMMIN).....	482
22.5.14 Hour Alarm Register (RTC_ALMHOUR).....	483
22.5.15 Week Alarm Register (RTC_ALMWEEK).....	483
22.5.16 Clock Error Compensation Register (RTC_ERRCRH, RTC_ERRCRL).....	484
22.5.17 Flag Clear Register (RTC_FCLR).....	488
22.6 Precautions.....	488
23 Watchdog Timer (WDT/SWDT).....	489
23.1 Introduction.....	489
23.2 Features.....	489
23.3 Functional Description.....	489
23.3.1 Start the Watchdog.....	489
23.3.2 Hardware Startup Mode.....	489
23.3.3 Software Startup Mode.....	490
23.3.4 Refresh Action.....	491
23.3.5 Flag Bit.....	492
23.3.6 Interrupt/Reset.....	492
23.3.7 Count Underflow.....	492
23.3.8 Refresh Error.....	494
23.4 Register Description.....	495
23.4.1 Register List.....	495
23.4.2 Control Register (SWDT_CR).....	496
23.4.3 Status Register (SWDT_SR).....	497
23.4.4 Refresh Register (SWDT_RR).....	498
23.5 Precautions.....	498
24 Universal Synchronous Asynchronous Receiver/Transmitter (USART).....	499
24.1 Introduction.....	499
24.2 Features.....	499
24.3 Functional Description.....	501
24.3.1 System Block Diagram.....	501
24.3.2 UART Function.....	502
24.3.3 Multiprocessor Communication.....	511
24.3.4 UART-LIN.....	515
24.3.5 Smart Card.....	517
24.3.6 Clock Synchronization Mode.....	521
24.3.7 Digital Filter Function.....	525
24.3.8 Interrupt.....	527
24.4 Register Description.....	528
24.4.1 Register List.....	528
24.4.2 Status Register (USART_SR).....	529
24.4.3 Transmit Data Register (USART_TDR).....	534

24.4.4 Receive Data Register (USART_RDR).....	534
24.4.5 Baud Rate Register (USART_BRR).....	535
24.4.6 Control Register 1 (USART_CR1).....	537
24.4.7 Control Register 2 (USART_CR2).....	542
24.4.8 Control Register 3 (USART_CR3).....	547
24.4.9 Prescaler Register (USART_PR).....	550
24.4.10 LIN Baudrate Measuring Counter Register (USART_LBMC).....	552
24.5 Precautions.....	553
24.5.1 UART Precautions.....	553
24.5.2 Clock Synchronous Mode Precautions.....	553
24.5.3 Other Precautions.....	553
25 Inter-Integrated Circuit (I2C).....	554
25.1 Introduction.....	554
25.2 Features.....	554
25.3 Functional Description.....	555
25.3.1 I2C System Block Diagram.....	555
25.3.2 I2C Protocol.....	556
25.3.3 Address Matching.....	563
25.3.4 SMBus Action.....	566
25.3.5 Reset.....	567
25.3.6 Programmable Digital Filtering.....	568
25.3.7 Interrupt and Event.....	568
25.4 Operation Example.....	569
25.4.1 I2C Initialization Process.....	569
25.5 Register Description.....	571
25.5.1 Register List.....	571
25.5.2 I2C Control Register 1 (I2C_CR1).....	572
25.5.3 I2C Control Register 2 (I2C_CR2).....	574
25.5.4 I2C Control Register 3 (I2C_CR3).....	576
25.5.5 I2C Control Register 4 (I2C_CR4).....	577
25.5.6 I2C Slave Address Register 0 (I2C_SLR0).....	579
25.5.7 I2C Slave Address Register 1 (I2C_SLR1).....	580
25.5.8 I2C SCL Level Timeout Control Register (I2C_SLTR).....	580
25.5.9 I2C Status Register (I2C_SR).....	582
25.5.10 I2C Status Clear Register (I2C_CLR).....	588
25.5.11 I2C Data Transmit Register (I2C_DTR).....	590
25.5.12 I2C Data Receive Register (I2C_DRR).....	591
25.5.13 I2C Clock Control Register (I2C_CCR).....	592
25.5.14 I2C Filter Control Register (I2C_FLTR).....	593
26 Serial Peripheral Interface (SPI).....	594
26.1 Introduction.....	594
26.2 Features.....	594
26.3 Functional Description.....	595
26.3.1 System Block Diagram.....	595
26.3.2 SPI Action System Description.....	596

26.3.3 Data Communication Description.....	598
26.3.4 Operating Instructions.....	604
26.3.5 Self-diagnosis of Parity.....	609
26.3.6 Error Detection.....	610
26.3.7 SPI Initialization.....	613
26.3.8 Interrupt and Events.....	614
26.4 Register Description.....	615
26.4.1 Register List.....	615
26.4.2 SPI Data Register (SPI_DR).....	616
26.4.3 SPI Control Register (SPI_CR).....	617
26.4.4 SPI Config Register 1 (SPI_CFG1).....	619
26.4.5 SPI Status Register (SPI_SR).....	620
26.4.6 SPI Config Register 2 (SPI_CFG2).....	622
27 Controller Area Network (MCAN).....	623
27.1 Introduction.....	623
27.2 Features.....	623
27.3 Functional Description.....	624
27.3.1 MCAN System Block Diagram.....	624
27.3.2 Pin Description.....	625
27.3.3 Working Mode.....	625
27.3.4 Timestamp Generation.....	633
27.3.5 Timeout Counter.....	634
27.3.6 Reception Processing.....	634
27.3.7 Transmission Processing.....	641
27.3.8 FIFO Confirmation Processing.....	645
27.3.9 Message RAM.....	646
27.4 Register Description.....	658
27.4.1 Register List.....	658
27.4.2 MCAN Endian Register (MCAN_ENDN).....	661
27.4.3 MCAN Data Bit Timing and Prescaler Register (MCAN_DBTP).....	662
27.4.4 MCAN Test Register (MCAN_TEST).....	664
27.4.5 MCAN RAM Watchdog Register (MCAN_RWD).....	666
27.4.6 MCAN CC Control Register (MCAN_CCCR).....	667
27.4.7 MCAN Nominal Bit Timing and Prescaler Register (MCAN_NBTP).....	670
27.4.8 MCAN Timestamp Counter Configuration Register (MCAN_TSCC).....	671
27.4.9 MCAN Timestamp Counter Value Register (MCAN_TSCV).....	672
27.4.10 MCAN Timeout Counter Configuration Register (MCAN_TOCC).....	673
27.4.11 MCAN Timeout Counter Value Register (MCAN_TOCV).....	674
27.4.12 MCAN Error Counter Register (MCAN_ECR).....	675
27.4.13 MCAN Protocol Status Register (MCAN_PSR).....	676
27.4.14 MCAN Transmit Delay Compensation Register (MCAN_TDCR).....	680
27.4.15 MCAN Interrupt Register (MCAN_IR).....	681
27.4.16 MCAN Interrupt Enable Register (MCAN_IE).....	685
27.4.17 MCAN Interrupt Line Select Register (MCAN_ILS).....	688
27.4.18 MCAN Interrupt Line Enable Register (MCAN_ILE).....	691

27.4.19 MCAN Global Filter Configuration Register (MCAN_GFC).....	692
27.4.20 MCAN Standard ID Filter Configuration Register (MCAN_SIDFC).....	693
27.4.21 MCAN Extended ID Filter Configuration Register (MCAN_XIDFC).....	694
27.4.22 MCAN Extended ID and Mask Register (MCAN_XIDAM).....	695
27.4.23 MCAN High Priority Message Status Register (MCAN_HPMS).....	696
27.4.24 MCAN New Data 1 Register (MCAN_NDAT1).....	697
27.4.25 MCAN New Data 2 Register (MCAN_NDAT2).....	698
27.4.26 MCAN Receive FIFO0 Configuration Register (MCAN_RXF0C).....	699
27.4.27 MCAN Receive FIFO0 Status Register (MCAN_RXF0S).....	700
27.4.28 MCAN Receive FIFO0 Acknowledge Register (MCAN_RXF0A).....	701
27.4.29 MCAN Receive Buffer Configuration Register (MCAN_RXBC).....	701
27.4.30 MCAN Receive FIFO1 Configuration Register (MCAN_RXF1C).....	702
27.4.31 MCAN Receive FIFO1 Status Register (MCAN_RXF1S).....	703
27.4.32 MCAN Receive FIFO1 Acknowledge Register (MCAN_RXF1A).....	704
27.4.33 MCAN Receive Buffer and FIFO Element Size Configuration Registers (MCAN_RXESC).....	705
27.4.34 MCAN Transmit Buffer Configuration Register (MCAN_TXBC).....	706
27.4.35 MCAN Transmit FIFO/Queue Status Register (MCAN_TXFQS).....	707
27.4.36 MCAN Transmit Buffer Element Size Configuration Register (MCAN_TXESC).....	708
27.4.37 MCAN Transmit Buffer Request Pending Register (MCAN_TXBRP).....	709
27.4.38 MCAN Transmit Buffer Add Request Register (MCAN_TXBAR).....	710
27.4.39 MCAN Transmit Buffer Cancellation Request Register (MCAN_TXBCR).....	711
27.4.40 MCAN Transmit Buffer Transmission Occurred Register (MCAN_TXBTO).....	712
27.4.41 MCAN Transmit Buffer Cancellation Finished Register (MCAN_TXBCF).....	713
27.4.42 MCAN Transmit Buffer Transmission Interrupt Enable Register (MCAN_TXBTIE).....	714
27.4.43 MCAN Transmit Buffer Cancellation Finished Interrupt Enable Register (MCAN_TXBCIE).....	714
27.4.44 MCAN Transmit Event FIFO Configuration Register (MCAN_TXEFC).....	715
27.4.45 MCAN Transmit Event FIFO Status Register (MCAN_TXEFS).....	716
27.4.46 MCAN Transmit Event FIFO Acknowledge Register (MCAN_TXEFA).....	717
27.4.47 MCAN Message RAM Check Control Register (MCANRAM_CKCR).....	718
27.4.48 MCAN Message RAM Check Status Register (MCANRAM_CKSR).....	719
27.4.49 MCAN Message RAM Error Injection Enable Register (MCANRAM_EIEN).....	719
27.4.50 MCAN Message RAM Error Injection Bit Enable Register 0 (MCANRAM_EIBIT0).....	720
27.4.51 MCAN Message RAM Error Injection Bit Enable Register 1 (MCANRAM_EIBIT1).....	720
27.4.52 MCAN Message RAM Check Error Address Register (MCANRAM_ECCERRADDR).....	721
27.5 Precautions.....	721
27.5.1 CAN Bus Anti-Interference Measures.....	721
27.5.2 CAN Controller Noise Constraints.....	721
28 CRC Operation (CRC).....	722
28.1 Introduction.....	722
28.2 Features.....	722
28.3 Functional Description.....	722
28.3.1 Working Mode.....	722
28.3.2 Encoding Method.....	722
28.3.3 Write Bit Width.....	723
28.4 Operation Example.....	723

28.4.1 CRC-16 Encoding Mode.....	723
28.4.2 CRC-16 Check Mode.....	723
28.4.3 CRC-32 Encoding Mode.....	723
28.4.4 CRC-32 Check Mode.....	723
28.5 Register Description.....	724
28.5.1 Register List.....	724
28.5.2 Control Register (CRC_CR).....	725
28.5.3 Result Register (CRC_RESLT).....	725
28.5.4 Data Register (CRC_DAT).....	726
29 Debug Controller (DBGC).....	727
29.1 Introduction.....	727
29.2 Functional Description.....	727
29.2.1 Function Block Diagram.....	727
29.2.2 Reference Documentation.....	727
29.2.3 Debug Port.....	728
29.2.4 Pin Arrangement and Debug Port Pins.....	728
29.3 Register Description.....	729
29.3.1 Register List.....	729
29.3.2 Chip Identification Register (CHIPID).....	730
29.3.3 Target Identification Register (TARGETID).....	730
29.3.4 DBG Status Register (MCUDBGSTAT).....	731
29.3.5 Peripheral Debug Stop Control Register (MCUSTPCTL).....	732
A Glossary.....	734
Revision History.....	735

List of Figures

Figure 3-1 Bus Architecture Diagram.....	33
Figure 4-1 Power-on Reset Timing Diagram.....	36
Figure 4-2 NRST Reset Timing Diagram.....	36
Figure 4-3 Low Voltage Detection 1 Reset Timing Diagram.....	37
Figure 4-4 Low Voltage Detection 2 Reset Timing Diagram.....	37
Figure 4-5 Special Watchdog Reset Timing Diagram.....	38
Figure 4-6 Software Reset Timing Diagram.....	38
Figure 4-7 RAM ECC Reset Timing Diagram.....	38
Figure 4-8 FLASH ECC Reset Timing Diagram.....	39
Figure 4-9 Clock Frequency Error Reset Timing Diagram.....	39
Figure 4-10 XTAL Error Reset Timing Diagram.....	39
Figure 4-11 Cortex-M0+ Lockup Reset Timing Diagram.....	40
Figure 5-1 Clock System Block Diagram.....	45
Figure 5-2 Clock Frequency Measurement Block Diagram.....	45
Figure 5-3 External High Speed Oscillator Connection Example.....	47
Figure 5-4 External Clock Input Connection Example.....	47
Figure 5-5 External High-speed Oscillator Fault Detection Example.....	48
Figure 5-6 System Clock Selected as XTAL, Example of Detecting XTAL Oscillation Fault.....	48
Figure 5-7 External Low-speed Oscillator Connection Example.....	49
Figure 5-8 Clock Source Switching Flowchart.....	51
Figure 5-9 Clock Frequency Measurement Timing Diagram.....	52
Figure 6-1 Power Supply Configuration Diagram.....	80
Figure 6-2 Power-on Reset, Power-off Reset Waveforms.....	81
Figure 6-3 LVD1 Non-maskable Interrupt/Reset Block Diagram.....	82
Figure 6-4 LVD1 Non-maskable Interrupt Timing Diagram.....	82
Figure 6-5 LVD1 Reset Timing Diagram.....	83
Figure 6-6 LVD2 Non-maskable Interrupt/Reset Block Diagram.....	84
Figure 6-7 LVD2 Non-maskable Interrupt Timing Diagram.....	85
Figure 6-8 LVD2 Reset Timing Diagram.....	85
Figure 8-1 Flash Address Distribution.....	114
Figure 8-2 Address Transformation After Boot Swap.....	123
Figure 8-3 Boot Swap Operation Flow.....	124
Figure 9-1 Error Injection Diagram for SRAM 39-bit.....	154
Figure 10-1 Port Basic Structure Diagram.....	163
Figure 11-1 Interrupt System Block Diagram.....	196
Figure 11-2 Interrupt Event Selection.....	197
Figure 11-3 Digital Filter Operation Schematic Diagram.....	199
Figure 12-1 AOS Module Block Diagram.....	215
Figure 13-1 CTC Function Block Diagram.....	227
Figure 13-2 CTC Calibration Diagram.....	235
Figure 14-1 DMA Architecture Diagram.....	246
Figure 14-2 Chain Transmission Diagram.....	249
Figure 14-3 Schematic Diagram of NonSequence Address Transfer (Taking source address as an example).....	250
Figure 14-4 Application Example 1: Memory-to-Memory Transfer.....	252
Figure 14-5 Application Example 2: Memory-to-Peripheral Transfer.....	253
Figure 15-1 CMP Function Block Diagram.....	271
Figure 15-2 Voltage Comparison Diagram.....	273
Figure 15-3 Window Output Diagram I.....	274
Figure 15-4 Window Output Diagram II.....	274
Figure 15-5 Window Output Diagram III.....	275

Figure 16-1	ADC Block Diagram.....	290
Figure 16-2	Sequence A Single Scan Mode.....	291
Figure 16-3	Continuous Scan.....	292
Figure 16-4	Dual Sequence Scan Mode (Sequence A Restarts from Interrupted Channel).....	294
Figure 16-5	Dual Sequence Scan Mode (Sequence A Restarts from First Channel).....	294
Figure 16-6	Analog Watchdog Protected Area (Comparative Mode).....	295
Figure 16-7	ADC Conversion Time Diagram.....	296
Figure 16-8	ADC Interrupt and Event Output Timing.....	297
Figure 17-1	Timer4 Basic Block Diagram.....	320
Figure 17-2	Timer4 Sawtooth Waveform.....	321
Figure 17-3	Timer4 Triangle Waveform.....	321
Figure 17-4	Timer4 Sawtooth Wave Mode Counting Action.....	321
Figure 17-5	Timer4 Triangle Wave Mode Counting Action.....	322
Figure 17-6	Sawtooth Mode Waveform Output Example.....	323
Figure 17-7	Triangular Wave Mode Waveform Output Example.....	324
Figure 17-8	Modify Sawtooth Count Period when the Buffer is Disabled.....	325
Figure 17-9	Modify Sawtooth Wave Count Period when the Buffer is Enabled.....	325
Figure 17-10	Modify Triangular Wave Count Period when Buffer is Enabled.....	325
Figure 17-11	OCCR Buffer Data Transmission (Period Interval Response Link Disabled).....	326
Figure 17-12	OCCR Buffer Data Transmission (Period Interval Response Link Enabled).....	327
Figure 17-13	Output Compare Buffer Data Transfer (OCMR Buffer Enabled).....	328
Figure 17-14	SCCR Buffer Transmission Operation (Period Interval Response Link Transmission Disabled).....	328
Figure 17-15	SCCR Buffer Transmission Operation (when Periodic Interval Response Link Transmission Enabled).....	329
Figure 17-16	Sawtooth Independent PWM Output Example.....	330
Figure 17-17	Triangle Independent PWM Output Example.....	331
Figure 17-18	Sawtooth Wave Extended PWM Output.....	332
Figure 17-19	Software Implementation of Complementary PWM Output.....	332
Figure 17-20	Complementary PWM Output in Dead-Zone Generator Mode.....	333
Figure 17-21	Waveform Output in Dead-Zone Timer Mode with Pulse Width Anomalies.....	334
Figure 17-22	Complementary PWM Output in Dead-Zone Timer Filter Mode.....	335
Figure 17-23	Periodic Interval Response Timing Diagram.....	336
Figure 17-24	Special Event Output Signal Periodic Interval Response Output.....	337
Figure 17-25	PWM Period Output Monitoring Example.....	338
Figure 17-26	Output Timing of Special Event Output Signal in Delayed Start Mode.....	340
Figure 18-1	EMB Channel Function Block Diagram.....	380
Figure 18-2	EMB Structure Block Diagram.....	381
Figure 19-1	TimerA Basic Block Diagram.....	396
Figure 19-2	Sawtooth Waveform (Counting Up).....	397
Figure 19-3	Triangle Waveform.....	397
Figure 19-4	Compare Output Action.....	397
Figure 19-5	Capture Input Action.....	398
Figure 19-6	Filtering Function of Clock Input Port.....	399
Figure 19-7	Buffer Action in Sawtooth Mode.....	401
Figure 19-8	Example of Single-Side Aligned PWM Output.....	401
Figure 19-9	Example of Double-Side Symmetric PWM Output.....	402
Figure 19-10	Position Mode-Basic Counting.....	403
Figure 19-11	Position Counting Mode-Phase Difference Counting (1x Counting).....	403
Figure 19-12	Position Counting Mode-Phase Difference Counting (2x Counting).....	403
Figure 19-13	Position Counting Mode-Phase Difference Counting (4x Counting).....	404
Figure 19-14	Position Counting Mode-Direction Counting.....	404
Figure 20-1	Timer2 Basic Block Diagram.....	435
Figure 20-2	Compare Output Action.....	437

Figure 20-3 Hardware Start and Clear Action.....	438
Figure 20-4 Capture Input Action.....	438
Figure 20-5 Pulse Width Measurement.....	439
Figure 20-6 Period Measurement.....	439
Figure 20-7 Digital Filtering of TRIG Input Port.....	440
Figure 21-1 Timer0 Basic Block Diagram.....	458
Figure 21-2 Timer0 Counting Timing Diagram.....	459
Figure 22-1 RTC Basic Block Diagram.....	468
Figure 23-1 Hardware Startup Mode Example.....	490
Figure 23-2 Software Startup Mode Example.....	491
Figure 23-3 Examples of Various Refresh Action Sequences (action confirmation, refresh required signal falling edge, etc.).....	491
Figure 23-4 Counter Underflow Operation Example.....	493
Figure 23-5 Counter Refresh Operation Example.....	494
Figure 24-1 USART System Block Diagram.....	501
Figure 24-2 UART Data Format.....	503
Figure 24-3 UART Transmit Data Diagram 1.....	504
Figure 24-4 UART transmit Data Diagram 2.....	504
Figure 24-5 UART Internal Synchronization and Sample Timing.....	505
Figure 24-6 UART Receive Data Diagram 1.....	507
Figure 24-7 UART Receive Data Diagram 2.....	507
Figure 24-8 Half-Duplex Operation Timing.....	509
Figure 24-9 OVER8=1 DE Asserted.....	510
Figure 24-10 OVER8=0 DE Asserted.....	510
Figure 24-11 OVER8=1 DE De-assertion.....	510
Figure 24-12 OVER8=0 DE De-assertion.....	511
Figure 24-13 Multiprocessor Communication Legend.....	512
Figure 24-14 Multiprocessor Mode Data Format.....	512
Figure 24-15 Multiprocessor Mode Transmit Data Diagram.....	513
Figure 24-16 Multiprocessor Mode Receive Data Diagram 1.....	514
Figure 24-17 Multiprocessor Mode Receive Data Diagram 2.....	514
Figure 24-18 LIN Bus Data Behavior.....	515
Figure 24-19 Smart Card Connection Schematic Diagram.....	518
Figure 24-20 Smart Card Mode Synchronization Timing and Sample Timing Diagram.....	518
Figure 24-21 Smart Card Mode Data Format.....	519
Figure 24-22 Smart Card Mode Transmit Data Diagram.....	520
Figure 24-23 Smart Card Mode Receive Data Diagram.....	520
Figure 24-24 Clock Synchronization Mode Data Format.....	522
Figure 24-25 Data Transmission in Clock Synchronization Mode Diagram 1 (LSB/without CTS function).....	523
Figure 24-26 Data Transmission in Clock Synchronization Mode Diagram 2 (LSB/with CTS function).....	523
Figure 24-27 Data Reception in Clock Synchronization Mode Diagram 1 (LSB/without RTS function).....	524
Figure 24-28 Data Reception in Clock Synchronization Mode Diagram 2 (LSB/with RTS function).....	524
Figure 24-29 Digital Filter Function Diagram.....	526
Figure 25-1 I2C System Block Diagram.....	555
Figure 25-2 Example of I2C Bus Structure.....	555
Figure 25-3 I2C Bus Timing Diagram.....	556
Figure 25-4 Data Format of I2C Bus.....	557
Figure 25-5 Sequence Diagram of Master Transmitting Data in 7-bit Address Format (Example).....	558
Figure 25-6 Sequence Diagram of Master Receiving Data in 7-bit Address Format (Example).....	559
Figure 25-7 Sequence Diagram of Slave Transmitting Mode in 7-bit Address Format (Example).....	559
Figure 25-8 Sequence Diagram of Slave Receiving Data in 7-bit Address Format (Example).....	560
Figure 25-9 SCL Synchronous Timing.....	561

Figure 25-10	Transmitting Timing Diagram of Slave (1).....	562
Figure 25-11	Transmitting Timing Diagram of Slave (2).....	562
Figure 25-12	Slave Receiving Timing Diagram.....	563
Figure 25-13	Fast ACK/NACK Timing Diagram.....	563
Figure 25-14	Timing when Selecting 7-bit Address Format.....	564
Figure 25-15	Timing when Selecting 10-bit Address Format.....	564
Figure 25-16	Broadcast Address Matching Timing Diagram.....	565
Figure 25-17	SMBus Master Address Matching Timing Diagram.....	565
Figure 25-18	SMBus Alarm Response Address Matching Timing Diagram.....	566
Figure 25-19	SMBus Default Address Timing Diagram.....	566
Figure 25-20	Digital Filtering Circuit Block Diagram.....	568
Figure 26-1	SPI System Block Diagram.....	595
Figure 26-2	Master Mode Structure.....	597
Figure 26-3	Three-wire Clock Synchronization Operation.....	598
Figure 26-4	Data Format.....	599
Figure 26-5	MSB Transmitted First, Parity Invalid.....	599
Figure 26-6	LSB Transmitted First, Parity Invalid.....	600
Figure 26-7	MSB Transmitted First, Parity Valid.....	600
Figure 26-8	LSB Transmitted First, Parity Valid.....	601
Figure 26-9	Data Transmission Format Diagram (CPHA=0).....	601
Figure 26-10	Data Transmission Format Diagram (CPHA=1).....	602
Figure 26-11	Full-duplex Synchronous Serial Communication.....	602
Figure 26-12	Transmit-only Communication.....	603
Figure 26-13	Normal Communication.....	603
Figure 26-14	Continuous Communication.....	604
Figure 26-15	Parity Check Process.....	610
Figure 26-16	Overload Error Pocessing.....	612
Figure 26-17	Parity Error.....	613
Figure 27-1	MCAN System Block Diagram.....	624
Figure 27-2	Transceiver Delay Measurement.....	629
Figure 27-3	Pin Control in Bus Monitoring Mode.....	631
Figure 27-4	External Loopback Mode Pin Control.....	633
Figure 27-5	Pin Control for Internal Loopback Mode.....	633
Figure 27-6	Standard Message ID Filter Path.....	636
Figure 27-7	Extended Message ID Filter Path.....	637
Figure 27-8	Receive FIFO Status.....	638
Figure 27-9	Receive FIFO Overflow Processing.....	639
Figure 27-10	Example of Mixed Configuration of Dedicated Transmit Buffers and Transmit FIFO.....	644
Figure 27-11	Example of Mixed Configuration of Dedicated Transmit Buffers and Transmit Queue.....	644
Figure 27-12	Message RAM Configuration.....	647
Figure 29-1	Debug Control System Block Diagram.....	727

List of Tables

Table 2-1	Memory Mapping.....	28
Table 4-1	Reset Mode and Generating Conditions.....	34
Table 4-2	Reset Mode and Reset Flag.....	35
Table 4-3	Reset Module Unit (RMU) Register Offset.....	41
Table 5-1	Clock Source Main Features.....	44
Table 5-2	Specifications for Each Internal Clock.....	46
Table 5-3	Clock Frequency Measurement (FCM) Register Offset.....	53
Table 5-4	Clock Module Unit (CMU) Register Offset.....	53
Table 5-5	Clock Module Unit (CMU) Register Offset.....	54
Table 6-1	LVD1 Features.....	81
Table 6-2	LVD1 Voltage Selection.....	82
Table 6-3	LVD2 Features.....	83
Table 6-4	LVD2 Voltage Selection.....	83
Table 6-6	Operating Conditions of Low Power Mode and Status of Each Module in Low Power Mode.....	86
Table 6-7	Register Protection List.....	89
Table 6-8	Power Control (PWC) Register Offset.....	89
Table 6-9	Power Control (PWC) Register Offset.....	89
Table 7-1	ICGA Register Offset Address.....	101
Table 7-2	ICGB Register Offset.....	103
Table 8-1	Flash Read Mode and Maximum Read Frequency Comparison Table.....	115
Table 8-2	OTPA Address Distribution Table.....	122
Table 8-3	OTPB Address Distribution Table.....	122
Table 8-4	EFM Module Interrupt Source and Reset Source Comparison Table.....	127
Table 8-5	ProgramFLASH Register Offset.....	129
Table 8-6	DataFLASH and EDataFLASH Register Offset.....	129
Table 9-1	SRAM Space Allocation.....	153
Table 9-2	Built-in SRAM (SRAM) Register Offset.....	155
Table 10-1	General-Purpose IO (GPIO) Register Offset.....	167
Table 10-2	Event Port Register Offset.....	167
Table 11-1	External Interrupt Input Pin.....	190
Table 11-2	Interrupt Vector Table.....	191
Table 11-4	Interrupt Controller (INTC) Register Offset.....	200
Table 12-1	AOS Target List.....	215
Table 12-2	Automatic Operation System (AOS) Register Offset.....	217
Table 13-1	Measurement Error for HRC Target Frequency of 64MHz.....	229
Table 13-2	Measurement Error for HRC Target Frequency of 48MHz.....	232
Table 13-3	Clock Trimming Controller (CTC) Register Offset.....	237
Table 14-1	DMA Controller (DMA) Register Offset.....	255
Table 15-1	CMP Pin List.....	271
Table 15-2	CMP Analog Input List.....	272
Table 15-3	Voltage Comparator (CMP) Register Offset.....	277
Table 15-20	Timer Window PWM List.....	286
Table 16-1	ADC Input Signal Description.....	290
Table 16-2	Various Conflicts Between Sequences A and B.....	293
Table 16-3	ADC Conversion Time.....	296
Table 16-4	Analog-to-Digital Converter (ADC) Register Offset.....	298
Table 17-1	Basic Functions and Features of Timer4.....	319
Table 17-2	Timer4 Port List.....	320
Table 17-3	Internal Signal of General Comparator Output.....	322
Table 17-4	Corresponding Relationship Between Internal Signal of General Comparator Output and Port Output Control....	330

Table 17-5	General Control Timer (Timer4) Register Offset.....	341
Table 17-21	OCCR*h Count Match List.....	355
Table 17-24	OCCR*h/l Count Match List.....	358
Table 18-1	System Error Sources.....	381
Table 18-2	EMB Port Assignment.....	381
Table 18-4	Emergency Brake Module (EMB) Register Offset.....	384
Table 19-1	Basic Functions and Features of TimerA.....	395
Table 19-2	TimerA Port List.....	396
Table 19-3	TRGSEL Selection Relationship Table for Internal Trigger Events.....	398
Table 19-4	General Timer (TimerA) Register Offset.....	406
Table 19-5	TMR_SYNENR Register Offset.....	406
Table 20-1	Basic Functions and Features of Timer2.....	434
Table 20-2	Timer2 Port List.....	435
Table 20-3	General Timer (Timer2) Register Offset.....	441
Table 21-1	General Timer (Timer0) Register Offset.....	461
Table 22-1	RTC Main Features.....	467
Table 22-2	Real Time Clock (RTC) Register Offset.....	472
Table 23-1	Basic Features of Watchdog Timer.....	489
Table 23-2	Watchdog Timer (SWDT) Register Offset.....	495
Table 24-2	USART pin description.....	501
Table 24-3	Tolerance of UART Receiver.....	506
Table 24-4	UART Interrupt/Event Table.....	511
Table 24-5	Multiprocessor Mode Interrupt/Event Table.....	514
Table 24-6	LIN Interrupt/Event Table.....	517
Table 24-7	Smart Card Mode Interrupt/Event Table.....	521
Table 24-8	Interrupt/Event Table in Synchronous Clock Mode.....	525
Table 24-9	USART Global Interrupt Summary Table.....	527
Table 24-10	Universal Synchronous Asynchronous Receiver/Transmitter (USART) Register Offset.....	528
Table 24-19	Baud Rate Calculation Formula (Fractional Baud Rate is disable where FBME=0).....	535
Table 24-20	Baud Rate Calculation Formula (Fractional Baud Rate is Enable where FBME=1).....	535
Table 25-1	Input/Output Pin.....	556
Table 25-2	Interrupt List.....	568
Table 25-3	Event Signal Output List.....	569
Table 25-4	Inter-Integrated Circuit (I2C) Register Offset.....	571
Table 26-1	SPI Main Features.....	594
Table 26-2	Pin Description.....	595
Table 26-3	SPI Pin Status Description in Master Mode.....	596
Table 26-4	SPI Pin Status Description in Slave Mode.....	596
Table 26-5	Bit Speed of Some Set Values.....	598
Table 26-6	SPI Mode and Register Setting Relationship.....	604
Table 26-7	Error Detection Correspondence Table.....	610
Table 26-8	SPI Interrupt Source and Trigger Source Description.....	614
Table 26-9	Serial Peripheral Interface (SPI) Register Offset.....	615
Table 27-1	MCAN Pin Description.....	625
Table 27-2	MCAN Configuration Register List.....	626
Table 27-3	DLC Encoding in MCAN.....	628
Table 27-4	Receive buffer and FIFO element size.....	638
Table 27-5	Receive Buffer Filter Configuration Example.....	640
Table 27-6	Debug Message Filter Configuration Example.....	641
Table 27-7	Possible configurations of the send frame.....	641
Table 27-8	Transmit Buffer, FIFO (queue) Element Size.....	642
Table 27-9	Receive Buffers and FIFO Elements.....	648

Table 27-10	Receive Buffer and FIFO Elements Description.....	648
Table 27-11	Transmit Buffer Elements.....	650
Table 27-12	Transmit Buffer Elements Description.....	651
Table 27-13	Transmit Event FIFO Elements.....	653
Table 27-14	Transmit Event FIFO Element Description.....	653
Table 27-15	Standard Message ID Filter Element.....	655
Table 27-16	Standard Message ID Filter Element Description.....	655
Table 27-17	Extended Message ID Filter Element.....	656
Table 27-18	Extended Message ID Filter Element Description.....	656
Table 27-19	Controller Area Network (MCAN) Register Offset.....	658
Table 27-20	MCANRAM register offset.....	660
Table 28-1	CRC Operation (CRC) Register Offset.....	724
Table 29-1	Debug Port Pins.....	728
Table 29-2	Debug Controller (DBGC) Register Offset.....	729

1 Product Introduction

HC32K11x Series is a high-performance MCU with maximum operating 64MHz frequency based on 32-bit ARM[®] Cortex[®]-M0+ Microcontroller. The Cortex-M0+ core supports a 2-stage pipeline with 0.95 Dhrystone MIPS/MHz performance. The hardware debugging circuit supports a 2-pin SWD debugging interface.

HC32K11x Series integrates on-chip memory, including 256KB ProgramFLASH, 32KB DataFLASH, 2KB EDataFLASH, and 32KB SRAM. Includes Flash prefetch acceleration to improve access efficiency. Supporting DMA and the mutual triggering of events among peripheral devices to reduce the transaction processing load of the CPU.

HC32K11x Series integrates rich peripheral functions. Includes 1 maximum 16-channel 12-bit 1Msps ADC, 1 independent voltage comparator (CMP) with built-in 8-bit DAC, 1 motor Timer (Timer4), 1 multi-functional 16-bit Timer (TimerA) supporting quadrature encoder input, input capture, output capture, and PWM output, 1 32-bit Timer (Timer2), 1 16-bit Timer (Timer2) supporting asynchronous counting, 1 16-bit Timer (Timer0), 1 I2C communication interface, 2 SPI communication interfaces, 4 USART communication interfaces (2 of which support LIN communication functionality), and 1 CAN FD controller (MCAN).

HC32K11x Series supports wide voltage (2.7-5.5V), temperature (-40°C-125°C), and low-power modes.

Typical Application

HC32K11x Series is available in 64-pin/48-pin LQFP and 32-pin QFN package. Suitable for motor control and full-width taillight applications.

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 ⁽¹⁾		Start Address	End Address	Space Size	Module ⁽²⁾	Protection ⁽³⁾	Description
System	-	0xE0100000	0xFFFFFFFF	511MB	Reserved	-	Custom space
	Private Peripheral External Bus	0xE00FF000	0xE00FFFFF	4KB	ROMTABLE	-	-
		0xE0042400	0xE00FEFFF	755KB	-	-	-
		0xE0042000	0xE00423FF	1KB	-	-	-
		0xE0041000	0xE0041FFF	4KB	-	-	-
		0xE0040000	0xE0040FFF	4KB	-	-	-
	Private Peripheral Internal Bus	0xE000F000	0xE003FFFF	196KB	-	-	-
		0xE000E000	0xE000EFFF	4KB	SCS	-	System control space NVIC/MPU, etc.
		0xE0003000	0xE000DFFF	44KB	-	-	-
		0xE0002000	0xE0002FFF	4KB	FPB	-	-
		0xE0001000	0xE0001FFF	4KB	DWT	-	-
		0xE0000000	0xE0000FFF	4KB	-	-	-
External Peripheral	-	0xA0000000	0xDFFFFFFF	1024MB	Reserved	-	-
External Storage	-	0x60000000	0x9FFFFFFF	1024MB	Reserved	-	-
Peripheral	-	0x44000000	0x5FFFFFFF	448MB	Reserved	-	-
		0x42000000	0x43FFFFFF	32MB	PeriBitBand	-	Non-CPU host reserved
		0x40100000	0x41FFFFFF	31MB	Reserved	-	-
		0x40020000	0x400FFFFF	896KB	BLANK	-	-
	AHB	0x40018000	0x4001FFFF	32KB	DataFLASH	-	-
		0x40017C00	0x40017FFF	1KB	Flash_TEST	-	-
		0x40017A00	0x40017BFF	0.5KB	OTPB	-	-
		0x40017800	0x400179FF	0.5KB	ICGB	-	-
		0x40017000	0x400177FF	2KB	EDataFLASH	-	-
		0x40015800	0x40016FFF	6KB	BLANK	-	-
		0x40015400	0x400157FF	1KB	CRC	-	-
		0x40015000	0x400153FF	1KB	DBGCC	-	-
		0x40014800	0x40014FFF	1KB	BLANK	-	-
		0x40014400	0x400147FF	1KB	CMU	-	-
		0x40014000	0x400143FF	1KB	EMU	-	-
		0x40013800	0x40013FFF	2KB	GPIO	-	Start 1K shared with FASTIO
		0x40013400	0x400137FF	1KB	PERIC	-	Peripheral control register
		0x40013000	0x400133FF	1KB	DMA	-	-
		0x40011C00	0x40012FFF	5KB	BLANK	-	-

Memory Classification ⁽¹⁾		Start Address	End Address	Space Size	Module ⁽²⁾	Protection ⁽³⁾	Description
Peripheral	AHB	0x40011800	0x40011BFF	1KB	MCAN-RAM_ECC	-	-
		0x40011400	0x400117FF	2KB	MCAN	-	-
		0x40011000	0x400113FF	1KB	INTC	-	-
		0x40010C00	0x40010FFF	1KB	RAMIF	-	-
		0x40010000	0x40010BFF	3KB	BLANK	-	-
	APB	0x4000D800	0x4000FFFF	10KB	BLANK	-	-
		0x4000D400	0x4000D7FF	1KB	RTC	-	-
		0x4000D000	0x4000D3FF	1KB	BLANK	-	-
		0x4000CC00	0x4000CFFF	1KB	SWDT	-	-
		0x4000C800	0x4000CBFF	1KB	CMP	-	-
		0x4000BC00	0x4000C7FF	3KB	BLANK	-	-
		0x4000B800	0x4000BBFF	1KB	ADC	-	-
		0x40007400	0x4000B7FF	17KB	BLANK	-	-
		0x40007000	0x400073FF	1KB	TimerA	-	-
		0x40006C00	0x40006FFF	1KB	EMB	-	-
		0x40006800	0x40006BFF	1KB	Timer4	-	-
		0x40006400	0x400067FF	1KB	BLANK	-	-
		0x40006000	0x400063FF	1KB	Timer2_2	-	-
		0x40005C00	0x40005FFF	1KB	Timer2_1	-	-
		0x40005800	0x40005BFF	1KB	Timer0	-	-
		0x40005400	0x400057FF	1KB	FCM	-	-
		0x40004C00	0x400053FF	2KB	BLANK	-	-
		0x40004800	0x40004BFF	1KB	I2C	-	-
		0x40004000	0x400047FF	2KB	BLANK	-	-
		0x40003C00	0x40003FFF	1KB	SPI_2	-	-
		0x40003800	0x40003BFF	1KB	SPI_1	-	-
		0x40002800	0x400037FF	4KB	BLANK	-	-
		0x40002400	0x400027FF	1KB	USART_4	-	-
		0x40002000	0x400023FF	1KB	USART_3	-	-
		0x40001C00	0x40001FFF	1KB	USART_2	-	-
		0x40001800	0x40001BFF	1KB	USART_1	-	-
		0x40001000	0x400017FF	2KB	BLANK	-	-
		0x40000C00	0x40000FFF	1KB	AOS	-	-
		0x40000400	0x40000BFF	2KB	EFM	-	-
		0x40000000	0x400003FF	1KB	CTC	-	-
SRAM	SRAM	0x24000000	0x3FFFFFFF	448MB	Reserved	-	-
		0x22000000	0x23FFFFFF	32MB	SRAMBit-Band	-	Non-CPU host reserved
		0x20100000	0x21FFFFFF	31MB	Reserved	-	-
		0x20008400	0x200FFFFF	991KB	Reserved	-	-
		0x20008000	0x200083FF	1KB	MCAN_RAM	-	-
		0x20000000	0x20007FFF	32KB	SRAM	Protection	-

Memory Classification ⁽¹⁾		Start Address	End Address	Space Size	Module ⁽²⁾	Protection ⁽³⁾	Description
CODE	FLASH	0x01100000	0x1FFFFFFF	495MB	Reserved	-	-
		0x01002000	0x010FFFFF	1016KB	Reserved	-	-
		0x01001200	0x01001FFF	3.5KB	Flash_TEST	-	-
		0x01001040	0x010011FF	448B	BLANK	-	-
		0x01001000	0x0100103F	64B	Data security protection	-	Used for configuration of data security protection
		0x01000E04	0x01000FFF	508B	BLANK	-	-
		0x01000E00	0x01000E03	4B	Boot swap	-	-
		0x01000C00	0x01000DFF	0.5KB	OTPA	-	-
		0x01000A00	0x01000BFF	0.5KB	ICGA	-	-
		0x01000000	0x010009FF	2.5KB	BLANK	-	-
		0x00040000	0x00FFFFFF	15.75MB	Reserved	-	-
		0x00000000	0x0003FFFF	256KB	Program-FLASH	-	-



Note

1. Please refer to the ARM Cortex-M0+ manual for the memory system.
2. Reserved: Accessing the bus will cause a bus error.

BLANK: Write access has no effect; reading from the FLASH area returns 1, and reading from non-FLASH areas returns 0.

Flash_TEST: A pre-configured sector for factory testing, which cannot be modified by users.

3. For protection functions, please refer to the relevant descriptions in the "Power Control (PWC) " chapter.

2.2 Bit Segment Space

Cortex-M0+ 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. Writing a word to the alias area is equivalent to a read-modify-write operation on the target bit of the bit segment 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-M0+ access and are invalid for other bus master interfaces (e.g., DMA).

In the bit segment region, each word is represented by the LSB of 32 words in the bit segment alias address region. In practice, when accessing a bit segment alias address, the address is re-mapped to the bit segment address. For read operations, the word is read, and the selected bit's position is shifted to the LSB of the returned read data. For write operations, the bit data to be written is shifted to the required position, followed by a read-modify-write operation.

There are two memory regions where bit segment operations can be performed:

- 0x20000000-0x200FFFFF (SRAM, 1MB)
- 0x40000000-0x400FFFFF (Peripherals, 1MB)

The example is as follows:

1. Set address 0x20000000 to 0x3355AACC.
2. Read address 0x22000008. This read access is mapped to a read access to 0x20000000, returning a value of 1 (Bit[2] of 0x3355AACC).

3. Write 0 to 0x22000008. This write access is re-mapped to a read-modify-write to address 0x20000000. The value 0x3355AACC is read from the register, and after clearing Bit[2], the result 0x3355AAC8 is written to address 0x20000000.
4. Now read 0x20000000, which will return the value 0x3355AAC8 (Bit[2] is cleared).
5. When accessing a bit segment alias address, only the LSB (Bit[0]) of the data is used. Additionally, access to the bit segment alias region should not be unaligned. If unaligned access is performed within the bit segment region, the result is unpredictable.

3 Bus Architecture (BUS)

3.1 Introduction

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

- Host bus
 - ▶ Cortex-M0+ core CPU bus
The CPU fetches instructions and data via this bus, with access targets including internal memory and peripherals.
 - ▶ System DMA bus
The system DMA has a dedicated bus through which it accesses memory and peripherals.
- Slave bus
 - ▶ FLASH bus
 - ▶ SRAM bus
 - ▶ Peripheral bus

With the bus matrix, efficient concurrent access from host buses to slave buses can be achieved. The bus matrix employs a fixed-priority arbitration scheme, where DMA access has higher priority than CPU access, to ensure the continuity of DMA data transfers.

3.2 Bus Architecture

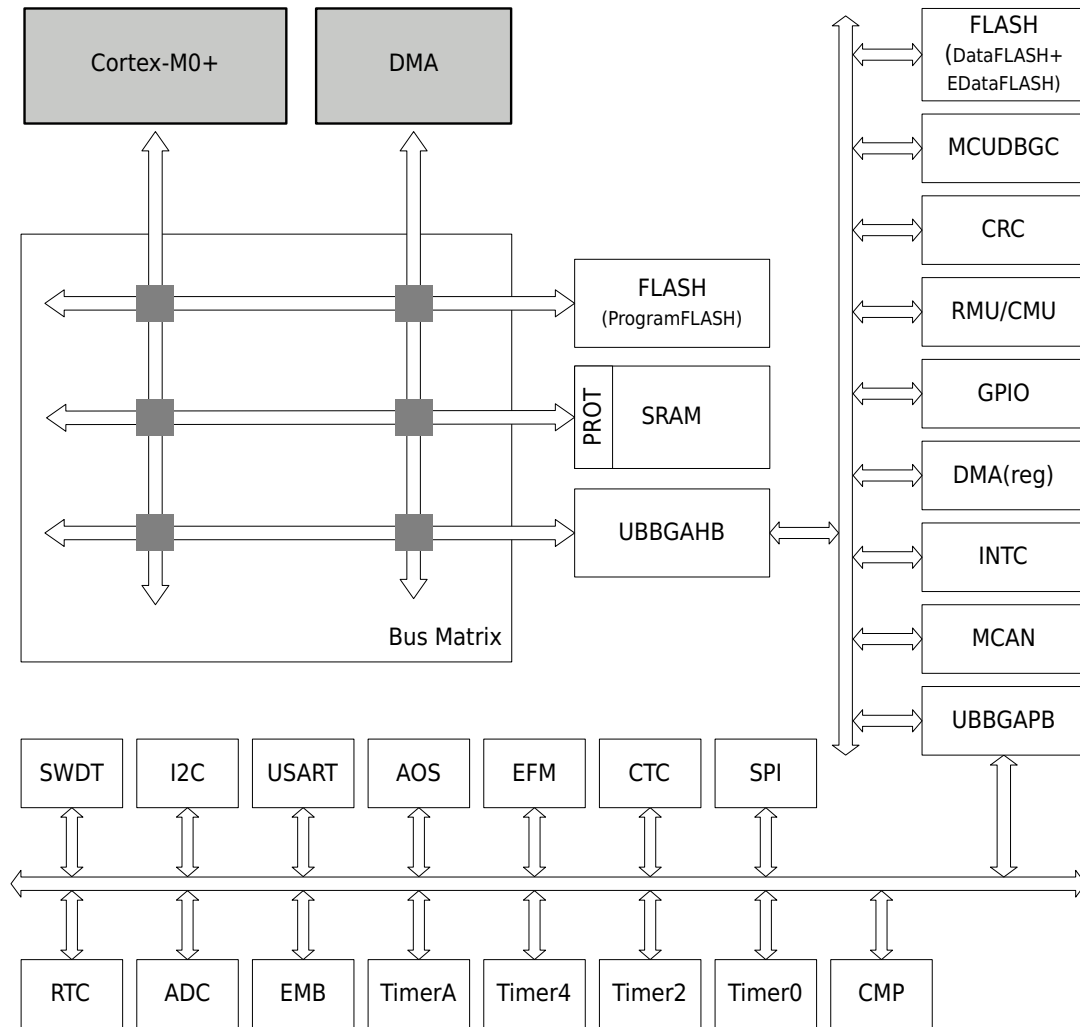


Figure 3-1 Bus Architecture Diagram



Note

- PROT: Control RAM access protection area, see PWC_RAMCR.RPRTA bit related description in "Power Control (PWC)" chapter for details.

3.3 Bus Function

The bus is responsible for implementing read and write access from the host to the slave. Access to AHB and APB peripherals can be achieved through the bus. The APB peripheral is connected to the AHB via a bus bridge.

Through the bus matrix, when the access targets of different host buses do not conflict, each access can be carried out simultaneously. For example, the CPU accesses Flash, and DMA accesses SRAM; The CPU accesses the APB peripheral, and the DMA accesses the SRAM. These accesses can be carried out simultaneously.

4 Reset Module Unit (RMU)

4.1 Introduction

The chip is configured with 11 reset modes:

- Power-on Reset (POR)
- NRST Pin Reset (NRST)
- Low Voltage Detect 1 Reset (LVD1R)
- Low Voltage Detect 2 Reset (LVD2R)
- Special Watchdog Reset (SWDTR)
- Software Reset (SRST)
- RAMECC Reset (RAMECCR)
- FLASHECC Reset (FLASHECCR)
- Clock Frequency Error Reset (CKFER)
- XTAL Error Reset (XTALER)
- Cortex-M0+ Lockup Reset

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
Low Voltage Detection 1 Reset	VCC voltage drops below LVD1 voltage
Low Voltage Detection 2 Reset	VCC voltage drops below LVD2 voltage
Special Watchdog Reset	Refresh error or overflow error occurred in special watchdog
Software Reset	Set the reset register bit (ARM register AIRCR.SYSRESETREQ bit)
RAM ECC Reset	Reset generated when ECC error occurs in RAM
FLASH ECC Reset	Reset generated when ECC error occurs in FLASH
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-M0+ Lockup Reset	When Cortex-M0+ 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	Voltage Detection 1 Reset	Voltage Detection 2 Reset	Special Watchdog Reset	Software Reset	RAM ECC Reset	FLASH ECC Reset	Clock Frequency Error Reset	XTAL Error Reset	Cortex-M0+ Lockup Reset
Power-on reset flag (RMU_RSTF0.PORF)	✓	-	-	-	-	-	-	-	-	-	-
Pin reset flag (RMU_RSTF0.PINRF)	-	✓	-	-	-	-	-	-	-	-	-
Low voltage detection 1 reset flag (RMU_RSTF0.LVD1RF)	-	-	✓	-	-	-	-	-	-	-	-
Low voltage detection 2 reset flag (RMU_RSTF0.LVD2RF)	-	-	-	✓	-	-	-	-	-	-	-
Special watchdog reset flag (RMU_RSTF0.WDRF)	-	-	-	-	✓	-	-	-	-	-	-
Software reset flag (RMU_RSTF0.SWRF)	-	-	-	-	-	✓	-	-	-	-	-
RAM ECC reset flag (RMU_RSTF0.RAECRF)	-	-	-	-	-	-	✓	-	-	-	-
FLASH ECC reset flag (RMU_RSTF0.FLECRF)	-	-	-	-	-	-	-	✓	-	-	-
Clock frequency error reset flag (RMU_RSTF0.CKFERF)	-	-	-	-	-	-	-	-	✓	-	-
XTAL error reset flag (RMU_RSTF0.XTALERF)	-	-	-	-	-	-	-	-	-	✓	-
Cortex-M0+ Lockup reset flag (RSTF0.CPULKUPRF)	-	-	-	-	-	-	-	-	-	-	✓

**Note**

"✓" indicates setting; "-" 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.

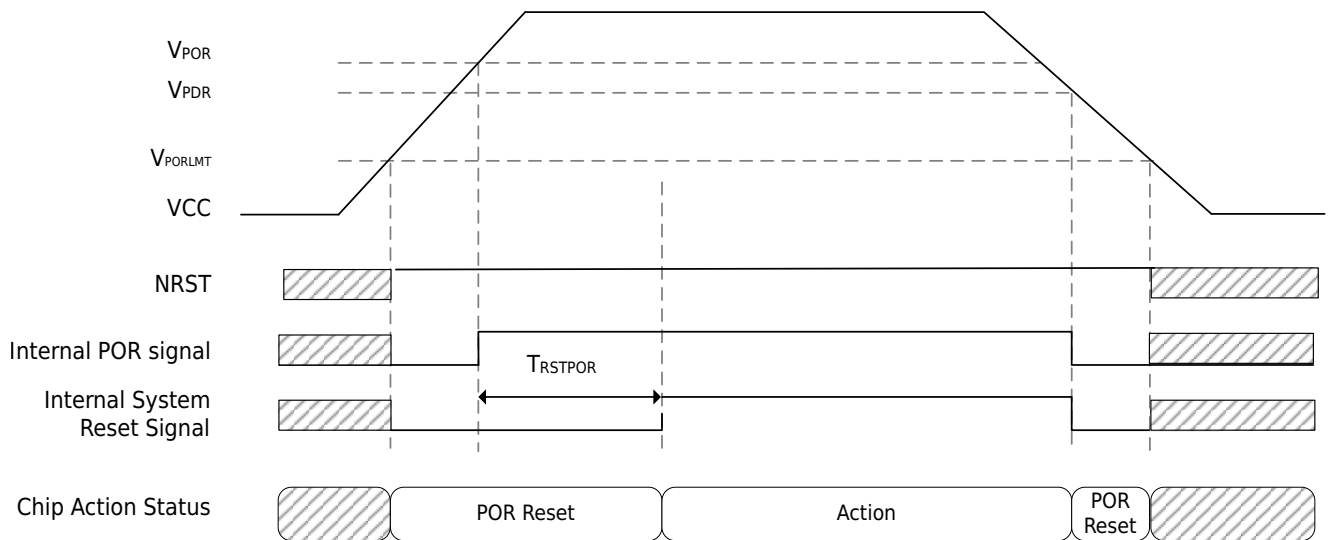


Figure 4-1 Power-on Reset Timing Diagram

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.

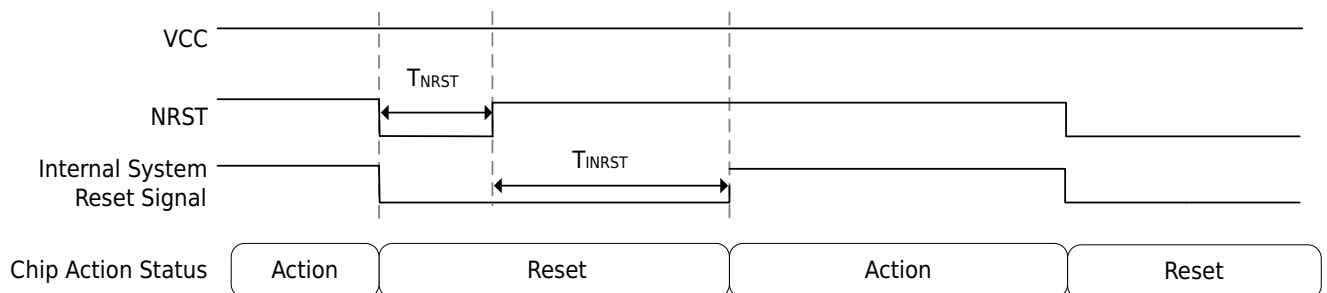


Figure 4-2 NRST Reset Timing Diagram

4.2.2.3 Low Voltage Detection 1 Reset, Low Voltage Detection 2 Reset

Low Voltage Detection 1 and Low Voltage Detection 2 resets are resets caused by the voltage monitoring circuit.

When the VCC voltage falls below the monitoring voltage of Low Voltage Detection 1, a Low Voltage Detection 1 reset can be generated according to the settings, and `RSTF0.LVD1RF` is set. After the VCC voltage rises above the monitoring voltage of Low Voltage Detection 1, the reset is released after the LVD1 reset time (T_{LVD1}).

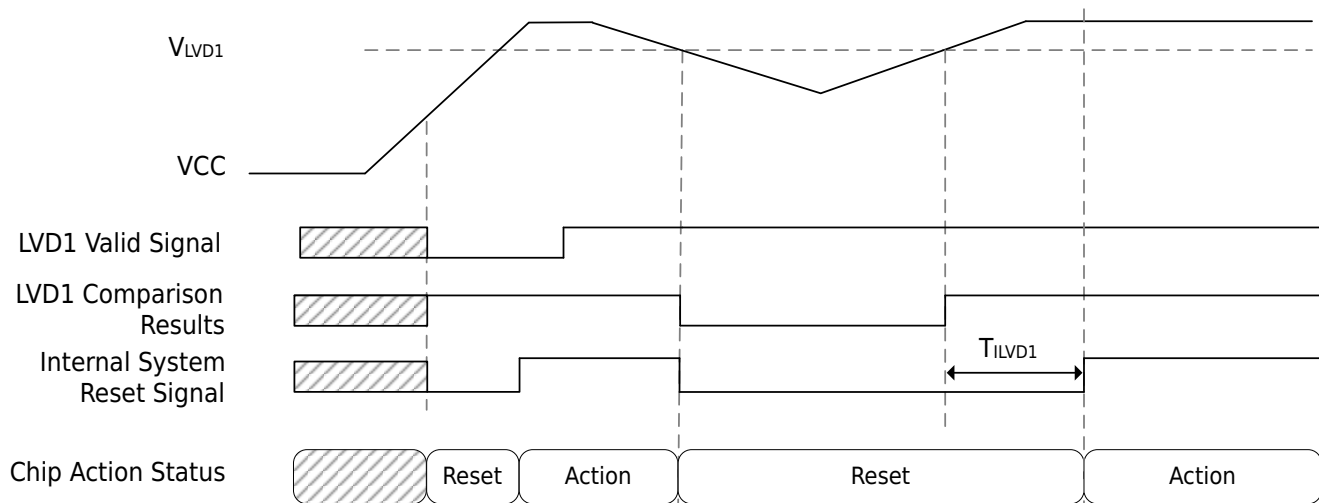


Figure 4-3 Low Voltage Detection 1 Reset Timing Diagram

When the VCC voltage falls below the monitoring voltage of Low Voltage Detection 2, a Low Voltage Detection 2 reset can be generated according to the settings, and RSTF0.LVD2RF is set. After the VCC voltage rises above the monitoring voltage of Low Voltage Detection 2, the reset is released after the LVD2 reset time (T_{ILVD2}).

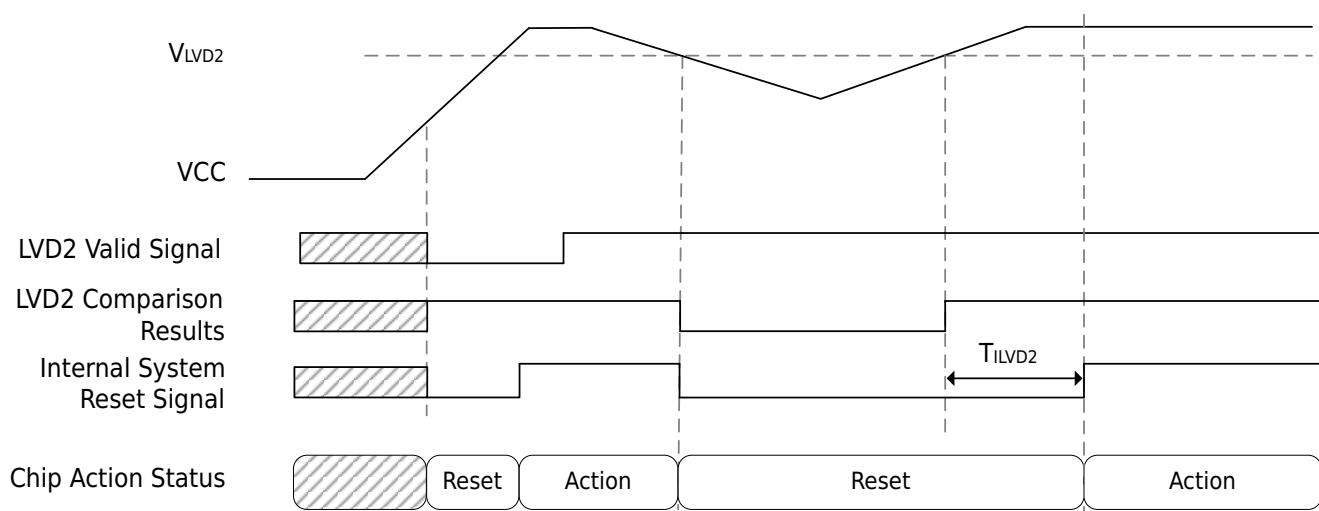


Figure 4-4 Low Voltage Detection 2 Reset Timing Diagram

For the reset settings of Low Voltage Detection 1 and Low Voltage Detection 2, please refer to the "[Power Voltage Detection Unit \(LVD\) Description](#)" section in the *Power Control (PWC)* chapter.

4.2.2.4 Special Watchdog Reset

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

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.WDRF setup. After generating special watchdog reset, after internal reset time T_{RIPT} , the chip is released.

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

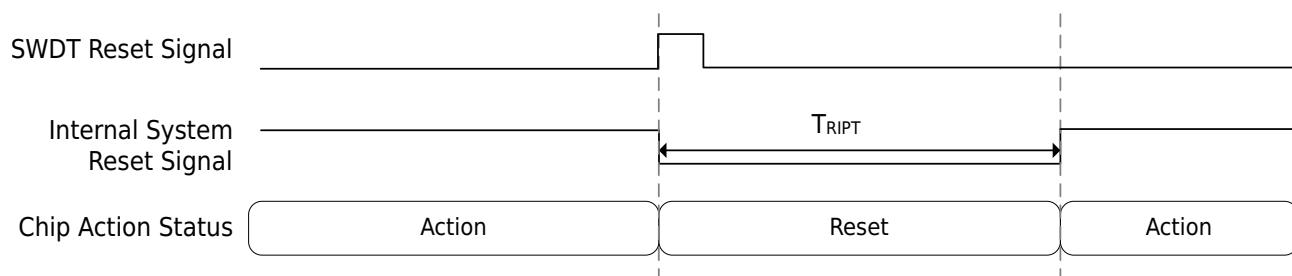


Figure 4-5 Special Watchdog Reset Timing Diagram

4.2.2.5 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.

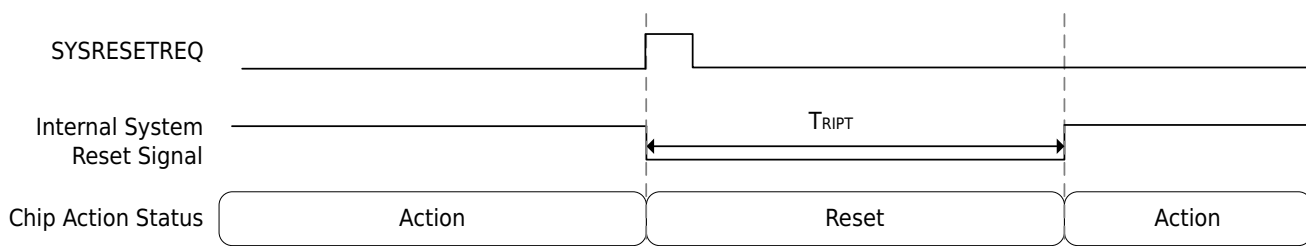


Figure 4-6 Software Reset Timing Diagram

4.2.2.6 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.

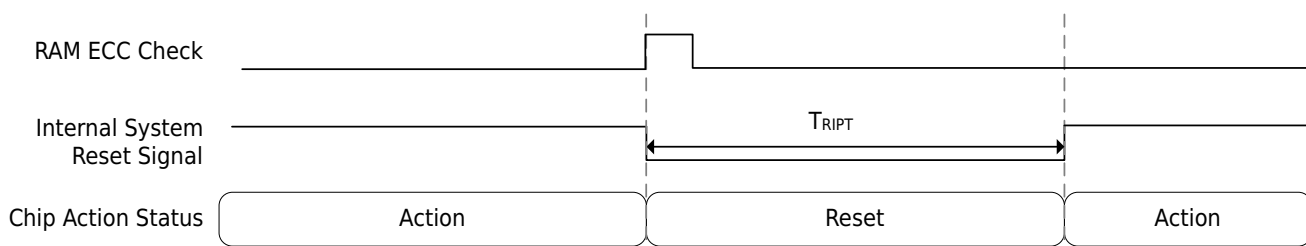


Figure 4-7 RAM ECC Reset Timing Diagram

4.2.2.7 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.

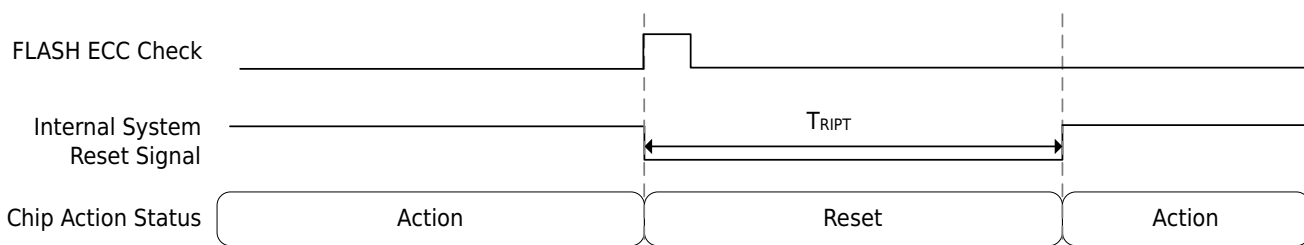


Figure 4-8 FLASH ECC Reset Timing Diagram

4.2.2.8 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.

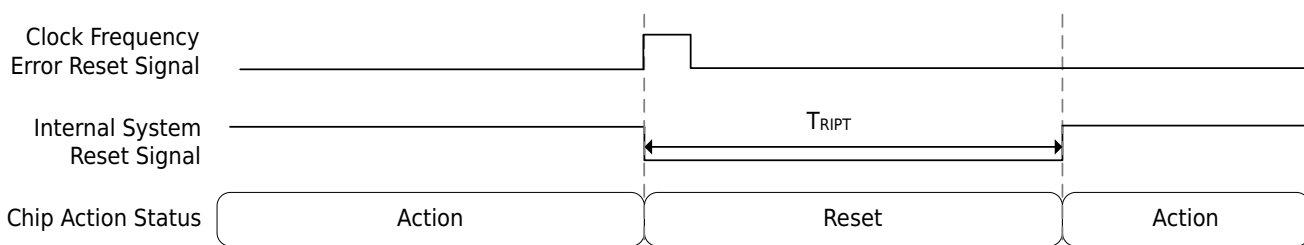


Figure 4-9 Clock Frequency Error Reset Timing Diagram

4.2.2.9 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.

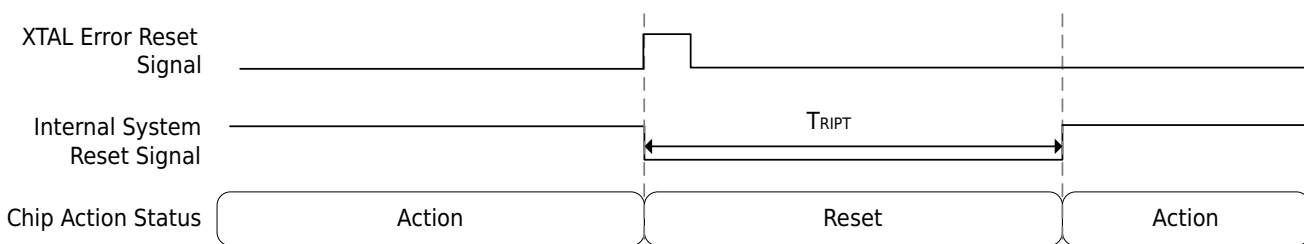


Figure 4-10 XTAL Error Reset Timing Diagram

4.2.2.10 Cortex-M0+ Lockup Reset

When Cortex-M0+ 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-M0+ Lockup reset sets `RMU_RSTF0.CPULKUPRF`. After the internal reset time T_{RIPT} , the chip is de-reset.

To utilize the Lockup reset function, the `ICGA1x.LKUPDIS` bit in the initialization configuration register must be programmed to 0.

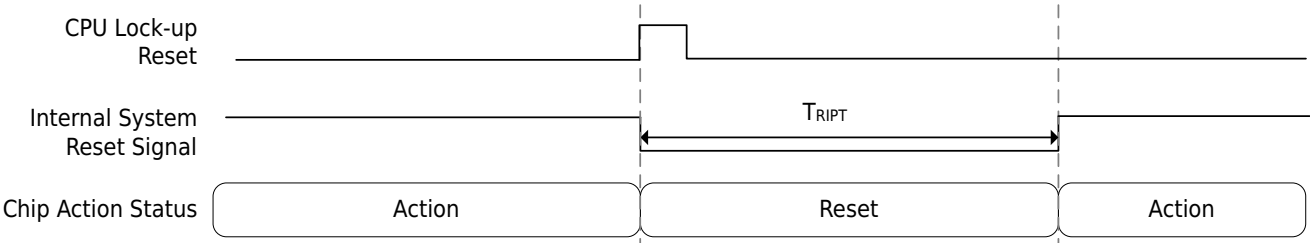


Figure 4-11 Cortex-M0+ Lockup Reset Timing Diagram

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 CLRf bit.

4.3 Register Description

4.3.1 Register List

Base address: 0x40014100

Table 4-3 Reset Module Unit (RMU) Register Offset

Offset	Register	Description	Width
0x00	RMU_RSTF0	Reset Flag Register 0	32

4.3.2 Reset Flag Register 0 (RMU_RSTF0)

Offset	Bit Position															
0x00	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0xXXXX (the reset value varies depending on the reset method)															
Name	CLRF	MULTIRF	XTALERF	CPULKUPRF	FLECRF	RAECRF	CKFERF	SWRF	Reserved	WDRF	Reserved	LVD2RF	LVD1RF	PINRF	PORF	
Access	RW	RO	RO	RO	RO	RO	RO	RO		RO		RO	RO	RO	RO	RO

Bit	Name	Function	Description	Access
15	CLRF	Clear Reset Flag	Write 1 to reset the flag, write 0 has no effect; read always returns 0.  Note The set action must be performed after RMU_RSTF0 is read.	RW
14	MULTIRF	2 or More Reset Occurrence Flags	Set by hardware when two or more resets occur. Clear by setting CLRF. Read: 0b0: Two or more resets did not occur 0b1: Two or more resets occurred	RO
13	XTALERF	XTAL Error Reset Flag	Set by hardware when a XTAL error reset occurs. Clear by setting CLRF. Read: 0b0: No reset occurred 0b1: Reset occurred	RO
12	CPULKUPRF	Cortex-M0+ Lockup Reset Flag	Set by hardware when a Cortex-M0+ Lockup reset occurs. Clear by setting CLRF. Read: 0b0: No reset occurred 0b1: Reset occurred	RO
11	FLECRF	FLASH ECC Reset Flag	Set by hardware when a FLASH ECC reset occurs. Clear by setting CLRF. Read: 0b0: No reset occurred 0b1: Reset occurred	RO
10	RAECRF	RAM ECC Reset Flag	Set by hardware when a RAM ECC reset occurs. Clear by setting CLRF. Read: 0b0: No reset occurred 0b1: Reset occurred	RO

Bit	Name	Function	Description	Access
9	CKFERF	Clock Frequency Error Reset Flag	Set by hardware when a clock frequency error reset occurs. Clear by setting CLRF. Read: ● 0b0: No reset occurred ● 0b1: Reset occurred	RO
8	SWRF	Software Reset Flag	Set by hardware when a software reset occurs. Clear by setting CLRF. Read: ● 0b0: No reset occurred ● 0b1: Reset occurred	RO
7:6	<i>Reserved</i>	<i>Reserved</i>	-	-
5	WDRF	Special Watchdog Reset Flag	Set by hardware when a special watchdog reset occurs. Clear by setting CLRF. Read: ● 0b0: No reset occurred ● 0b1: Reset occurred	RO
4	<i>Reserved</i>	<i>Reserved</i>	-	-
3	LVD2RF	Low Voltage Detection 2 Reset Flag	Set by hardware when a low voltage detection 2 reset occurs. Clear by setting CLRF. Read: ● 0b0: No reset occurred ● 0b1: Reset occurred	RO
2	LVD1RF	Low Voltage Detection 1 Reset Flag	Set by hardware when a low voltage detection 1 reset occurs. Clear by setting CLRF. Read: ● 0b0: No reset occurred ● 0b1: Reset occurred	RO
1	PINRF	NRST Pin Reset Flag	Set by hardware when a pin reset occurs. Clear by setting CLRF. Read: ● 0b0: No reset occurred ● 0b1: Reset occurred	RO
0	PORF	Power-on Reset Flag	Set by hardware when a power-on reset occurs. Clear by setting CLRF. Read: ● 0b0: No reset occurred ● 0b1: Reset occurred	RO

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, an internal high-speed oscillator, an internal low-speed oscillator, a clock prescaler, 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-M0+ clocks are derived from the System Clock. The maximum operating clock frequency of the System Clock can reach 64MHz, with 4 selectable clock sources:

1. External High-speed Oscillator (XTAL)
2. External Low-speed Oscillator (XTAL32)
3. Internal High-speed Oscillator (HRC)
4. 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.

5.2 Features

Table 5-1 Clock Source Main Features

Clock Source	Specifications
External High-speed Oscillator (XTAL)	Crystal frequency range: 4-40MHz External clock input: Up to 48MHz Oscillator fault detection function
External Low-speed Oscillator (XTAL32)	Crystal frequency: 32.768kHz
Internal High-speed Oscillator (HRC)	Frequency: 64MHz, 32MHz, 24MHz, 16MHz, 12MHz, 8MHz, 6MHz, 4MHz, 3MHz, 2MHz, 1.5MHz Configured via the ICGA1x.HRCFREQS bit
Internal Low-speed Oscillator (LRC)	Frequency: 32.768kHz Can be used as the counting clock for SWDT, Timer0, and Timer2

5.3.2 Working Clock Specifications

Table 5-2 Specifications for Each Internal Clock

Clock	Scope of action	Specifications
HCLK	CPU, DMA, AOS, CRC, DEFM (Data-FLASH control), EFM (ProgramFLASH control), SRAM, GPIO, INTC, USARTn (n=1-4), SPIn (n=1-2), Timer0, Timer2_n (n=1-2), Timer4, TimerA, I2C, SWDT (control logic), ADC(control logic), CTC	Maximum frequency: 64MHz
ADCLK	AD Conversion Clock	Maximum frequency: 32MHz  Note Under the XTAL clock input mode and when the XTAL clock is selected as the system clock, the maximum frequency of the AD conversion clock is 20 MHz.
STICCLK	The clock for the CPU's SysTickTimer counter uses LRC as its clock source	Frequency: 32.768kHz
SWDTCLK	SWDT Counter Clock	Frequency: 32.768kHz
TCK	SWD Clock	Maximum frequency: 20MHz

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.

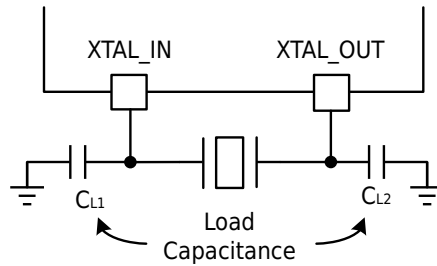


Figure 5-3 External High Speed Oscillator Connection Example

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 bit to 1 and the CMU_XTALCR.XTALSTP bit to 0. An external clock signal with a duty period of approximately 50% must be used to drive the XTAL_IN or XTAL_OUT pins. The terminal for driving the external clock signal is selected by setting the CMU_XTALCFGR.EXTALS bit, and the unselected terminal can be configured as GPIO according to register settings.

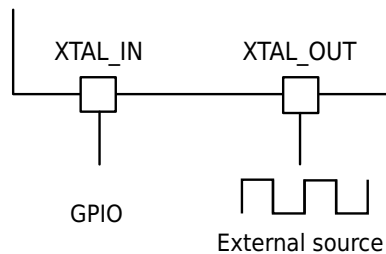


Figure 5-4 External Clock Input Connection Example

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.

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.

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](#)".

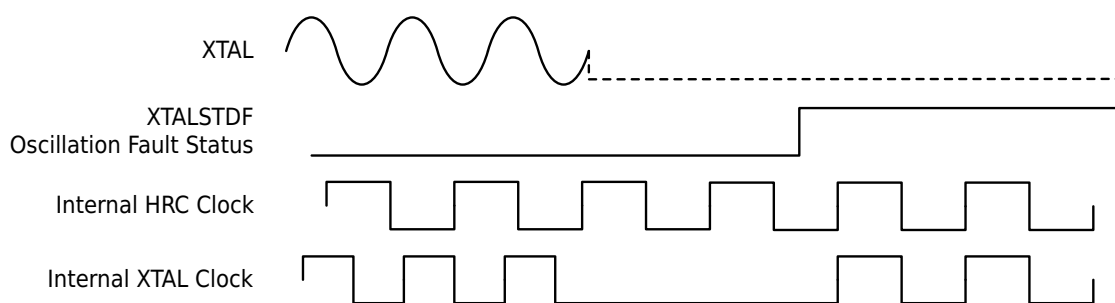


Figure 5-5 External High-speed Oscillator Fault Detection Example

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 HRC.

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

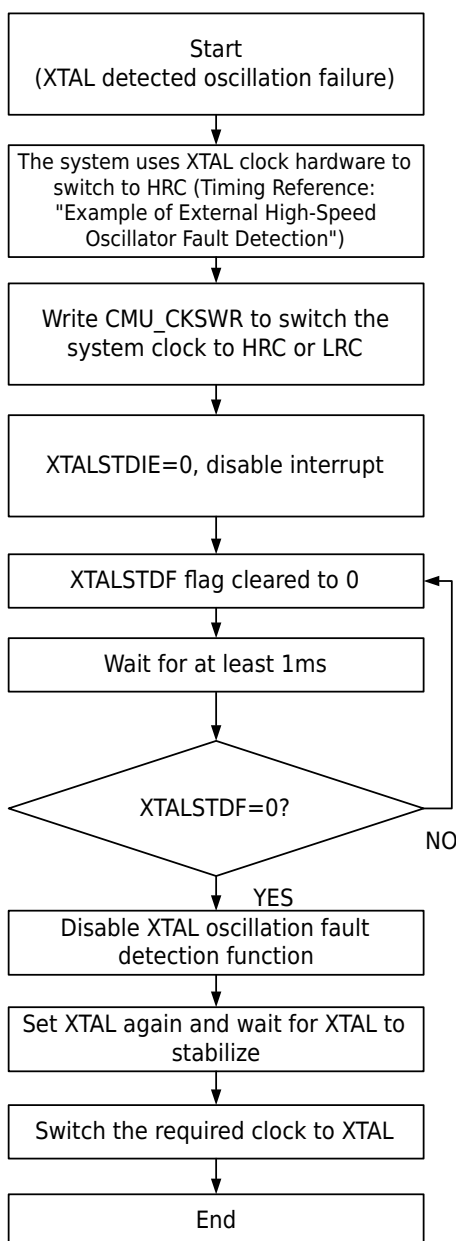


Figure 5-6 System Clock Selected as XTAL, Example of Detecting XTAL Oscillation Fault

5.3.3.2 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 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.

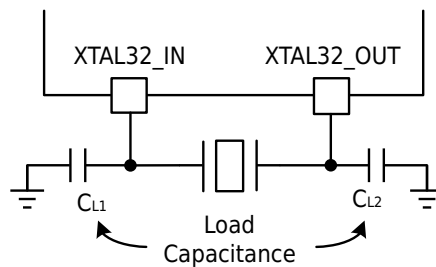


Figure 5-7 External Low-speed Oscillator Connection Example

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

1. Set the matching XTAL32 drive capability through CMU_XTAL32FGR;
2. Set the filter function through CMU_XTAL32NFR;
3. Write 0 to the CMU_XTAL32CR.XTAL32STP bit to oscillate XTAL32;
4. Wait for XTAL32 to stabilize via the CMU_OSCSTBSR.XTAL32STBF bit.

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

HRC clock signals are generated by the internal high-speed oscillator and can be directly used as the system clock. The frequency of HRC can be configured via ICGA1x.HRCFREQS to 64MHz, 32MHz, 24MHz, 16MHz, 12MHz, 8MHz, 6MHz, 4MHz, 3MHz, 2MHz, or 1.5MHz. It can also be set by software via the CMU_HRCCFGR.HRCFREQS register.

After reset release, the CPU must start operating with this internal high-speed oscillation clock.

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.

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 LRC Clock

LRC clock signals are generated by the internal 32.768kHz low-speed oscillator and can be directly used as the system clock. LRC can serve as a low-power clock source, remaining operational in Stop mode for use by Timer0/Timer2/SWDT.

The LRC oscillator has a fast start-up speed and can be used immediately after start-up without waiting for stabilization.

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.

LRC can be turned on or off through the CMU_LRCCR.LRCSTP bit.

5.3.5 Clock Switching Steps

After system reset, the default system clock is HRC. 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 should be configured correctly to prevent the system clock frequency from being larger than the maximum operating frequency of Flash. For detailed configuration, please refer to the "CPU Clock and Flash Read Time Relationship" section of the "Embedded Flash Memory (EFM)".

Clock Source Switching

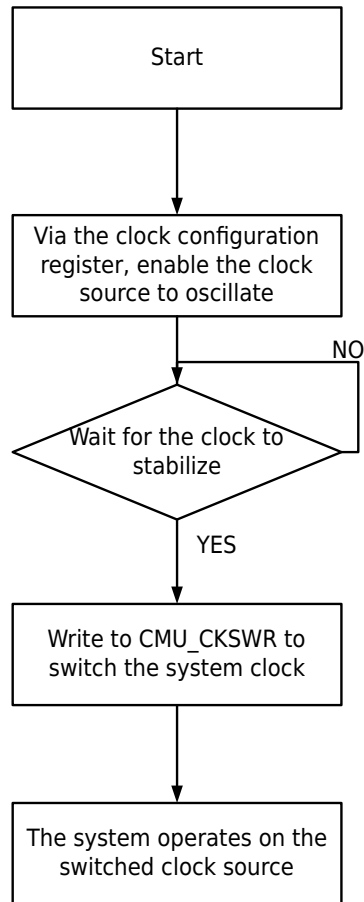


Figure 5-8 Clock Source Switching Flowchart

5.3.6 Clock Output Function

MCO:

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

- HRC Clock
- LRC Clock
- XTAL Clock
- XTAL32 Clock
- System Clock

The desired clock source is selected by the CMU_MCO1CFGR.MCO1SEL bit.

The output clock of MCO should not exceed 16MHz (maximum I/O speed).

5.3.7 Clock Frequency Measurement

5.3.7.1 Clock Frequency Measurement Function

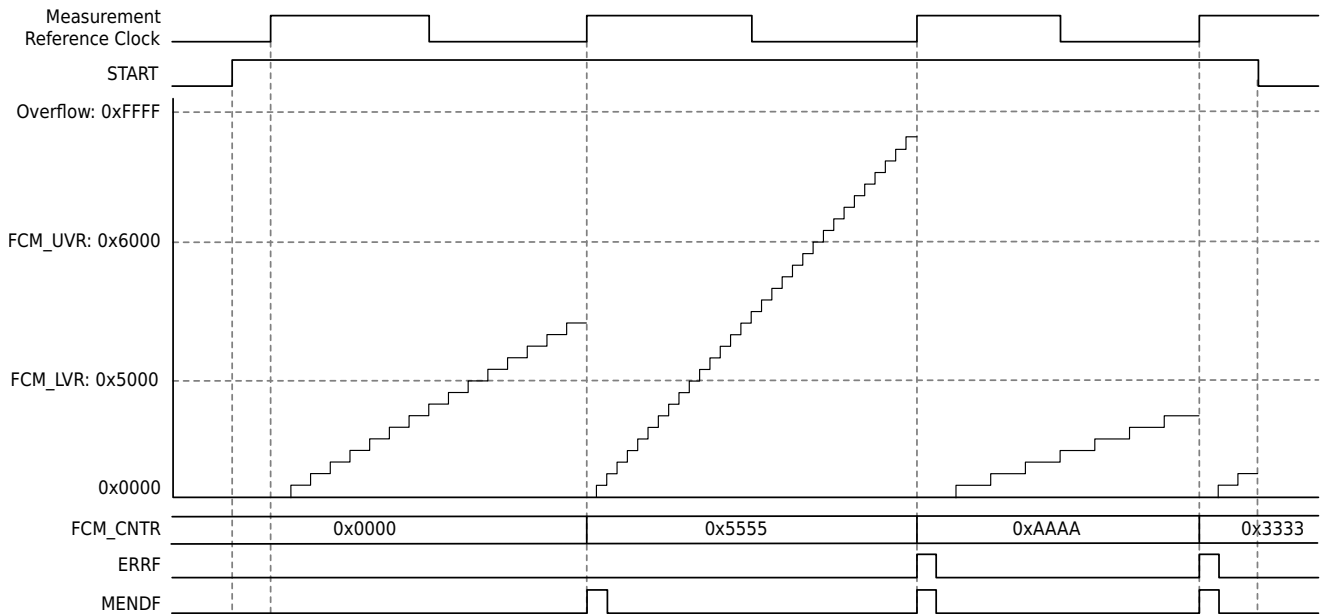


Figure 5-9 Clock Frequency Measurement Timing Diagram

- 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 $\text{FCM\_LVR} > \text{FCM\_CNTR}$ or $\text{FCM\_CNTR} > \text{FCM\_UVR}$, 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.

5.3.7.2 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.

5.3.7.3 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:

- Clock frequency exceptionally reset

5.4 Register Description

5.4.1 Register List

FCM base address: 0x40005400

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

Offset	Register	Description	Width
0x00	FCM_LVR	FCM Lower Limit Control Register	32
0x04	FCM_UVR	FCM Upper Limit Control Register	32
0x08	FCM_CNTR	FCM Counter Register	32
0x0C	FCM_STR	FCM Start Control Register	32
0x10	FCM_MCCR	FCM Measurement Control Register	32
0x14	FCM_RCCR	FCM Reference Control Register	32
0x18	FCM_RIER	FCM Reset Interrupt Control Register	32
0x1C	FCM_SR	FCM Flag Register	32
0x20	FCM_CLR	FCM Flag Clear Register	32

CMU base address 1: 0x40014400

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

Offset	Register	Description	Width
0x00	CMU_PERICKSEL	CMU ADC Clock Config Register	8
0x04	CMU_XTALSTDSR	CMU XTAL Clock Stop Detect Status Register	8
0x08	CMU_SCKDIVR	CMU Clock Divider Config Register	8
0x0C	CMU_CKSWR	CMU System Clock Source Switch Register	8
0x10	CMU_XTALCR	CMU XTAL Control Register	8
0x14	CMU_XTALCFGR	CMU XTAL Config Register	8
0x15	CMU_XTALSTBCR	CMU XTAL Stable Control Register	8
0x18	CMU_HRCCR	CMU HRC Control Register	8
0x20	CMU_OSCSTBSR	CMU Clock Source Stable Status Register	8
0x24	CMU_MCO1CFGR	CMU MCO Clock Output Config Register	8
0x28	CMU_XTALSTDCR	CMU XTAL Clock Stop Detect Control Register	8
0x30	CMU_FCG	CMU Function Clock Control Register	32
0x34	CMU_XTAL32CR	CMU XTAL32 Control Register	8

Offset	Register	Description	Width
0x38	CMU_XTAL32CFGR	CMU XTAL32 Config Register	8
0x39	CMU_XTAL32NFR	CMU XTAL32 Filter Control Register	8
0x3C	CMU_LRCCR	CMU LRC Control Register	8
0x40	CMU_CANCKCFGR	CMU MCAN Clock Config Register	8

CMU base address 2: 0x40000800

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

Offset	Register	Description	Width
0x0286	CMU_HRCCFGR	CMU HRC Config Register	8

5.4.2 CMU XTAL Config Register (CMU_XTALCFGR)

Offset	Bit Position							
0x14	7	6	5	4	3	2	1	0
Reset	0x80							
Name	SUPDRV	XTALMS	EXTALS	RFCUT	XTALDRV			
Access	RW	RW	RW	RW	RW			

Bit	Name	Function	Description	Access
7	SUPDRV	XTAL Ultra-High-Speed Drive Enabled	0b0: Disable ultra-high-speed drive 0b1: Enable ultra-high-speed drive  Note When ultra-high-speed drive is enabled, this bit setting is ignored after the XTAL stabilizes, and ultra-high-speed drive is disabled to reduce power consumption.	RW
6	XTALMS	XTAL Mode Selection Bit	0b0: Oscillator mode 0b1: External clock input mode	RW
5	EXTALS	EXTAL Pin Selection	0b0: Select pin XTAL_IN 0b1: Select pin XTAL_OUT	RW
4	RFCUT	Feedback Resistor Selection Bit	0b0: XTAL uses internal feedback resistor 0b1: XTAL uses external feedback resistor	RW
3:0	XTALDRV	XTAL Drive Capability Selection	0b1000: Drive crystal oscillator frequency: $35\text{MHz} \leq \text{Crystal oscillator frequency} \leq 40\text{MHz}$ 0b1001: Drive crystal oscillator frequency: $30\text{MHz} \leq \text{Crystal oscillator frequency} < 35\text{MHz}$ 0b1010: Drive crystal oscillator frequency: $25\text{MHz} \leq \text{Crystal oscillator frequency} < 30\text{MHz}$ 0b1011: Drive crystal oscillator frequency: $20\text{MHz} \leq \text{Crystal oscillator frequency} < 25\text{MHz}$ 0b1100: Drive crystal oscillator frequency: $16\text{MHz} \leq \text{Crystal oscillator frequency} < 20\text{MHz}$ 0b1101: Drive crystal oscillator frequency: $12\text{MHz} \leq \text{Crystal oscillator frequency} < 16\text{MHz}$ 0b1110: Drive crystal oscillator frequency: $4\text{MHz} < \text{Crystal oscillator frequency} < 12\text{MHz}$ 0b1111: Drive crystal oscillator frequency: 4MHz	RW

5.4.3 CMU XTAL Stable Control Register (CMU_XTALSTBCR)

Offset	Bit Position							
0x15	7	6	5	4	3	2	1	0
Reset	0x00							
Name	Reserved					XTALSTB		
Access						RW		

Bit	Name	Function	Description	Access
7:3	<i>Reserved</i>	<i>Reserved</i>	-	-
2:0	XTALSTB	XTAL Stabilization Time Selection	0b000: $2^8 \times \text{XTAL}$ periods 0b001: $2^9 \times \text{XTAL}$ periods 0b010: $2^{10} \times \text{XTAL}$ periods 0b011: $2^{11} \times \text{XTAL}$ periods 0b100: $2^{13} \times \text{XTAL}$ periods 0b101: $2^{15} \times \text{XTAL}$ periods 0b110: $2^{16} \times \text{XTAL}$ periods 0b111: $2^{17} \times \text{XTAL}$ periods  Note This register can only be configured when CMU_XTALCR.XTALSTP=1 and CMU_OSCSTBSR.XTALSTBF=0. When CMU_XTALCFGR.XTALMS=1, this waiting time is ignored.	RW

5.4.4 CMU XTAL Control Register (CMU_XTALCR)

Offset	Bit Position							
0x10	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	0b0: XTAL oscillator oscillating 0b1: XTAL oscillator stopped	RW

**Notice**

When XTAL is selected as the system clock source, it is prohibited to stop the XTAL oscillator by writing 1 to XTALSTP.

5.4.5 CMU XTAL Clock Stop Detect Control Register (CMU_XTALSTDCR)

Offset	Bit Position							
0x28	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	0b0: XTAL clock stop detect disabled 0b1: XTAL clock stop detect enabled  Note Clock stop detect involves detecting abnormal oscillations caused by external factors. Before entering Stop mode, please disable the clock stop detect function.	RW
6:3	Reserved	Reserved	-	-
2	XTALSTDRIS	XTAL Clock Stop Detect Reset Interrupt Selection	0b0: XTAL clock stop detect generates an interrupt 0b1: XTAL clock stop detect generates a reset	RW
1	XTALSTDRE	XTAL Clock Stop Detect Reset Enabled	0b0: XTAL clock stop detect reset disabled 0b1: XTAL clock stop detect reset enabled	RW
0	XTALSTDIE	XTAL Clock Stop Detect Interrupt Enabled	0b0: XTAL clock stop detect interrupt disabled 0b1: XTAL clock stop detect interrupt enabled	RW

5.4.6 CMU XTAL Clock Stop Detect Status Register (CMU_XTALSTDSR)

Offset	Bit Position							
0x04	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	0b0: XTAL clock stop not detected 0b1: XTAL clock stop detected Setting conditions: - XTAL clock stop detect under XTALSTDE=1 Clearing conditions: - When the system clock selects a clock other than XTAL, read as 1, write as 0	RW

5.4.7 CMU XTAL32 Config Register (CMU_XTAL32CFGR)

Offset	Bit Position							
0x38	7	6	5	4	3	2	1	0
Reset	0x00							
Name	Reserved							XTAL32DRV
Access								RW

Bit	Name	Function	Description	Access
7:1	Reserved	Reserved	-	-
0	XTAL32DRV	XTAL32 Drive Capability Selection	0b0: Medium drive capability 0b1: High drive capability	RW

5.4.8 CMU XTAL32 Filter Control Register (CMU_XTAL32NFR)

Offset	Bit Position							
0x39	7	6	5	4	3	2	1	0
Reset	0x00							
Name	Reserved						XTAL32NF	
Access							RW	

Bit	Name	Function	Description	Access
7:2	Reserved	Reserved	-	-
1:0	XTAL32NF	XTAL32 Oscillator Filter Selection	0b0x: XTAL32 3μs filter enabled in RUN mode/ Stop mode 0b10: XTAL32 3μs filter enabled in RUN mode, disabled in Stop mode 0b11: XTAL32 3μs filter disabled in RUN mode/ Stop mode	RW

5.4.9 CMU XTAL32 Control Register (CMU_XTAL32CR)

Offset	Bit Position							
0x34	7	6	5	4	3	2	1	0
Reset	0x01							
Name	Reserved						XTAL32STP	
Access							RW	

Bit	Name	Function	Description	Access
7:1	Reserved	Reserved	-	-
0	XTAL32STP	XTAL32 Oscillator Start/Stop bit	0b0: XTAL32 oscillator oscillating 0b1: XTAL32 oscillator stopped	RW

**Notice**

When XTAL32 is selected as the system clock source, it is prohibited to stop the XTAL32 oscillator by writing 1 to XTAL32STP.

5.4.10 CMU HRC Config Register (CMU_HRCCFGR)

Offset	Bit Position							
0x0286	7	6	5	4	3	2	1	0
Reset	Dependent on the ICGA1x.HRCFREQS value							
Name	Reserved				HRCFREQS			
Access					RW			

Bit	Name	Function	Description	Access																							
7:4	Reserved	Reserved	-	-																							
3:0	HRCFREQS	HRC Frequency Selection Bit	HRCFREQS[3] can only be set by IC-GA1x.HRCFREQS[3] and cannot be modified by software. HRCFREQS[2:0] can be set by IC-GA1x.HRCFREQS[2:0] or modified by software. Write protection for this register is removed by setting the EFM_FAPRT register. <table><tr><th rowspan="2">HRCFREQS[2:0]</th><th colspan="2">HRC Frequency</th></tr><tr><th>HRCFREQS[3]=0</th><th>HRCFREQS[3]=1</th></tr><tr><td>0b000</td><td>64MHz</td><td>48MHz</td></tr><tr><td>0b001</td><td>32MHz</td><td>24MHz</td></tr><tr><td>0b010</td><td>16MHz</td><td>12MHz</td></tr><tr><td>0b011</td><td>8MHz</td><td>6MHz</td></tr><tr><td>0b100</td><td>4MHz</td><td>3MHz</td></tr><tr><td>0b101-111</td><td>2MHz</td><td>1.5MHz</td></tr></table>	HRCFREQS[2:0]	HRC Frequency		HRCFREQS[3]=0	HRCFREQS[3]=1	0b000	64MHz	48MHz	0b001	32MHz	24MHz	0b010	16MHz	12MHz	0b011	8MHz	6MHz	0b100	4MHz	3MHz	0b101-111	2MHz	1.5MHz	RW
HRCFREQS[2:0]	HRC Frequency																										
	HRCFREQS[3]=0	HRCFREQS[3]=1																									
0b000	64MHz	48MHz																									
0b001	32MHz	24MHz																									
0b010	16MHz	12MHz																									
0b011	8MHz	6MHz																									
0b100	4MHz	3MHz																									
0b101-111	2MHz	1.5MHz																									

5.4.11 CMU HRC Control Register (CMU_HRCCR)

Offset	Bit Position							
0x18	7	6	5	4	3	2	1	0
Reset	0x00							
Name	Reserved							HRCSTP
Access								RW

Bit	Name	Function	Description	Access
7:1	Reserved	Reserved	-	-
0	HRCSTP	HRC Oscillator Start Stop Bit	0b0: HRC oscillator oscillating 0b1: HRC oscillator stopped	RW

**Notice**

When HRC is selected as the system clock source, it is prohibited to stop the HRC clock by writing 1 to HRCSTP.

5.4.12 CMU LRC Control Register (CMU_LRCCR)

Offset	Bit Position							
0x3C	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	0b0: LRC oscillator oscillating 0b1: LRC oscillator stopped	RW

**Notice**

- When LRC is selected as the system clock source, stopping the LRC clock by writing 1 to LRCSTP is prohibited.

5.4.13 CMU Clock Source Stable Status Register (CMU_OSCSTBSR)

Offset	Bit Position							
0x20	7	6	5	4	3	2	1	0
Reset	0x00							
Name	Reserved			XTAL32STBF	XTALSTBF	Reserved		HRCSTBF
Access				RO	RO			RO

Bit	Name	Function	Description	Access
7:5	Reserved	Reserved	-	-
4	Reserved	Reserved	-	-
4	XTAL32STBF	XTAL32 Stability Flag Bit	0b0: XTAL32 is stopped or unstable 0b1: XTAL32 is stable	RO
3	XTALSTBF	XTAL Stability Flag Bit	0b0: XTAL is stopped or unstable 0b1: XTAL is stable	RO
2:1	Reserved	Reserved	-	-
0	HRCSTBF	HRC Stability Flag Bit	0b0: HRC is stopped or unstable 0b1: HRC is stable	RO

5.4.14 CMU System Clock Source Switch Register (CMU_CKSWR)

Offset	Bit Position							
0x0C	7	6	5	4	3	2	1	0
Reset	0x00							
Name	Reserved						CKSW	
Access							RW	

Bit	Name	Function	Description	Access
7:2	<i>Reserved</i>	<i>Reserved</i>	-	-
1:0	CKSW	System Clock Source Switch	0b00: Select HRC clock as the system clock 0b01: Select XTAL clock as the system clock 0b10: Select LRC clock as the system clock 0b11: Select XTAL32 clock as the system clock  Note - The target clock source for switching must be in a stable state. For the operation procedure, refer to Clock Source Switching. - When the PWC_STPMCR.CKSMRC bit is 1, this register is initialized after wake-up in the Stop mode, and the HRC clock is selected as the system clock source.	RW

5.4.15 CMU Clock Divider Config Register (CMU_SCKDIVR)

Offset	Bit Position							
0x08	7	6	5	4	3	2	1	0
Reset	0x00							
Name	Reserved					SCKDIV		
Access						RW		

Bit	Name	Function	Description	Access
7:3	<i>Reserved</i>	<i>Reserved</i>	-	-
2:0	SCKDIV	CPU and Peripheral Clock (HCLK) Division Coefficient Selection	0b000: System clock 0b001: 2 division of system clock 0b010: 4 division of the system clock 0b011: 8 division of the system clock 0b100: 16 division of the system clock 0b101: 32 division of the system clock 0b110: 64 division of the system clock - Others: Setting is prohibited  Notice When the Stop mode is reset, the setting depends on the PWC_STPMCR.CKSMRC bit. If CKSMRC is 1, this register is initialized after wake-up from Stop mode. If CKSMRC is 0, the division coefficient setting before entering Stop mode is retained after wake-up.	RW

5.4.16 CMU MCAN Clock Config Register (CMU_CANCKCFGR)

Offset	Bit Position							
0x40	7	6	5	4	3	2	1	0
Reset	0x00							
Name	Reserved							CANCKS
Access								RW

Bit	Name	Function	Description	Access
7:1	Reserved	Reserved	-	-
0	CANCKS	MCAN Communication Clock Selection	● 0b0: XTAL ● 0b1: HCLK  Note When the CMU_FCG.MCAN bit is 1 and the MCAN function is disabled, this bit can be rewritten.	RW

5.4.17 CMU ADC Clock Config Register (CMU_PERICKSEL)

Offset	Bit Position							
0x00	7	6	5	4	3	2	1	0
Reset	0x00							
Name	Reserved					PERICKSEL		
Access						RW		

Bit	Name	Function	Description	Access
7:3	Reserved	Reserved	-	-
2:0	PERICKSEL	ADC Conversion Clock (ADCLK) Division Selection	0b000: HCLK 0b001: 2 division of the HCLK 0b010: 4 division of the HCLK 0b011: 8 division of the HCLK 0b100: 16 division of the HCLK 0b101: 32 division of the HCLK - Others: Setting prohibited	RW

5.4.18 CMU Function Clock Control Register (CMU_FCG)

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	0xFFFFFFFF																															
Name	Reserved	SPI2	SPI1	I2C	USART4	USART3	USART2	USART1	RTC	Reserved	EMB	TMRA	TMR4	TMR2_2	TMR2_1	Reserved	MCAN	FCM	Reserved	Reserved	TMR0	CRC	Reserved	DMA	AOS	Reserved	CMP	CTC	ADC			
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	All bits read as 1, Write as 1.	-
30	SPI2	SPI2 Function Control	0b0: Function enabled 0b1: Function disabled	RW
29	SPI1	SPI1 Function Control	0b0: Function enabled 0b1: Function disabled	RW
28	I2C	I2C Function Control	0b0: Function enabled 0b1: Function disabled	RW
27	USART4	USART4 Function Control	0b0: Function enabled 0b1: Function disabled	RW
26	USART3	USART3 Function Control	0b0: Function enabled 0b1: Function disabled	RW
25	USART2	USART2 Function Control	0b0: Function enabled 0b1: Function disabled	RW
24	USART1	USART1 Function Control	0b0: Function enabled 0b1: Function disabled	RW
23	RTC	RTC Function Control	0b0: Function enabled 0b1: Function disabled	RW
22:21	Reserved	Reserved	All bits read as 1, Write as 1.	-
20	EMB	EMB Function Control	0b0: Function enabled 0b1: Function disabled	RW
19	TMRA	TimerA Function Control	0b0: Function enabled 0b1: Function disabled	RW
18	TMR4	Timer4 Function Control	0b0: Function enabled 0b1: Function disabled	RW
17	TMR2_2	Timer2_2 Function Control	0b0: Function enabled 0b1: Function disabled	RW
16	TMR2_1	Timer2_1 Function Control	0b0: Function enabled 0b1: Function disabled	RW

Bit	Name	Function	Description	Access
15:14	<i>Reserved</i>	<i>Reserved</i>	<i>All bits read as 1, Write as 1.</i>	-
13	MCAN	MCAN Function Control	● 0b0: Function enabled ● 0b1: Function disabled	RW
12	FCM	FCM Function Control	● 0b0: Function enabled ● 0b1: Function disabled	RW
11:9	<i>Reserved</i>	<i>Reserved</i>	<i>All bits read as 1, Write as 1.</i>	-
8	TMR0	Timer0 Function Control	● 0b0: Function enabled ● 0b1: Function disabled	RW
7	CRC	CRC Function Control	● 0b0: Function enabled ● 0b1: Function disabled	RW
6	<i>Reserved</i>	<i>Reserved</i>	<i>All bits read as 1, Write as 1.</i>	-
5	DMA	DMA Function Control	● 0b0: Function enabled ● 0b1: Function disabled	RW
4	AOS	AOS Function Control	● 0b0: Function enabled ● 0b1: Function disabled	RW
3	<i>Reserved</i>	<i>Reserved</i>	<i>All bits read as 1, Write as 1.</i>	-
2	CMP	CMP Function Control	● 0b0: Function enabled ● 0b1: Function disabled	RW
1	CTC	CTC Function Control	● 0b0: Function enabled ● 0b1: Function disabled	RW
0	ADC	ADC Function Control	● 0b0: Function enabled ● 0b1: Function disabled	RW

5.4.19 CMU MCO1 Clock Output Config Register (CMU_MCO1CFGR)

Offset	Bit Position							
0x24	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 Output Permission	● 0b0: Disable ● 0b1: Enable	RW
6:4	MCO1DIV	MCO Frequency Division Selection	● 0b000: Divide by 1 ● 0b001: Divide by 2 ● 0b010: Divide by 4 ● 0b011: Divide by 8 ● 0b100: Divide by 16 ● 0b101: Divide by 32 ● 0b110: Divide by 64 ● 0b111: Divide by 128	RW
3:0	MCO1SEL	MCO Clock Source Selection	● 0b0000: HRC clock ● 0b0001: LRC clock ● 0b0010: XTAL clock ● 0b0011: XTAL32 clock ● 0b0100: System clock ● Others: Setting prohibited	RW

5.4.20 FCM Lower Limit Control Register (FCM_LVR)

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																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

5.4.21 FCM Upper Limit Control Register (FCM_UVR)

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																UVR															
Access																	RW															

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

5.4.22 FCM Counter Register (FCM_CNTR)

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																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

5.4.23 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																																		START			
Access																																			RW			

Bit	Name	Function	Description	Access
31:1	Reserved	Reserved	-	-
0	START	Frequency Measurement Start Bit	0b0: Frequency measurement stop 0b1: Frequency measurement start	RW

5.4.24 FCM Measurement Control Register (FCM_MCCR)

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																								MCKS				Reserved		MDIVS	
Access																									RW						RW	

Bit	Name	Function	Description	Access
31:8	Reserved	Reserved	-	-
7:4	MCKS	Measurement Object Clock Selection Bit	0b0000: XTAL 0b0001: XTAL32 0b0010: HRC 0b0011: LRC 0b0100: HCLK - Others: Setting prohibited	RW
3:2	Reserved	Reserved	-	-
1:0	MDIVS	Measurement Object Division Selection	0b00: No division 0b01: Divide by 4 0b10: Divide by 8 0b11: Divide by 32	RW

5.4.25 FCM Reference Control Register (FCM_RCCR)

Offset	Bit Position															
0x14	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reset	0x00000000															
Name	Reserved												EXREFE	Reserved	EDGES	Reserved
Access													RW		RW	

Bit	Name	Function	Description	Access
31:16	Reserved	Reserved	-	-
15	EXREFE	External Pin Input Reference Clock FCMREF Enable Bit	0b0: Disable external pin input reference clock FCMREF 0b1: Enable external pin input reference clock FCMREF	RW
14	Reserved	Reserved	-	-
13:12	EDGES	Measurement Reference Edge Selection Bit	0b00: Rising edge 0b01: Falling Edge 0b10: Rising and falling edges 0b11: Setting prohibited	RW
11:10	Reserved	Reserved	-	-
9:8	DNFS	Digital Filtering Function Selection Bit	0b00: No filter function 0b01: Clock selected by MCKS bit as filter clock 0b10: 4 divisions of the clock selected by MCKS bit as filter clock 0b11: 16 divisions of the clock selected by MCKS bit as filter clock	RW
7	INEXS	Measurement Reference, Internal Clock, and Terminal Selection Bit	0b0: External pin input reference clock FCMREF 0b1: Clock selected by RCKS bit	RW
6:3	RCKS	Measurement Reference Clock Selection Bit	0b0000: XTAL 0b0001: XTAL32 0b0010: HRC 0b0011: LRC 0b0100: HCLK - Others: Setting prohibited	RW
2	Reserved	Reserved	-	-

Bit	Name	Function	Description	Access
1:0	RDIVS	Measurement Reference Division Selection	● 0b00: Divide by 32 ● 0b01: Divide by 128 ● 0b10: Divide by 1024 ● 0b11: Divide by 8192	RW

5.4.26 FCM Reset Interrupt Control Register (FCM_RIER)

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																								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	0b0: Disable 0b1: Enable	RW
6:5	Reserved	Reserved	-	-
4	ERRINTRS	Frequency Exception Interrupt Reset Selection Bit	0b0: Interrupt occurs when frequency exception occurs 0b1: Reset occurs when frequency exception occurs	RW
3	Reserved	Reserved	-	-
2	OVFIE	Counter Overflow Interrupt Enable Bit	0b0: Disable counter overflow interrupt 0b1: Enable counter overflow interrupt	RW
1	MENDIE	Measurement End Interrupt Enable Bit	0b0: Disable interrupt when measurement ends 0b1: Enable interrupt when measurement ends	RW
0	ERRIE	Frequency Exception Interrupt Enable Bit	0b0: Disable interrupt when frequency exception occurs 0b1: Enable interrupt when frequency exception occurs	RW

5.4.27 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	11	10	9	8	7	6	5	4	3	2	1	0	
Reset	0x00000000																																
Name	Reserved																												OVF		MENDF		ERRF
Access																													RO	RO	RO		

Bit	Name	Function	Description	Access
31:3	Reserved	Reserved	-	-
2	OVF	Counter Over-flow Flag Bit	0b0: Counter has not overflowed 0b1: Counter has overflowed	RO
1	MENDF	Measurement End Flag Bit	0b0: In measurement 0b1: Measurement end	RO
0	ERRF	Frequency Exception Flag Bit	0b0: No frequency exception occurred 0b1: Frequency exception occurred	RO

5.4.28 FCM Flag Clear Register (FCM_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																												OVFCLR		MENDFCLR		ERRFCLR	
WO																													WO		WO			
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 operating voltage (VCC) of the chip is 2.7V to 5.5V.

The voltage detection unit provides functions such as power-on reset (POR), power-off reset (PDR), low voltage detection 1 (LVD1), and low voltage detection 2 (LVD2). Among them, POR and PDR control the reset action of the chip by detecting the VCC voltage. LVD1 detects the VCC voltage and reset or non-maskable interrupt the chip according to the register setting. LVD2 detects the VCC voltage or external input comparison, and sets the chip to reset or non-maskable interrupt according to the register setting.

6.2 Functional Description

6.2.1 Power Distribution

6.2.1.1 Power Distribution Structure

The chip consists of VCC domain, core voltage domain and AVCC power domain.

- The VCC domain is powered by VCC/VSS pins and consists of power control logic (PWC), low voltage detection unit (LVD1/LVD2), voltage regulator (LDO) and oscillator circuit. The oscillator circuit includes an external high-speed oscillator (XTAL), an external low-speed oscillator (XTAL32), etc.
- The core voltage domain consists of digital logic such as CPU, digital peripherals, RAM, FLASH, etc., and is powered by the core voltage generated by LDO.
- The AVCC power domain is an analog power domain, mainly composed of digital-to-analog converter (ADC) and voltage comparator (CMP), and is powered by the AVCC/VSS pins.

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

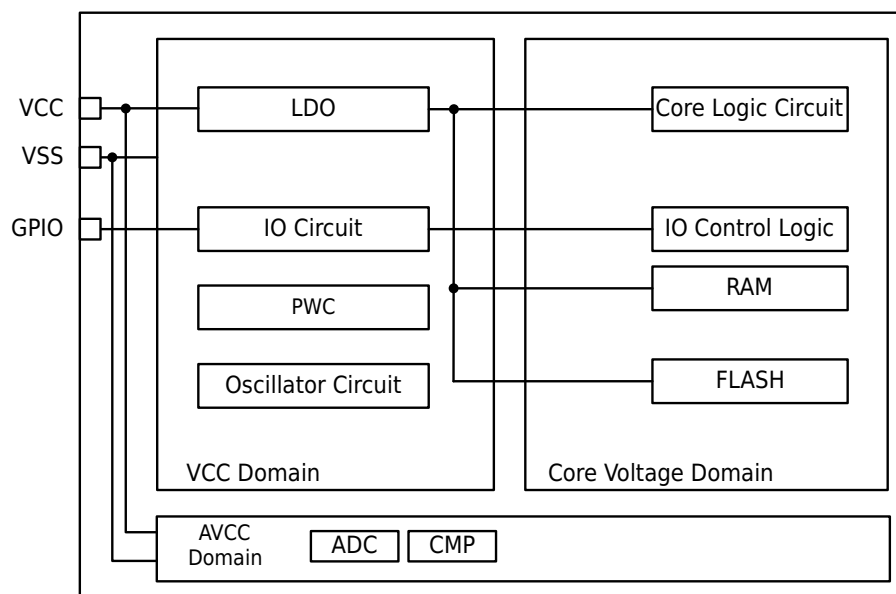


Figure 6-1 Power Supply Configuration Diagram

6.2.2 Power Voltage Detection Unit (LVD) Description

The power voltage detection unit (LVD) includes power on reset (POR), power down reset (PDR), low voltage detection 1 (LVD1), low voltage detection 2 (LVD2).

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 VCC 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 VCC 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.

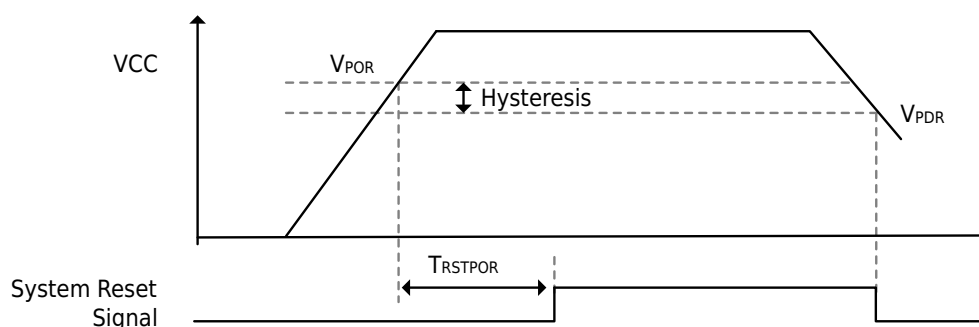


Figure 6-2 Power-on Reset, Power-off Reset Waveforms

6.2.2.2 Low Voltage Detection (LVD1)

The low voltage detection (LVD1) threshold is configured via LVD1ICGCR.LVD1LVL/LVD1DIS bits. The detection voltage of LVD1 can be selected from four thresholds. The low voltage detection circuit selects to generate reset or non-maskable interrupt function through LVD1ICGCR.LVD1IRDIS/LVD1IRS bits.

When the power supply voltage passes through the threshold voltage points of each detection circuit, this event can be programmed and configured as reset/non-maskable interrupt. The main features of low-voltage detection 1 are shown in the following table.

Table 6-1 LVD1 Features

Item	LVD
Test Object	Whether VCC passes the threshold voltage point (LVD1VLVD) during the falling/rising process Or whether the external comparison voltage passes the internal reference voltage
Detection Voltage Point	Configure LVD1ICGCR.LVD1LVL
Reset	Reset: $VCC < LVD1VLVD$; Reset release: $VCC > LVD1VLVD$ after a certain reset processing time.
Non-maskable Interrupt	LVD1ICGCR.LVD1NMIS bit is set to 0b1 for low voltage detection
	VCC passes through the threshold voltage point (LVD1VLVD)
Filter Function	Digital filtering

By configuring LVD1ICGCR.LVD1DIS and LVD1LVL bits, LVD1 effective and threshold can be selected.

Table 6-2 LVD1 Voltage Selection

LVD1ICGCR.LVDDIS	LVD1ICGCR.LVD1LVL	Description
0b1	-	LVD1 Inactive
0b0	0b00	LVD1 Active, Select LVD1 Threshold 0 (LVD1VLVD0)
0b0	0b01	LVD1 Active , Select LVD1 Threshold 1 (LVD1VLVD1)
0b0	0b10	LVD1 Active , Select LVD1 Threshold 2 (LVD1VLVD2)
0b0	0b11	LVD1 Active , Select LVD1 Threshold 3 (LVD1VLVD3)

6.2.2.3 LVD1 Non-maskable Interrupt/Reset Block Diagram

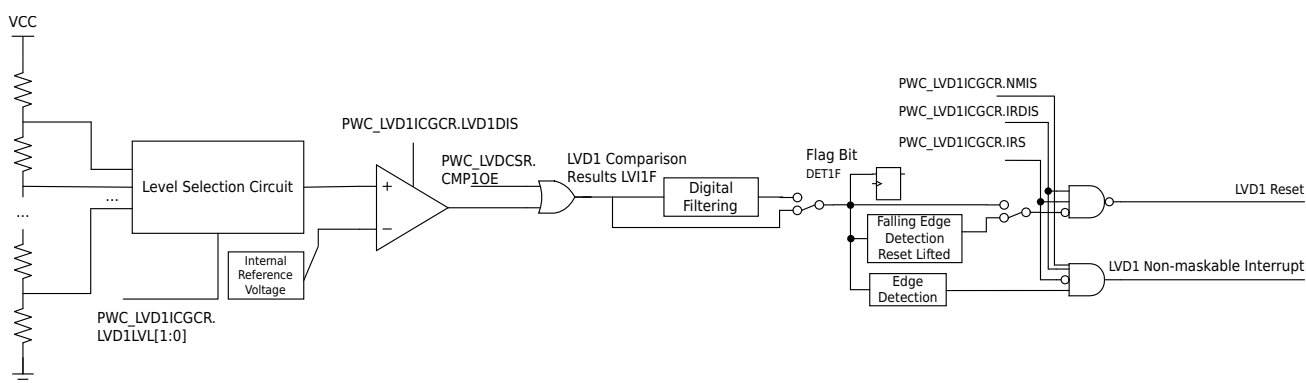


Figure 6-3 LVD1 Non-maskable Interrupt/Reset Block Diagram

6.2.2.4 LVD1 Non-maskable Interrupt/Reset

When using the LVD1 circuit in Stop mode, the digital filter must be invalid.

The following figure is the operation sequence diagram of the LVD1 non-maskable interrupt. DET1F needs to be clear to 0 before an non-maskable interrupt can occur again.

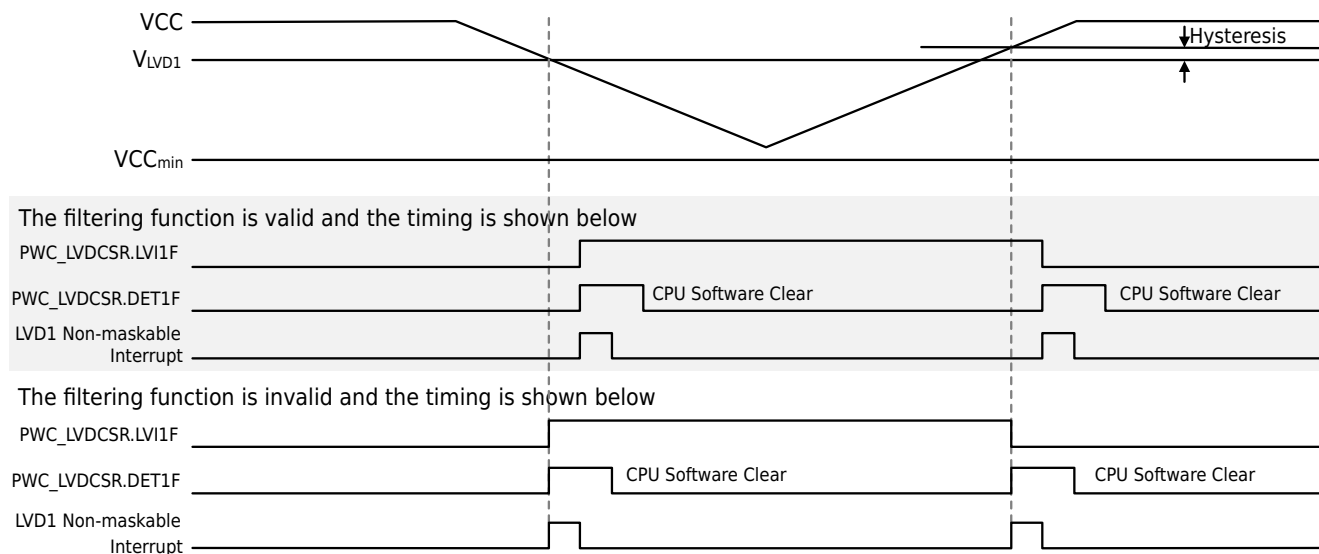


Figure 6-4 LVD1 Non-maskable Interrupt Timing Diagram

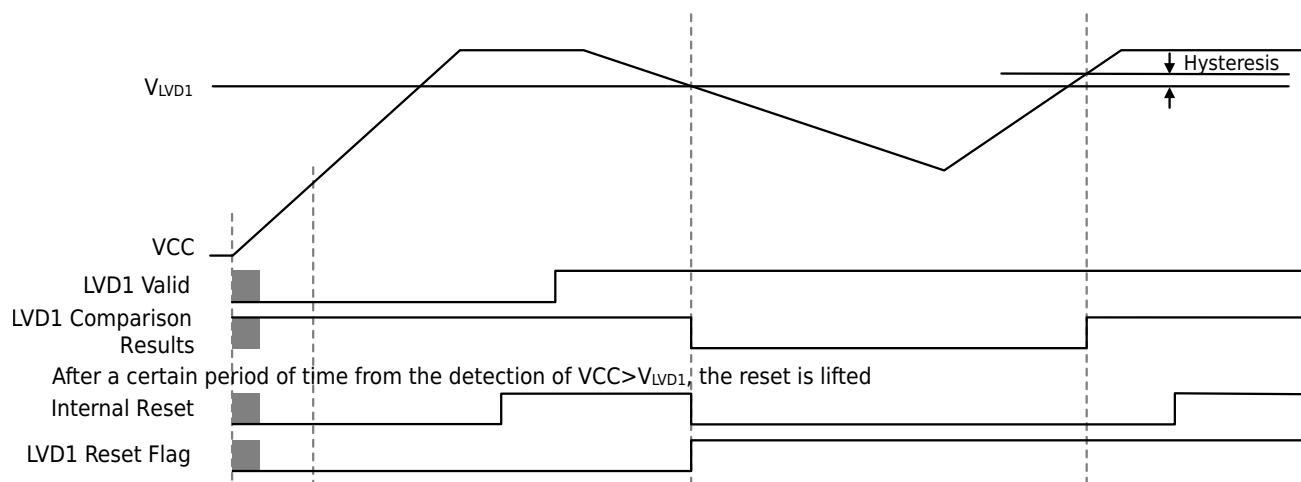


Figure 6-5 LVD1 Reset Timing Diagram

6.2.2.5 Low Voltage Detection (LVD2)

The low voltage detection (LVD2) threshold is configured through the LVD2CR.LVD2LVL/LVD2DIS bits. The detection voltage of LVD2 can be selected from 12 thresholds. The low voltage detection circuit selects to generate reset or non-maskable interrupt function through LVD2CR.LVD2IRDIS/LVD2IRS bits.

When the power supply voltage passes through the threshold voltage points of each detection circuit, this event can be programmed and configured as reset/non-maskable interrupt. The main features of low-voltage detection 2 are shown in the following table.

Table 6-3 LVD2 Features

Item	LVD
Test Object	Whether VCC passes the threshold voltage point (LVD2VLVD) during the falling/rising process Or whether the external comparison voltage passes the internal reference voltage
Detection Voltage Point	Configure LVD2CR.LVD2LVL
Reset	Reset: $VCC < LVD2VLVD$; Reset release: $VCC > LVD2VLVD$ after a certain reset processing time.
Non-maskable Interrupt	LVD2CR.LVD2NMIS bit is set to 0b1 for low voltage detection VCC passes through the threshold voltage point (LVD2VLVD)
Filter Function	Digital filtering

By configuring LVD2CR.LVD2DIS/LVD2LVL bits, LVD2 effective and threshold can be selected.

Table 6-4 LVD2 Voltage Selection

LVD2CR.LVD2DIS	LVD2CR.LVD2LVL	Description
0b1	-	LVD2 Inactive
0b0	0b0000	LVD2 Active, Select LVD2 Threshold 0 (LVD2VLVD0)
0b0	0b0001	LVD2 Active, Select LVD2 Threshold 1 (LVD2VLVD1)
0b0	0b0010	LVD2 Active, Select LVD2 Threshold 2 (LVD2VLVD2)

LVD2CR.LVD2DIS	LVD2CR.LVD2LVL	Description
0b0	0b0011	LVD2 Active, Select LVD2 Threshold 3 (LVD2VLVD3)
0b0	0b0100	LVD2 Active, Select LVD2 Threshold 4 (LVD2VLVD4)
0b0	0b0101	LVD2 Active, Select LVD2 Threshold 5 (LVD2VLVD5)
0b0	0b0110	Setting Prohibited
0b0	0b0111	Setting Prohibited
0b0	0b1000	Setting Prohibited
0b0	0b1001	Setting Prohibited
0b0	0b1010	Setting Prohibited
0b0	0b1011	Setting Prohibited
0b0	0b1100	Setting Prohibited
0b0	0b1101	Setting Prohibited
0b0	0b1110	External Comparison Voltage Input
0b0	0b1111	Setting Prohibited

6.2.2.6 LVD2 Non-maskable Interrupt/Reset Block Diagram

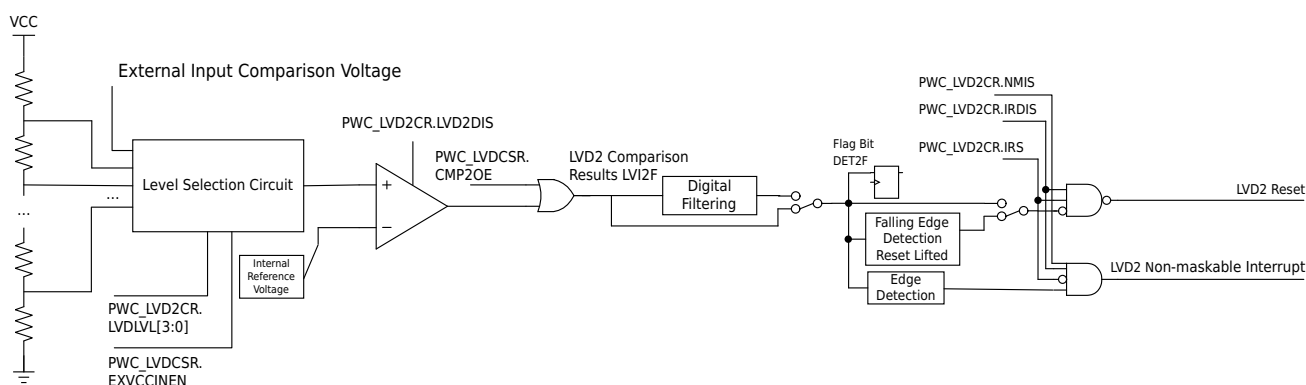


Figure 6-6 LVD2 Non-maskable Interrupt/Reset Block Diagram

6.2.2.7 LVD2 Non-maskable Interrupt/Reset

When using the LVD2 circuit in Stop mode, the digital filter must be invalid.

The following figure is the operation sequence diagram of the LVD2 non-maskable interrupt, DET2F needs to be clear to 0 before a non-maskable interrupt can occur again.

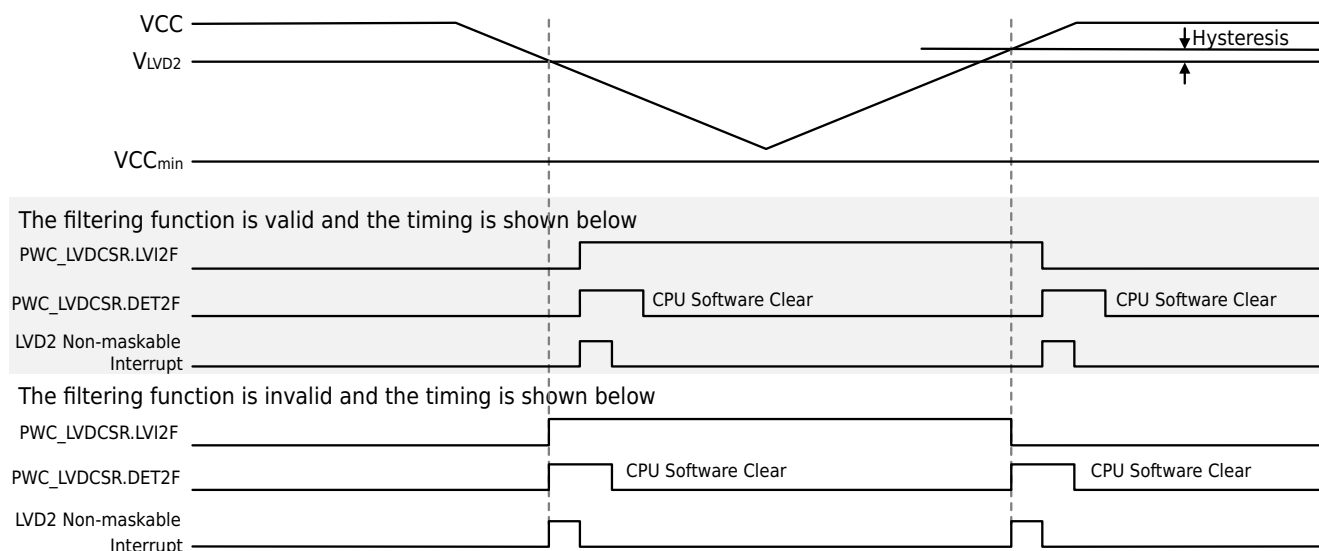


Figure 6-7 LVD2 Non-maskable Interrupt Timing Diagram

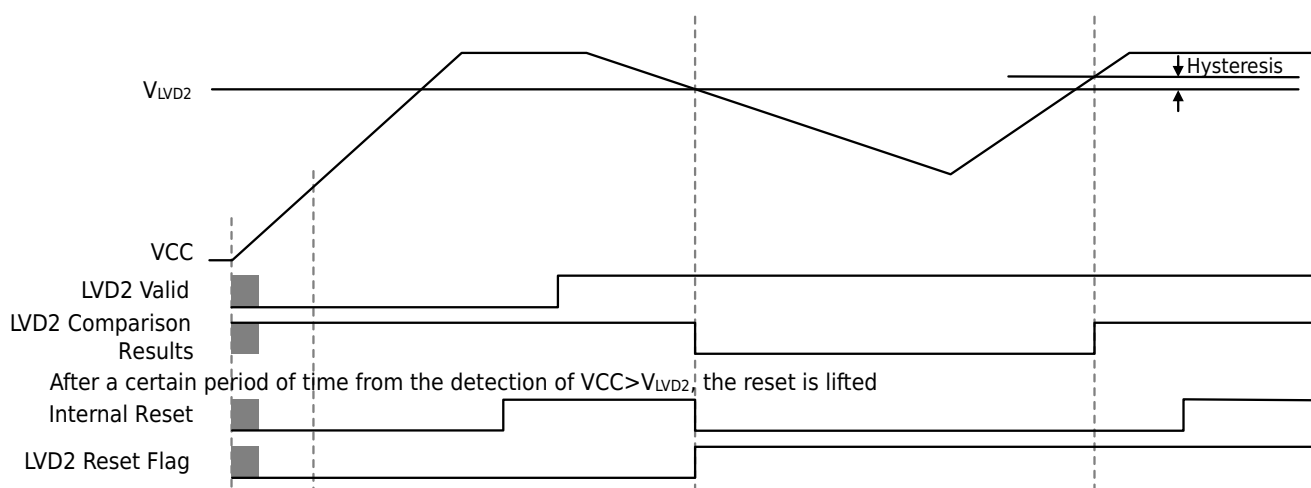


Figure 6-8 LVD2 Reset Timing Diagram

6.2.2.8 Input/Output Pin

Pin Name	Input/Output	Function
LVD2INP	Input	External input LVD2 comparison voltage

6.2.3 Action Mode and Low Power Mode

6.2.3.1 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 run mode. In run mode, the CPU provides the clock through HCLK and executes program code.

In order to save the power consumption of the CPU when it does not need to run, the system provides two low-power modes, such as Sleep mode and Stop mode. In Sleep mode, the Cortex-M0+ core of the chip stops operating, and the peripherals remain running. In Stop mode, both peripherals and CPU of the chip stop operating.

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.

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
Entry	PWC_STPMCR.STOP=0 WFI	PWC_STPMCR.STOP=1 WFI
Release	To interrupt or reset arbitrarily	Interrupt or reset that can be used in Stop mode
External high-speed oscillator	Optional, can be enabled by software	Stop
External low-speed oscillator	Optional, can be enabled by software	Optional, can be enabled by software
Internal high-speed oscillator	Optional, can be enabled by software	Stop
Internal low-speed oscillator	Optional, can be enabled by software	Optional, can be enabled by software
CPU	Stop (hold)	Stop (hold)
RAM	Optional, can be enabled by software	Stop (hold)
Flash	Optional, can be enabled by software	Stop (hold)
Voltage regulator	Work	Work
Power-on reset circuit	Work	Work
Low voltage detection module LVD	Optional, can be enabled by software	Optional, can be enabled by software
SWDT	Optional, can be enabled by software	Optional, can be enabled by software
RTC	Optional, can be enabled by software	Optional, can be enabled by software
Timer0	Optional, can be enabled by software	Optional, can be enabled by software
Timer2	Optional, can be enabled by software	Optional, can be enabled by software
Other peripheral modules	Optional, can be enabled by software	Stop (hold)
ADC	Optional, can be enabled by software	Stop
CMP	Optional, can be enabled by software	Optional, can be enabled by software
PD16, PD15	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
PB7, PB6	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 set as GPIO or other peripheral functions, keep the status before STOP
NRST pin	Work The built-in pull-up circuit is effective. The outside of the chip can also be pulled up to the VCC through the resistor	Work The built-in pull-up circuit is effective. The outside of the chip can also be pulled up to the VCC through the resistor
PA4, PC4	Optional, can be enabled by software When used as the SWD function, the built-in pull-up circuit is effective	Hold When used as the SWD function, the built-in pull-up circuit is effective
Other GPIO	Optional, can be enabled by software	Hold

6.2.3.2 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 dedicated watchdog does not change.

When set to auto-start through ICG, if the initialization configuration register ICGA0x.SWDTSLPOFF bit is 1, the watchdog stops counting in Sleep mode; if the SWDTSLPOFF bit is 0, the watchdog continues counting in Sleep mode. When the ICG is not set to start automatically, start the watchdog via software startup, if the SWDT_CR.SLPOFF bit is 1, the watchdog stops counting in Sleep mode; if the SWDT_CR.SLPOFF bit is 0, the watchdog does not stop 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.

6.2.3.3 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 ICGA0x.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.

In Stop mode, the ADC also consumes power unless it is disabled before entering Stop mode. To disable the ADC, clear the ADC_STR.STRT bit, ensure the stabilization time for the A/D converter analog module to stop, and then execute the WFI instruction to enter Stop mode.

When waking from STOP mode, the clock after wake-up is selected via the STPMCR.CKSHRC bit. When CKSHRC=1, the system clock source after wake-up is selected as HRC; when CKSHRC=0, the system clock after wake-up remains the same as the clock source before entering STOP.

Setting PWC_STPMCR.HAPORDIS to 1 before entering STOP mode disables the high-precision POR circuit after entering STOP mode, which can reduce power consumption in STOP mode. However, the POR monitoring voltage cannot be guaranteed in this case, and the system must ensure the power supply voltage is stable.

- Enter Stop mode

Step1. Clear PWC_LPMCSR.LPMF;

Step2. When PWC_STPMCR.STOP=0b1, 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, LVD1 reset, LVD2 reset and dedicated watchdog reset. Interrupt events that can be used to release stop mode: Pin EIRQ0-7 interrupt, watchdog underflow interrupt, real-time clock periodic interrupt, real-time clock alarm interrupt, CMP interrupt, Timer0 compare match interrupt, Timer2 compare match interrupt, Timer2 overflow interrupt, USART_4_RX line stop mode wake-up interrupt, USART_3_RX line stop mode wake-up interrupt, SPI_1_SSLO line stop mode wake-up 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 exiting Stop mode via an interrupt, it is necessary to set the enable bit for the non-maskable interrupt in the INTC_NMIER register or the corresponding wake-up enable bit in the INTC_WUPEN register. For detailed information, please refer to the relevant description in the "Interrupt Controller (INTC)" chapter.

6.2.4 Methods to Reduce Power Consumption

The following methods can be used to optimize power consumption in Active Mode:

- Reduce the system clock speed
- Turn off unused clock sources
- Configure the CMU_FCG register to disable unused functions

6.2.4.1 Reduce System Clock Speed

In run mode, the speed of system clock (HCLK) 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)".

6.2.4.2 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)
- Internal high-speed oscillator (HRC)
- Internal low-speed oscillator (LRC)

The RTC can select the external low-speed oscillator as the clock source.

For each clock source, it can be individually disabled when not in use to reduce system power consumption. The HRC is equipped with an independent power circuit. After the HRC is disabled, clearing the PWC_PWRC.VHRCE bit can cut off the power supply to the HRC, further reducing power consumption. After cutting off the HRC power supply by clearing the PWC_PWRC.VHRCE bit, restarting the HRC requires setting the PWC_PWRC.VHRCE bit and waiting for 30 μ s.

6.2.4.3 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. For detailed information about the CMU_FCG register, please refer to "Clock Module Unit (CMU)".

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 6-7 Register Protection List

Protected Register Bit	Protected Register
PWC_FPRC.CKRWE	CMU_PERICKSEL, CMU_XTALSTDSR, CMU_SCKDIVR, CMU_CKSWR, CMU_XTALCR, CMU_XTALCFGR, CMU_XTALSTBCR, CMU_HRCCR, CMU_OSCSTBSR, CMU_MCO1CFGR, CMU_XTALSTDCR, CMU_FCG, CMU_XTAL32CR, CMU_XTAL32CFGR, CMU_XTAL32NFR, CMU_LRCCR
PWC_FPRC.PWRWE	PWC_STPMCR, PWC_PWRC, PWC_PWRMON, PWC_RAMCR, PWC_DBGC, RMU_RSTF0
PWC_FPRC.LVRWE	PWC_LVDCSR, PWC_LVD1ICGCR, PWC_LVD2CR, PWC_LPMCSCR

6.3.1 Register List

PWC Base Address 1: 0x40014000

Table 6-8 Power Control (PWC) Register Offset

Offset	Register	Description	Width
0x00	PWC_STPMCR	Stop Mode Control Register	8
0x04	PWC_PWRC	Power Mode Control Register	8
0x08	PWC_PWRMON	Power Monitoring Register	8
0x0C	PWC_RAMCR	RAM Control Register	8
0x10	PWC_LVDCSR	LVD Control Status Register	8
0x14	PWC_LVD2CR	LVD2 Control Register	16
0x18	PWC_LPMCSCR	Low Power Mode Status Register	16
0x1A	PWC_LPMCSCR	Low Power Mode Status Clear Register	16
0x20	PWC_FPRC	Function Protection Control Register	16
0x40	PWC_DBGC	Debug Mode Control Register	8

PWC Base Address 2: 0x40000A84

Table 6-9 Power Control (PWC) Register Offset

Offset	Register	Description	Width
0x00	PWC_LVD1ICGCR	LVD1 ICG Control Register	16

6.3.2 Stop Mode Control Register (PWC_STPMCR)

Offset	Bit Position							
0x00	7	6	5	4	3	2	1	0
Reset	0x00							
Name	STOP	Reserved			HAPORDIS	Reserved	CKSHRC	Reserved
Access	RW				RW		RW	

Bit	Name	Function	Description	Access
7	STOP	STOP Mode Control Bit	0b0: CPU WFI enters SLEEP 0b1: CPU WFI enters STOP	RW
6:4	Reserved	Reserved	-	-
3	HAPORDIS	High Precision POR Disable Control	0b0: High precision POR is valid in STOP mode 0b1: High precision POR is invalid in STOP mode	RW
2	Reserved	Reserved	-	-
1	CKSHRC	System Clock Switch Control after STOP Wake-up	0b0: Do not switch system clock 0b1: Switch to HRC	RW
0	Reserved	Reserved	-	-

6.3.3 Power Mode Control Register (PWC_PWRC)

Offset	Bit Position							
0x04	7	6	5	4	3	2	1	0
Reset	0xFF							
Name	HRCPWRDY	VHRCE	Reserved					
Access	RO	RW						

Bit	Name	Function	Description	Access
7	HRCPWRDY	HRC LDO Status	0b0: HRC LDO not ready 0b1: HRC LDO ready for use  Notice The HRC must wait 30μs after this bit becomes 1 before starting.	RO

Bit	Name	Function	Description	Access
6	VHRCE	HRC LDO Enable Control	● 0b0: HRC LDO powered down ● 0b1: HRC LDO not powered down	RW
5:0	<i>Reserved</i>	<i>Reserved</i>	<i>All bits read as 1, Write as 1</i>	-

6.3.4 Power Monitoring Register (PWC_PWRMON)

Offset	Bit Position							
0x08	7	6	5	4	3	2	1	0
Reset	0x00							
Name	Reserved						PWMONE	Reserved
Access							RW	

Bit	Name	Function	Description	Access
7:2	Reserved	Reserved	-	-
1	PWMONE	Power Supply Internal Voltage Monitoring Enable	0b0: Disable 0b1: Enable  Note When using the ADC to sample the chip's internal voltage, this bit must be set to 1.	RW
0	Reserved	Reserved	-	-

6.3.5 RAM Control Register (PWC_RAMCR)

Offset	Bit Position							
0x0C	7	6	5	4	3	2	1	0
Reset	0x00							
Name	Reserved						RPRTA	
Access							RW	

Bit	Name	Function	Description	Access
7:2	Reserved	Reserved	-	-
1:0	RPRTA	RAM Access Protection Region	0b00: Invalid, RAM can be written 0b01: 128 bytes from the RAM start address 0b10: 256 bytes from the RAM start address 0b11: 512 bytes from the RAM start address	RW

6.3.6 Function Protection Control Register (PWC_FPRC)

Offset	Bit Position															
0x20	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00															
Name	PWC_FPRCWE								FPRCB7	FPRCB6	FPRCB5	FPRCB4	LVRWE	FPRCB2	PWRWE	CKRWE
Access	RW								RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	Function	Description	Access
15:8	PWC_FPRCWE	PWC_FPRC Register Write Enable	When writing the lower 8 bits, 0xA5 must be written to PWC_FPRCWE simultaneously; reads always return 0.	RW
7	FPRCB7	FPRC Bit 7	Reserved. Read as 0, Write as 0.	RW
6	FPRCB6	FPRC Bit 6	Reserved. Read as 0, Write as 0.	RW
5	FPRCB5	FPRC Bit 5	Reserved. Read as 0, Write as 0.	RW
4	FPRCB4	FPRC Bit 4	Reserved. Read as 0, Write as 0.	RW
3	LVRWE	FPRC Bit 3	LVD-related registers and PWC_LPMCSCR register write enable. For protected objects, refer to " Register Protection List ": 0b0: Write protected 0b1: Write enabled	RW
2	FPRCB2	FPRC Bit 2	Reserved. Reads as 0, must be written as 0.	RW
1	PWRWE	FPRC Bit 1	Power control related registers write enable. For protected objects, refer to " Register Protection List ": 0b0: Write protected 0b1: Write enabled	RW
0	CKRWE	FPRC Bit 0	Clock control related registers write enable. For protected objects, refer to " Register Protection List ": 0b0: Write protected 0b1: Write enabled	RW

6.3.7 LVD1 ICG Control Register (PWC_LVD1ICGCR)

Offset	Bit Position															
0x00	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	Determined by the ICGA1x[15:0] value															
Name	LVD1DIS	LVD1IRDIS	LVD1IRS	LVD1NMIS	Reserved	LVD1LVL	Reserved						LVD1DFDIS	LVD1DFS		
Access	RW	RW	RW	RW		RW							RW	RW		

Bit	Name	Function	Description	Access																				
15	LVD1DIS	Low Voltage Detection Enable	0b0: Low voltage detection circuit is enabled 0b1: Low voltage detection circuit is disabled	RW																				
14	LVD1IRDIS	LVD1 Interrupt and Reset Disable Bit	0b0: Interrupt and reset are enabled 0b1: Interrupt and reset are disabled	RW																				
13	LVD1IRS	LVD1 Action Mode Selection	0b0: Non-Maskable Interrupt mode 0b1: Reset mode	RW																				
12	LVD1NMIS	LVD1 Interrupt Type Selection	When LVD1IRS is set to interrupt mode, this bit must be set to 1.	RW																				
11:10	Reserved	Reserved	-	-																				
9:8	LVD1LVL	LVD1 Threshold Voltage Selection	Select the LVD1 threshold voltage (LVD1VLVD0-LVD1VLVD3) <table><tr><th>LV1DLVL</th><th>Threshold Se- lection</th><th>Falling Voltage</th><th>Rising Voltage</th></tr><tr><td>0b00</td><td>LVD1VLVD0</td><td>4.30V</td><td>4.45V</td></tr><tr><td>0b01</td><td>LVD1VLVD1</td><td>3.85V</td><td>3.96V</td></tr><tr><td>0b10</td><td>LVD1VLVD2</td><td>3.17V</td><td>3.27V</td></tr><tr><td>0b11</td><td>LVD1VLVD3</td><td>2.87V</td><td>2.96V</td></tr></table> The falling voltage and rising voltage are typical values. For detailed data, refer to the "Electrical Specifications" section in the "Datasheet".	LV1DLVL	Threshold Se- lection	Falling Voltage	Rising Voltage	0b00	LVD1VLVD0	4.30V	4.45V	0b01	LVD1VLVD1	3.85V	3.96V	0b10	LVD1VLVD2	3.17V	3.27V	0b11	LVD1VLVD3	2.87V	2.96V	RW
LV1DLVL	Threshold Se- lection	Falling Voltage	Rising Voltage																					
0b00	LVD1VLVD0	4.30V	4.45V																					
0b01	LVD1VLVD1	3.85V	3.96V																					
0b10	LVD1VLVD2	3.17V	3.27V																					
0b11	LVD1VLVD3	2.87V	2.96V																					
7:3	Reserved	Reserved	-	-																				
2	LVD1DFDIS	LVD1 Digital Filter Disable	0b0: Digital filter is effective 0b1: Digital filter is ineffective	RW																				
1:0	LVD1DFS	LVD1 Digital Filter Sampling Capability Selection	0b00: 2 LRC cycles 0b01: 4 LRC cycles 0b10: 8 LRC cycles 0b11: 16 LRC cycles	RW																				

6.3.8 LVD2 Control Register (PWC_LVD2CR)

Offset	Bit Position															
0x14	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x8000															
Name	LVD2DIS	LVD2IRDIS	LVD2IRS	LVD2NMIS	LVD2LVL					Reserved				LVD2DFDIS	LVD2DFS	
Access	RW	RW	RW	RW	RW									RW	RW	

Bit	Name	Function	Description	Access
15	LVD2DIS	Low Voltage Detection Enable	0b0: Low voltage detection circuit is enabled 0b1: Low voltage detection circuit is disable	RW
14	LVD2IRDIS	LVD Interrupt and Reset Disable Bit	0b0: Interrupt and reset are enabled 0b1: Interrupt and reset are disable	RW
13	LVD2IRS	LVD Action Mode Selection	0b0: Non-Maskable Interrupt mode 0b1: Reset mode	RW
12	LVD2NMIS	LVD Interrupt Type Selection	When LVD2IRS is set to interrupt mode, this bit must be set to 1.	RW

Bit	Name	Function	Description	Access																																								
11:8	LVD2LVL	LVD2 Threshold Voltage Selection	Select the LVD2 threshold voltage (LVD2VLVD0-LVD2VLVD5) or the LVDINP input																																									
			<table><tr><th>LVD2LVL</th><th>Threshold Selection</th><th>Falling Voltage</th><th>Rising Voltage</th></tr><tr><td>0b0000</td><td>LVD2VLVD0</td><td>3.96V</td><td>4.08V</td></tr><tr><td>0b0001</td><td>LVD2VLVD1</td><td>3.67V</td><td>3.77V</td></tr><tr><td>0b0010</td><td>LVD2VLVD2</td><td>3.07V</td><td>3.17V</td></tr><tr><td>0b0011</td><td>LVD2VLVD3</td><td>2.96V</td><td>3.05V</td></tr><tr><td>0b0100</td><td>LVD2VLVD4</td><td>2.87V</td><td>2.96V</td></tr><tr><td>0b0101</td><td>LVD2VLVD5</td><td>2.75V</td><td>2.83V</td></tr><tr><td>0b0110-0b1101</td><td>-</td><td>-</td><td>-</td></tr><tr><td>0b1110^{*1}</td><td>-</td><td>-</td><td>-</td></tr><tr><td>0b1111</td><td>-</td><td>-</td><td>-</td></tr></table>		LVD2LVL	Threshold Selection	Falling Voltage	Rising Voltage	0b0000	LVD2VLVD0	3.96V	4.08V	0b0001	LVD2VLVD1	3.67V	3.77V	0b0010	LVD2VLVD2	3.07V	3.17V	0b0011	LVD2VLVD3	2.96V	3.05V	0b0100	LVD2VLVD4	2.87V	2.96V	0b0101	LVD2VLVD5	2.75V	2.83V	0b0110-0b1101	-	-	-	0b1110 ^{*1}	-	-	-	0b1111	-	-	-
			LVD2LVL		Threshold Selection	Falling Voltage	Rising Voltage																																					
			0b0000		LVD2VLVD0	3.96V	4.08V																																					
			0b0001		LVD2VLVD1	3.67V	3.77V																																					
			0b0010		LVD2VLVD2	3.07V	3.17V																																					
			0b0011		LVD2VLVD3	2.96V	3.05V																																					
			0b0100		LVD2VLVD4	2.87V	2.96V																																					
			0b0101		LVD2VLVD5	2.75V	2.83V																																					
			0b0110-0b1101		-	-	-																																					
			0b1110 ^{*1}		-	-	-																																					
0b1111	-	-	-																																									
 Note *1: When LVD2LVL=0b1110 and PWC_LVDCSR.EX-VCCINEN=1, select the external voltage (LVD2INP) input. The falling and rising voltages are typical values. For detailed data, refer to the "Electrical Specifications" section in the "Datasheet".																																												
7:3	Reserved	Reserved	-	-																																								
2	LVD2DFDIS	LVD Digital Filter Disable	● 0b0: Digital filter is enabled ● 0b1: Digital filter is disable	RW																																								
1:0	LVD2DFS	LVD Digital Filter Sampling Capability Selection	● 0b00: 2 LRC cycles ● 0b01: 4 LRC cycles ● 0b10: 8 LRC cycles ● 0b11: 16 LRC cycles  Notice After the LVD2DFDIS bit is set to 1, wait for 100μs before modifying this bit.	RW																																								

6.3.9 LVD Control and Status Register (PWC_LVDCSR)

Offset	Bit Position							
0x10	7	6	5	4	3	2	1	0
Reset	0x00							
Name	CMP2OE	CMP1OE	Reserved	DET2F	LVI2F	DET1F	LVI1F	EXVCCINEN
Access	RW	RW		RW	RO	RW	RO	RW

Bit	Name	Function	Description	Access
7	CMP2OE	LVD2 Comparison Result Output Control	0b0: LVD2 comparison result output disable 0b1: LVD2 comparison result output enable  Note This bit is fixed to 0 when the PWC_LVD2CR.LVD2DIS bit is 1.	RW
6	CMP1OE	LVD1 Comparison Result Output Control	0b0: LVD1 comparison result output disable 0b1: LVD1 comparison result output enable  Note This bit is fixed to 0 when the PWC_LVD1ICGCR.LVD1DIS bit is 1.	RW
5	<i>Reserved</i>	<i>Reserved</i>	-	-
4	DET2F	LVD2 Detection Flag Bit	0b0: LVD2 has not detected VCC crossing VLVD2 or LVD2INP has not crossed the internal reference voltage 0b1: LVD2 has detected VCC crossing VLVD2 or LVD2INP has crossed the internal reference voltage  Note This bit is valid when the PWC_LVD2CR.LVD2DIS bit is 0. When the PWC_LVD2CR.LVD2DIS bit is 1, this bit is cleared. This bit can only be written as 0. When this bit is 1, writing 0 clears it.	RW
3	LVI2F	LVD2 Low Voltage Detection Flag	0b0: VCC > VLVD2 or LVD2 closed 0b1: VCC < VLVD2	RO

Bit	Name	Function	Description	Access
2	DET1F	LVD1 Detection Flag Bit	0b0: LVD1 has not detected VCC crossing VLVD1 0b1: LVD1 has detected VCC crossing VLVD1  Note This bit is valid when the PWC_LVD1ICGCR.LVD1DIS bit is 0. When the PWC_LVD1ICGCR.LVD1DIS bit is 1, this bit is cleared. This bit can only be written as 0. When this bit is 1, writing 0 clears it.	RW
1	LV1F	LVD1 Low Voltage Detection Flag	0b0: VCC>VLVD1 or LVD1 closed 0b1: VCC<VLVD1	RO
0	EXVCCINEN	External Input Control Bit	0b0: External input disable 0b1: External input enable  Notice When using external input mode, PWC_LVD2CR.LVD2LVL must be set to 0b1110.	RW

6.3.10 Low Power Mode Status Register (PWC_LPMCSR)

Offset	Bit Position															
0x18	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	0b0: The chip has not entered Stop mode 0b1: The chip has entered Stop mode	RO

6.3.11 Low Power Mode Status Clear Register (PWC_LPMCSR)

Offset	Bit Position															
0x1A	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

6.3.12 Debug Mode Control Register (PWC_DBGC)

Offset	Bit Position							
0x40	7	6	5	4	3	2	1	0
Reset	0x00							
Name	Reserved							DBGEN
Access								RW

Bit	Name	Function	Description	Access
7:1	Reserved	Reserved	-	-
0	DBGEN	Debug Enable	0b0: Debug mode disabled 0b1: Debug mode enabled	RW

7 Initialization Configuration (ICG)

7.1 Introduction

After the chip's reset is released, the hardware circuit will read the data from the ProgramFLASH addresses ranging from 0x01000A00 to 0x01000A4F and from 0x01000B00 to 0x01000B2F, as well as the DataFLASH addresses from 0x40017800 to 0x4001781F. Subsequently, it will load this data into the Initialization Configuration A (ICGA) register and the Initialization Configuration B (ICGB) register. Users can modify the ICGA register by programming, erasing specific sectors, or performing a full erase of the entire ProgramFLASH (256KB). Users can modify the ICGB register by programming or erasing specific sectors. Addresses 0x01000B00-0x01000B2F 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

ICGA base address: 0x01000A00

Table 7-1 ICGA Register Offset Address

Offset	Register	Description	Width
0x0000	ICGA0A	Initialization Configuration Register A0A	32
0x0004	ICGA0B	Initialization Configuration Register A0B	32
0x0008	ICGA0C	Initialization Configuration Register A0C	32
0x0010	ICGA1A	Initialization Configuration Register A1A	32
0x0014	ICGA1B	Initialization Configuration Register A1B	32
0x0018	ICGA1C	Initialization Configuration Register A1C	32
0x0020	ICGA2A	Initialization Configuration Register A2A	32
0x0024	ICGA2B	Initialization Configuration Register A2B	32
0x0028	ICGA2C	Initialization Configuration Register A2C	32
0x0030	ICGA3A	Initialization Configuration Register A3A	32
0x0034	ICGA3B	Initialization Configuration Register A3B	32
0x0038	ICGA3C	Initialization Configuration Register A3C	32
0x0040	ICGA4A	Initialization Configuration Register A4A	32
0x0044	ICGA4B	Initialization Configuration Register A4B	32
0x0048	ICGA4C	Initialization Configuration Register A4C	32

**Note**

The ICGAyA/B/C (where y = 0-4) require the joint use of three registers, and their values must be the same. Please program to write identical data into them.

ICGB base address: 0x40017800

Table 7-2 ICGB Register Offset

Offset	Register	Description	Width
0x0000	ICGB0LLA	Initialization Configuration Register B0LLA	8
0x0001	ICGB0LLB	Initialization Configuration Register B0LLB	8
0x0002	ICGB0LLC	Initialization Configuration Register B0LLC	8
0x0004	ICGB0LHA	Initialization Configuration Register B0LHA	8
0x0005	ICGB0LHB	Initialization Configuration Register B0LHB	8
0x0006	ICGB0LHC	Initialization Configuration Register B0LHC	8
0x0008	ICGB0HLA	Initialization Configuration Register B0HLA	8
0x0009	ICGB0HLB	Initialization Configuration Register B0HLB	8
0x000A	ICGB0HLC	Initialization Configuration Register B0HLC	8
0x000C	ICGB0HHA	Initialization Configuration Register B0HHA	8
0x000D	ICGB0HHB	Initialization Configuration Register B0HHB	8
0x000E	ICGB0HHC	Initialization Configuration Register B0HHC	8
0x0010	ICGB1LLA	Initialization Configuration Register B1LLA	8
0x0011	ICGB1LLB	Initialization Configuration Register B1LLB	8
0x0012	ICGB1LLC	Initialization Configuration Register B1LLC	8
0x0014	ICGB1LHA	Initialization Configuration Register B1LHA	8
0x0015	ICGB1LHB	Initialization Configuration Register B1LHB	8
0x0016	ICGB1LHC	Initialization Configuration Register B1LHC	8
0x0018	ICGB1HLA	Initialization Configuration Register B1HLA	8
0x0019	ICGB1HLB	Initialization Configuration Register B1HLB	8
0x001A	ICGB1HLC	Initialization Configuration Register B1HLC	8
0x001C	ICGB1HHA	Initialization Configuration Register B1HHA	8
0x001D	ICGB1HHB	Initialization Configuration Register B1HHB	8
0x001E	ICGB1HHC	Initialization Configuration Register B1HHC	8

**Note**

The ICGByzA/B/C (where y = 0-1, z = LL/LH/HL/HH) require the combined use of three registers, and their values must be identical. Please program to write the same data into them.

For the ICGB_y (y = 0-1) register, bits [7:0] are set via ICGB_yLLA/B/C, bits [15:8] via ICGB_yLHA/B/C, bits [23:16] via ICGB_yHLA/B/C, and bits [31:24] via ICGB_yHHA/B/C.

7.2.2 Initialization Configuration Register A0 (ICGA0x) (x=A/B/C)

Offset: 0x0000 (ICGA0A), 0x0004 (ICGA0B), 0x0008 (ICGA0C)

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																				SWDTSLOFF	SWDWDPT		SWDTCKS		SWDTPERI		SWDTITS		SWDTAITS		
Access																					RO											RO

Bit	Name	Function	Description	Access
31:13	Reserved	Reserved	-	-
12	SWDTSLOPF	SWDT Counting Stops in Sleep/ Stop Mode	0b0: SWDT counting does not stop in Sleep mode and Stop mode 0b1: SWDT counting stops in Sleep mode and Stop mode	RO
11:8	SWDWDPT	Refresh Allowed Area Count Percentage	Allowed interval for refreshing the SWDT count value: 0b0000: 0%-100% 0b0001: 0%-25% 0b0010: 25%-50% 0b0011: 0%-50% 0b0100: 50%-75% 0b0101: 0%-25%, 50%-75% 0b0110: 25%-75% 0b0111: 0%-75% 0b1000: 75%-100% 0b1001: 0%-25%, 75%-100% 0b1010: 25%-50%, 75%-100% 0b1011: 0%-50%, 75%-100% 0b1100: 50%-100% 0b1101: 0%-25%, 50%-100% 0b1110: 25%-100% 0b1111: 0%-100%	RO
7:4	SWDTCKS	SWDT Count Clock	0b0000: SWDTCLK 0b0100: SWDTCLK/16 0b0101: SWDTCLK/32 0b0110: SWDTCLK/64 0b0111: SWDTCLK/128 0b1000: SWDTCLK/256 0b1011: SWDTCLK/2048 - Others: Reserved	RO

Bit	Name	Function	Description	Access
3:2	SWDTPERI	SWDT Count Overflow Period	● 0b00: 256 periods ● 0b01: 4096 periods ● 0b10: 16384 periods ● 0b11: 65536 periods	RO
1	SWDTITS	SWDT Interrupt Selection	● 0b0: Interrupt request ● 0b1: Reset request	RO
0	SWDTAUTS	SWDT Auto Start	● 0b0: After reset, SWDT auto-starts (hardware start) ● 0b1: After reset, SWDT is in stopped state	RO

7.2.3 Initialization Configuration Register A1 (ICGA1x) (x=A/B/C)

Offset: 0x0010 (ICGA1A), 0x0014 (ICGA1B), 0x0018 (ICGA1C)

Offset	Bit Position															
-	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reset	0xFFFFFFFF															
Name	NMIICGEN	NMIEN	NMITRG	NMIFEN	NMIFCLK	Reserved	NRSTEN	LKUPDIS	Reserved	HRCFREQS	LVD1DIS	LVD1IRDIS	LVD1IRS	LVD1NMIS	Reserved	LVD1LVL
Access	RO	RO	RO	RO	RO		RO	RO		RO	RO	RO	RO	RO		RO

Bit	Name	Function	Description	Access																				
31	NMIICGEN	NMI Pin ICG Setting Enable	● 0b0: NMI pin ICG setting enable ● 0b1: NMI pin ICG setting disable	RO																				
30	NMIEN	NMI Pin Interrupt Enable	● 0b0: Disable NMI pin interrupt ● 0b1: Enable NMI pin interrupt	RO																				
29	NMITRG	NMI Pin Edge Trigger	● 0b0: Falling edge ● 0b1: Rising edge	RO																				
28	NMIFEN	NMI Digital Filter Enable	● 0b0: Disable digital filter function ● 0b1: Enable digital filter function	RO																				
27:26	NMIFCLK	Filter Sampling Period Selection	● 0b00: 1 x HCLK period ● 0b01: 8 x HCLK periods ● 0b10: 32 x HCLK periods ● 0b11: 64 x HCLK periods	RO																				
25	Reserved	Reserved	Read as 1, Write as 1	-																				
24	NRSTEN	NRST Pin Enable	● 0b0: Disable NRST pin function ● 0b1: Enable NRST pin function	RO																				
23	LKUPDIS	LOCKUP Reset Disable Bit	● 0b0: CPU LOCKUP reset is effective ● 0b1: CPU LOCKUP reset is ineffective	RO																				
22:20	Reserved	Reserved	All bits read as 1, Write as 1	-																				
19:16	HRCFREQS	HRC Frequency Selection Bit	<table><tr><th rowspan="2">HRCFREQS[2:0]</th><th colspan="2">HRC Frequency</th></tr><tr><th>HRCFREQS[3]=0</th><th>HRCFREQS[3]=1</th></tr><tr><td>0b000</td><td>64MHz</td><td>48MHz</td></tr><tr><td>0b001</td><td>32MHz</td><td>24MHz</td></tr><tr><td>0b010</td><td>16MHz</td><td>12MHz</td></tr><tr><td>0b011</td><td>8MHz</td><td>6MHz</td></tr><tr><td>0b100</td><td>4MHz</td><td>3MHz</td></tr></table>	HRCFREQS[2:0]	HRC Frequency		HRCFREQS[3]=0	HRCFREQS[3]=1	0b000	64MHz	48MHz	0b001	32MHz	24MHz	0b010	16MHz	12MHz	0b011	8MHz	6MHz	0b100	4MHz	3MHz	RO
HRCFREQS[2:0]	HRC Frequency																							
	HRCFREQS[3]=0	HRCFREQS[3]=1																						
0b000	64MHz	48MHz																						
0b001	32MHz	24MHz																						
0b010	16MHz	12MHz																						
0b011	8MHz	6MHz																						
0b100	4MHz	3MHz																						

Bit	Name	Function	Description			Access																				
19:16	HRCFREQS	HRC Frequency Selection Bit	HRCFREQS[2:0]	HRC Frequency		RO																				
				HRCFREQS[3]=0	HRCFREQS[3]=1																					
			0b101-111	2MHz	1.5MHz																					
15	LVD1DIS	Voltage Detection Disable Bit	0b0: Voltage detection circuit is enabled 0b1: Voltage detection circuit is disabled			RO																				
14	LVD1IRDIS	LVD1 Interrupt and Reset Disable Bit	0b0: Interrupt and reset are enabled 0b1: Interrupt and reset are disabled			RO																				
13	LVD1IRS	LVD1 Action Mode Selection	0b0: Non-Maskable Interrupt mode 0b1: Reset mode			RO																				
12	LVD1NMIS	LVD1 Interrupt Type Selection	When LVD1IRS is set to interrupt mode, this bit must be set to 1.			RO																				
11:10	Reserved	Reserved	All bits read as 1, Write as 1			-																				
9:8	LVD1LVL	LVD1 Threshold Voltage Selection	Select the LVD1 threshold voltage (LVD1VLVD0-LVD1VLVD3) <table><tr><th>LV1DLVL</th><th>Threshold Selection</th><th>Falling Voltage</th><th>Rising Voltage</th></tr><tr><td>0b00</td><td>LVD1VLVD0</td><td>4.30V</td><td>4.45V</td></tr><tr><td>0b01</td><td>LVD1VLVD1</td><td>3.85V</td><td>3.96V</td></tr><tr><td>0b10</td><td>LVD1VLVD2</td><td>3.17V</td><td>3.27V</td></tr><tr><td>0b11</td><td>LVD1VLVD3</td><td>2.87V</td><td>2.96V</td></tr></table> The falling voltage and rising voltage are typical values. For detailed data, refer to the "Electrical Specifications" chapter in the "Datasheet".			LV1DLVL	Threshold Selection	Falling Voltage	Rising Voltage	0b00	LVD1VLVD0	4.30V	4.45V	0b01	LVD1VLVD1	3.85V	3.96V	0b10	LVD1VLVD2	3.17V	3.27V	0b11	LVD1VLVD3	2.87V	2.96V	RO
LV1DLVL	Threshold Selection	Falling Voltage	Rising Voltage																							
0b00	LVD1VLVD0	4.30V	4.45V																							
0b01	LVD1VLVD1	3.85V	3.96V																							
0b10	LVD1VLVD2	3.17V	3.27V																							
0b11	LVD1VLVD3	2.87V	2.96V																							
7:3	Reserved	Reserved	All bits read as 1, Write as 1			-																				
2	LVD1DFDIS	LVD1 Digital Filter Disable	0b0: Digital filter is effective 0b1: Digital filter is ineffective			RO																				
1:0	LVD1DFS	LVD1 Digital Filter Sampling Capability Selection	0b00: 2 LRC periods 0b01: 4 LRC periods 0b10: 8 LRC periods 0b11: 16 LRC periods			RO																				

7.2.4 Initialization Configuration Register A2 (ICGA2x) (x=A/B/C)

Offset: 0x0020(ICGA2A), 0x0024(ICGA2B), 0x0028(ICGA2C)

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	DBUSPRT												Reserved	DBUSPSZ	Reserved												ECCMOD		ECCOAD			
Access	RO													RO													RO	RO				

Bit	Name	Function	Description	Access
31:20	DBUSPRT	D-BUS Read Protection	When DBUSPRT=0x445, D-BUS read protection is enabled for the space starting from 0x00000000. Otherwise, the D-BUS read protection function is disabled.	RO
19:17	Reserved	Reserved	-	-
16	DBUSPSZ	D-BUS Read Protection Space	Valid when DBUSPRT=0x445: 0b0: Address 0x00000000-0x00003FFF (16KB) 0b1: Address 0x00000000-0x0000FFFF (64KB)	RO
15:3	Reserved	Reserved	-	-
2:1	ECCMOD	Program FLASH ECC Check Enable Bit	0b00: Disable ECC check function 0b01: - If 1-bit error, ECC corrects the error, does not generate a 1-bit error flag, and does not generate an interrupt/reset - If 2-bit error, ECC detects the error, generates a 2-bit error flag, and generates an interrupt/reset 0b10: - If 1-bit error, ECC corrects the error, generates a 1-bit error flag, and does not generate an interrupt/reset - If 2-bit error, ECC detects the error, generates a 2-bit error flag, and generates an interrupt/reset 0b11: - If 1-bit error, ECC corrects the error, generates a 1-bit error flag, and generates an interrupt/reset - If 2-bit error, ECC detects the error, generates a 2-bit error flag, and generates an interrupt/reset	RO
0	ECCOAD	Operation after ECC Check Error in Program FLASH	0b0: Non-maskable interrupt 0b1: Reset	RO

7.2.5 Initialization Configuration Register A3 (ICGA3x) (x=A/B/C)

Offset: 0x0030 (ICGA3A), 0x0034 (ICGA3B), 0x0038 (ICGA3C)

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	FNWPRT31	FNWPRT30	FNWPRT29	FNWPRT28	FNWPRT27	FNWPRT26	FNWPRT25	FNWPRT24	FNWPRT23	FNWPRT22	FNWPRT21	FNWPRT20	FNWPRT19	FNWPRT18	FNWPRT17	FNWPRT16	FNWPRT15	FNWPRT14	FNWPRT13	FNWPRT12	FNWPRT11	FNWPRT10	FNWPRT9	FNWPRT8	FNWPRT7	FNWPRT6	FNWPRT5	FNWPRT4	FNWPRT3	FNWPRT2	FNWPRT1	FNWPRT0	
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	FNWPRT31	ProgramFLASH Sector Write Protection Bit 31	0b0: Sector 248-255 programming and sector erase are disabled. 0b1: Sector 248-255 programming and sector erase are enabled.	RO
30:2	...	...	...	RO
1	FNWPRT1	ProgramFLASH Sector Write Protection Bit 1	0b0: Sector 8-15 programming and sector erase are disabled. 0b1: Sector 8-15 programming and sector erase are enabled.	RO
0	FNWPRT0	ProgramFLASH Sector Write Protection Bit 0	0b0: Sector 0-7 programming and sector erase are disabled. 0b1: Sector 0-7 programming and sector erase are enabled.	RO

7.2.6 Initialization Configuration Register A4 (ICGA4x) (x=A/B/C)

Offset: 0x0040 (ICGA4A), 0x0044 (ICGA4B), 0x0048 (ICGA4C)

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	FNWPRT63	FNWPRT62	FNWPRT61	FNWPRT60	FNWPRT59	FNWPRT58	FNWPRT57	FNWPRT56	FNWPRT55	FNWPRT54	FNWPRT53	FNWPRT52	FNWPRT51	FNWPRT50	FNWPRT49	FNWPRT48	FNWPRT47	FNWPRT46	FNWPRT45	FNWPRT44	FNWPRT43	FNWPRT42	FNWPRT41	FNWPRT40	FNWPRT39	FNWPRT38	FNWPRT37	FNWPRT36	FNWPRT35	FNWPRT34	FNWPRT33	FNWPRT32	
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	FNWPRT63	ProgramFLASH Sector Write Protection Bit 63	0b0: Sector 504-511 programming and sector erase are disabled. 0b1: Sector 504-511 programming and sector erase are enabled.	RO

Bit	Name	Function	Description	Access
30:2	...	...	...	RO
1	FNWPRT33	ProgramFLASH Sector Write Protection Bit 33	● 0b0: Sector 264-271 programming and sector erase are disabled. ● 0b1: Sector 264-271 programming and sector erase are enabled.	RO
0	FNWPRT32	ProgramFLASH Sector Write Protection Bit 32	● 0b0: Sector 256-263 programming and sector erase are disabled. ● 0b1: Sector 256-263 programming and sector erase are enabled.	RO

7.2.7 Initialization Configuration Register B0 (ICGB0)

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	FNWPRT31	FNWPRT30	FNWPRT29	FNWPRT28	FNWPRT27	FNWPRT26	FNWPRT25	FNWPRT24	FNWPRT23	FNWPRT22	FNWPRT21	FNWPRT20	FNWPRT19	FNWPRT18	FNWPRT17	FNWPRT16	FNWPRT15	FNWPRT14	FNWPRT13	FNWPRT12	FNWPRT11	FNWPRT10	FNWPRT9	FNWPRT8	FNWPRT7	FNWPRT6	FNWPRT5	FNWPRT4	FNWPRT3	FNWPRT2	FNWPRT1	FNWPRT0	
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	FNWPRT31	DataFLASH Sector Write Protection Bit 31	0b0: Sector 62-63 programming and sector erase are disabled 0b1: Sector 62-63 programming and sector erase are enabled	RO
30:2	...	...	...	RO
1	FNWPRT1	DataFLASH Sector Write Protection Bit 1	0b0: Sector 2-3 programming and sector erase are disabled 0b1: Sector 2-3 programming and sector erase are enabled	RO
0	FNWPRT0	DataFLASH Sector Write Protection Bit 0	0b0: Sector 0-1 programming and sector erase are disabled 0b1: Sector 0-1 programming and sector erase are enabled	RO

7.2.8 Initialization Configuration Register B1 (ICGB1)

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	ENWPRT31	ENWPRT30	ENWPRT29	ENWPRT28	ENWPRT27	ENWPRT26	ENWPRT25	ENWPRT24	ENWPRT23	ENWPRT22	ENWPRT21	ENWPRT20	ENWPRT19	ENWPRT18	ENWPRT17	ENWPRT16	ENWPRT15	ENWPRT14	ENWPRT13	ENWPRT12	ENWPRT11	ENWPRT10	ENWPRT9	ENWPRT8	ENWPRT7	ENWPRT6	ENWPRT5	ENWPRT4	ENWPRT3	ENWPRT2	ENWPRT1	ENWPRT0	
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	ENWPRT31	EDataFLASH Sector Write Protection Bit 31	0b0: Sector 62-63 programming and sector erase are disabled 0b1: Sector 62-63 programming and sector erase are enabled	RO
30:2	...	...	...	RO

Bit	Name	Function	Description	Access
1	ENWPRT1	EDataFLASH Sector Write Protection Bit 1	● 0b0: Sector 2-3 programming and sector erase are disabled ● 0b1: Sector 2-3 programming and sector erase are enabled	RO
0	ENWPRT0	EDataFLASH Sector Write Protection Bit 0	● 0b0: Sector 0-1 programming and sector erase are disabled ● 0b1: Sector 0-1 programming and sector erase are enabled	RO

8 Embedded Flash Memory (EFM)

8.1 Introduction

This product features a maximum of 256KB of ProgramFLASH, 32KB of DataFLASH, and 2KB of EDataFLASH. Programming, sector erase, and full erase operations can be performed on ProgramFLASH, DataFLASH, and EDataFLASH. ProgramFLASH can accelerate code execution through instruction prefetch.

ProgramFLASH is connected to the Flash bus and is primarily used for instruction execution and constant reading. DataFLASH and EDataFLASH are located on the AHB peripheral bus and are commonly used for data access:

- ProgramFLASH: 256KBytes capacity, 72-bit width (including 64-bit data and 8-bit ECC code), programming unit of 8Bytes, sector erase unit of 512Bytes.
- DataFLASH: 32KBytes capacity, 39-bit width (including 32-bit data and 7-bit ECC code), programming unit of 4Bytes, sector erase unit of 512Bytes.
- EDataFLASH: 2KBytes capacity, 39-bit width (including 32-bit data and 7-bit ECC code), programming unit of 4Bytes, sector erase unit of 32Bytes.

8.2 Features

- ProgramFLASH and DataFLASH, EDataFLASH can implement BGO (BackGround Operation) function
- ProgramFLASH and DataFLASH each have an OTP space:
 - ▶ ProgramFLASH-OTP space (OTPA): 8 data areas, 7 of 64Bytes, 1 of 16KBytes
 - ▶ DataFLASH-OTP space (OTPB): 24 data areas, each 16Bytes
- Flash bus 8Bytes prefetching
- Supports boot swap function
- Support data security protection
- Supports MARGIN read mode, can predict Flash read errors in advance
- Provides error injection function for ECC circuit self-test
- D-BUS read protection function
- ProgramFLASH, DataFLASH and EDataFLASH have ECC (Error Checking and Correcting) function, correct 1 and detect 2, can correct 1-bit error and detect 2-bit errors for data

8.3 Functional Description

8.3.1 Embedded Flash

The embedded Flash address allocation is shown in the figure below.

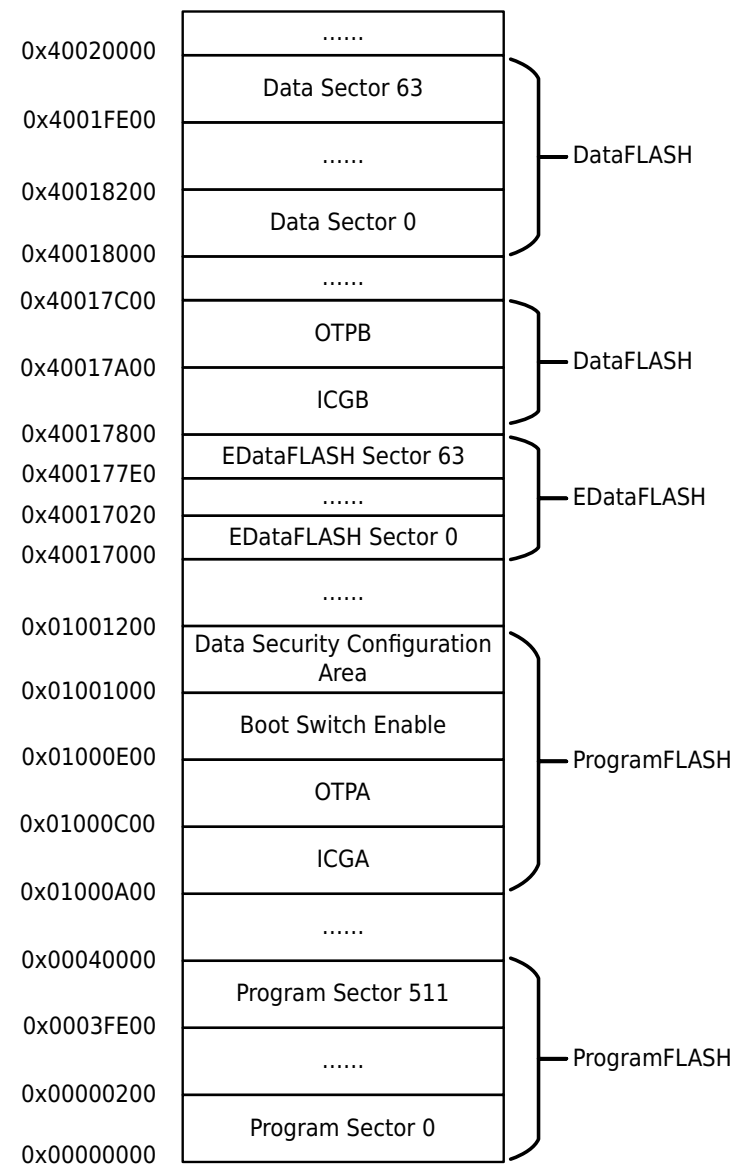


Figure 8-1 Flash Address Distribution

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_(D)FRMC.FLWT) based on the CPU operation frequency.

The bus clock connected to ProgramFLASH, DataFLASH, and EDataFLASH is HCLK. After the chip reset is released, the clock frequency is 64MHz, and the Flash read cycle is 4, meaning 3 read wait cycles are inserted.

In Flash read mode, the maximum single-cycle read frequency is shown in the table below.

Table 8-1 Flash Read Mode and Maximum Read Frequency Comparison Table

Flash Read Mode	Maximum Single Cycle Read Frequency
Non-MARGIN Read Mode	16MHz
MARGIN Read Mode	5MHz

In read mode, the formula for calculating the number of inserted read wait cycles n is as follows:

$$n = \text{INT}\left(\frac{\text{Bus Clock Frequency}}{\text{Maximum Single-Cycle Reading Frequency in Reading Mode}}\right) - 1$$

Where INT indicates that the calculation result is rounded up. For example, when the HCLK frequency is 64MHz, the calculation yields $n=3$ for non-MARGIN read mode; $n=12$ for MARGIN read mode.

The steps for modifying the system frequency and setting the read wait cycle are as follows:

CPU Frequency Increase Steps:

- Step1.** Write the new read wait period setting to the EFM_(D)FRMC.FLWT register.
- Step2.** Read the EFM_(D)FRMC register to check if the new wait period setting is successful.
- Step3.** Increase the CPU clock frequency by setting the CMU_CKSWR.CKSW register or the CMU_SCKDIVR.SCKDIV register.
- Step4.** Read the CMU_CKSWR or CMU_SCKDIVR register to check if the new setting is successful.

CPU Frequency Reduction Steps:

- Step1.** Decrease the CPU clock frequency by setting the CMU_CKSWR.CKSW register or the CMU_SCKDIVR.SCKDIV register.
- Step2.** Read the register CMU_CKSWR or CMU_SCKDIVR to check if the new setting is successful.
- Step3.** Write the new read wait period setting to the EFM_(D)FRMC.FLWT register.
- Step4.** Read the EFM_(D)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_(D)FAPRT, then write 0x3210)
2. Set Blank read mode. (EFM_(D)FRMC.BLKDE=1)
3. Read the Flash address to be checked.
4. Check if the read data contains any "0" flags (EFM_(D)FSR.BLKERR).
If the read data (including ECC data) is not all "1" in the Blank state, EFM_(D)FSR.BLKERR=1.
5. Write 1 to the corresponding flag clear bit in the EFM_(D)FSCLR register to clear the status bit.
6. Set back to Normal read mode.(EFM_(D)FRMC.BLKDE=0)

The above procedure should be executed on a different Flash block from the check address (The program is in ProgramFLASH, and the check address is in DataFLASH or EDataFLASH), or on other memory (e.g., RAM).

8.3.2.4 D-BUS Read Protection Function

The D-BUS read protection function protects the D-BUS read protection space set by ICGA2.DBUSPSZ. To enable D-BUS protection, set ICGA2.DBUSPRT to 0x445, set the space with DBUSPSZ, and reset and restart the system to enable the function. When the CPU-PC pointer is within this D-BUS read protection

space (i.e., the CPU is executing programs in this area), the CPU can access it normally; when the CPU-PC pointer is outside this D-BUS read protection space (i.e., the CPU is executing programs outside the D-BUS read protection space), D-BUS reads by the CPU to the D-BUS read protection space are prohibited, 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.

8.3.3 Flash Read Prefetch

To improve CPU performance, the Flash controller is configured with a read prefetch function. When the CPU is idle, the host bus can use prefetch operations to read the next 64-bit data from Flash. Enable the prefetch function by setting the EFM_FRMC.PREFE bit to 1. When the CPU reads an instruction or data address that matches the prefetch address, the prefetched data is sent to the CPU; if the addresses do not match, the prefetched data is invalidated, and data read from Flash is sent to the CPU.

8.3.4 Flash Programming and Erasing Operations

8.3.4.1 Overview of Programming and Erasing Operations

ProgramFLASH, DataFLASH and EDataFLASH supports programming, sector erasing, and full erasing operations.

Flash programming is divided into three modes: single programming without read-back mode, single programming with read-back mode, and continuous programming mode.

The ProgramFLASH programming unit is 8Bytes, written in two 32-bit writes; the DataFLASH and EDataFLASH programming unit is 4Bytes, written in a single 32-bit write. The sector erasing unit for ProgramFLASH and DataFLASH is 512Bytes, and for EDataFLASH it is 32Bytes. The full erasing unit for ProgramFLASH is 256KBytes, for DataFLASH is 32KBytes, and for EDataFLASH is 2KBytes.

During Flash programming/erasing, if EFM_(D)FWMC.BUSHLDCTL=0 is set, the bus is occupied until the erase/write operation ends; if EFM_(D)FWMC.BUSHLDCTL=1 is set, the bus is released, and the bus can continue to access another Flash address. Before Flash programming/erasing, please disable the prefetch function. The following sections describe the setup steps for programming and erasing operations respectively.

The control and status registers for EDataFLASH programming, sector erasing, and full erasing are consistent with those of DataFLASH.

8.3.4.2 Unlock EFM_(D)KEY1 Register

After reset, the EFM_(D)FWMC register is in write-prohibited status. It is necessary to first disable the EFM_(D)FAPRT register, and then disable the protection of EFM_(D)KEY1. The following steps are used to disable the above protection:

Step1. Disable the Flash register access write protection (EFM_(D)FAPRT first writes 0x0123, then writes 0x3210)

Step2. Unlock EFM_(D)KEY1 (write 0x01234567 to EFM_(D)KEY1 first, then write 0xFEDCBA98)

If an incorrect sequence is written to EFM_(D)KEY1, a bus error will occur, and it will self-lock. The EFM_(D)FWMC register will be write-prohibited until a reset occurs.

After correctly writing the set value to the EFM_(D)FWMC register, if you need to return to the EFM_(D)FWMC write-prohibited status, you can write 1 to EFM_(D)FWMC.KEY1LOCK.

8.3.4.3 Write Protection Function

The Flash write protection function exists. After write protection takes effect, programming and sector erasing operations on protected addresses are unsuccessful and generate an error flag (EFM_(D)FSR.PRTWERR=1). Write protection is invalid for full chip erase. For ProgramFLASH, there is one write protection bit for every 8 sectors. For DataFLASH and EDataFLASH, there is one write protection bit for every 2 sectors.

The write protection setting for ProgramFLASH is in the write protection register EFM_FNWPRTx (x=0-1). The write protection setting for DataFLASH is in the write protection register EFM_DFNWPRT0. The write protection setting for EDataFLASH is in the write protection register EFM_DFNWPRT1. The initial value of register EFM_FNWPRTx (x=0-1) is loaded from the ICGA address (0x01000A30-0x01000A4F). The initial value of register EFM_DFNWPRTx (x=0-1) is loaded from the ICGB address (0x40017800-0x4001781F). After reset startup, the bits in register EFM_(D)FNWPRTx (x=0-1) can only be written as 0, not as 1.

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_(D)FAPRT first, then write 0x3210)
- Step2.** Unlock the EFM_(D)KEY1 lock. (EFM_(D)KEY1 writes 0x01234567 first, then writes 0xFEDCBA98)
- Step3.** Set the single programming mode. (EFM_(D)FWMC.PEMOD=0b001)
- Step4.** Process according to Flash type:
 - ProgramFLASH: Write 64-bit data to the target ProgramFLASH address in two steps.
The first 32-bit write address must end with 0 or 8, and the second 32-bit write address must end with 4 or C.
The first write operation does not trigger Flash programming action. The Flash controller temporarily stores the address and 32-bit data, and the Flash remains readable. The status can be queried via the EFM_FSR.FSTWRST bit. When the second 32-bit write operation occurs, it triggers the Flash programming action, and the hardware clears the EFM_FSR.FSTWRST bit. If the first and second write addresses are not separated by 4bytes, if the first write address does not end with 0 or 8, or if the second write address does not end with 4 or C, then the EFM_FSR.PGSZERR flag is set and the programming action does not occur.
 - DataFLASH/EDataFLASH: Programming DataFLASH and EDataFLASH requires writing 32-bit data to an address aligned to 4 (ending with 0/4/8/C). Non-32-bit writes are invalid for programming, and EFM_DFSR.PGSZERR is set. (The Cortex-M0+ core will trigger a hardware fault exception when attempting a non-4-aligned access.)
- Step5.** Wait for the Flash operation to end and for it to be in an idle state.
 - For ProgramFLASH address programming, wait for EFM_FSR.OPTEND=1, EFM_FSR.RDY=1.
 - For DataFLASH and EDataFLASH address programming, wait for EFM_DFSR.OPTEND=1, EFM_DFSR.RDY=1.
- Step6.** 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.
- Step7.** Clear the end of programming flag. (EFM_(D)FSR.OPTEND)

Performing a single programming write operation on a locked OTP address without read-back will fail, and the flag bit EFM_(D)FSR.OTPWERR will be set.

Performing a single programming write operation without read-back on a write-protected sector will fail, and the flag bit EFM_(D)FSR.PRTWERR 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_(D)FSR.MISMATCH.

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

- Step1.** Disable the Flash register write protection. (EFM_(D)FAPRT writes 0x0123 first, then writes 0x3210)
- Step2.** Unlock the EFM_(D)KEY1 lock. (EFM_(D)KEY1 writes 0x01234567 first, then writes 0xFEDCBA98)
- Step3.** Set single programming with read-back mode. (EFM_(D)FWMC.PEMOD=0b010)
- Step4.** Write 64/32 bits of data to the programming address. The writing method is the same as "[Single Programming Without Read-back Function](#)".
- Step5.** Wait for the Flash operation to end and for it to be in an idle state. (EFM_(D)FSR.OPTEND=1, EFM_(D)FSR.RDY=1)
- Step6.** Check the programming self-read result flag bit (EFM_(D)FSR.MISMATCH) :
If 0, programming is successful. If 1, the Flash address has been corrupted and should be permanently discarded.
- Step7.** Clear the programming end flag bit (EFM_(D)FSR.OPTEND).

Consistent with single programming without read-back function, performing a single programming read-and-write operation on a locked OTP address or a write-protected sector will fail, and the error flag bit 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 8μs. The steps for setting up continuous programming operations are as follows:

- Step1.** Disable the Flash register write protection. (EFM_(D)FAPRT writes 0x0123 first, then writes 0x3210)
- Step2.** Unlock the EFM_(D)KEY1. (EFM_(D)KEY1 writes 0x01234567 first, then writes 0xFEDCBA98)
- Step3.** Set continuous programming mode. (EFM_(D)FWMC.PEMOD=0b011)
- Step4.** Write 64/32-bit data to the programming address.
The write method is the same as "[Single Programming Without Read-back Function](#)". The programming address and the executing program must reside in different Flash blocks, or the executing program must run in an area outside Flash.
- Step5.** Wait for the operation end flag bit (EFM_(D)FSR.OPTEND) to be set.
- Step6.** Clear the operation end flag bit until EFM_(D)FSR.OPTEND reads 0.
- Step7.** Repeat steps 4-6 until all data is written.
- Step8.** Change continuous programming mode to non-continuous programming mode.
- Step9.** Wait for Flash to be in an idle state. (EFM_(D)FSR.RDY=1)
- Step10.** 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.

Consistent with the single programming without read-back function, performing continuous programming write operations on locked OTP addresses and write-protected sectors will not succeed, and the error flag bit will be set.

8.3.4.7 Sector Erasing Function

After performing a sector erasing operation on Flash, the data in the sector (ProgramFLASH: 512Bytes, DataFLASH: 512Bytes, EDataFLASH: 32Bytes) is refreshed to all 1s.

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

- Step1.** Remove the Flash register write protection (EFM_(D)FAPRT first writes 0x0123, then writes 0x3210).
- Step2.** Unlock the EFM_(D)KEY1 lock. (EFM_(D)KEY1 writes 0x01234567 first, then writes 0xFEDCBA98)
- Step3.** Set sector erasing mode (EFM_(D)FWMC.PEMOD=0b100).
- Step4.** Write any 32-bit value to any address within the sector to be erased (address must be 4-aligned).
- Step5.** Wait for the Flash operation to end and for it to be in an idle state. (EFM_(D)FSR.OPTEND=1, EFM_(D)FSR.RDY=1)
- Step6.** Clear the erasing end flag bit. (EFM_(D)FSR.OPTEND)

Consistent with the single program without read-back function, performing a sector erasing write operation on a locked OTP address or a write-protected sector will result in a programming failure, and the error flag bit will be set.

8.3.4.8 Full Erasing Function

EFM provides full erasing operations for ProgramFLASH, DataFLASH, and EDataFLASH. The full erasing space for ProgramFLASH is 256KB, for DataFLASH is 32KB, and for EDataFLASH is 2KB. After a full erasing of the OTPA, boot swap enable area, data security configuration area, ICGB area, and OTPB area, the data is preserved; data at these addresses can be modified by programming and sector erasing.

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

- Step1.** Remove the Flash register write protection (EFM_(D)FAPRT first writes 0x0123, then writes 0x3210).
- Step2.** Unlock the EFM_(D)KEY1 lock. (EFM_(D)KEY1 first writes 0x01234567, then writes 0xFEDCBA98)
- Step3.** Set the full erasing mode (EFM_(D)FWMC.PEMOD=0b101).
- Step4.** Write any 32-bit value to any address within the Flash block to be erased (address must be 4-aligned).
- Step5.** Wait for the Flash operation to end and for it to be in an idle state. (EFM_(D)FSR.OPTEND=1, EFM_(D)FSR.RDY=1)
- Step6.** Clear the erasing end flag bit. (EFM_(D)FSR.OPTEND)

After OTP is enabled at OTPA data address 7, sending a full erasing write operation to a ProgramFLASH address will not succeed, ProgramFLASH data will be preserved, and the flag bit EFM_FSR.OTPWERR will be set.

A successful full erasing of ProgramFLASH will synchronously erasing the ICGA area as well.

8.3.4.9 Sector (Full) Erasing Cancel Function

During Flash sector erasing or full erasing operations, the erasing cancel function can be used to exit the erasing and full erasing operations, enabling a quick response to interrupts. After the (full) erasing cancel function is enabled, if an MCU interrupt occurs during Flash sector (full) erasing, the sector (full) erasing operation is exited, the Flash operation end flag and the sector (full) erasing cancel flag are set,

and the Flash returns to the idle state. After the interrupt event is handled, sector (full) erasing must be performed again to ensure the Flash data is refreshed to all 1s.

Erasing cancel function setup steps:

- Step1.** Configure MCU interrupt
 - For maskable interrupt configuration, refer to the "Vector Digraph - Interrupt Event List" section in the "Interrupt Controller (INTC)" chapter.
 - For non-maskable interrupt configuration, refer to the "Functional Description - Non-Maskable Interrupt" section in the "Interrupt Controller (INTC)" chapter.
- Step2.** Set the Flash sector (full) erasing mode according to the sector erasing or full erasing operation steps.
- Step3.** Enable Flash sector (full) erasing cancel (EFM_(D)FWMC.ERCALE=1).
- Step4.** Process according to Flash type:
 - For ProgramFLASH sector erasing/full erasing operation: Set bus hold during sector (full) erasing (EFM_FWMC.BUSHLDCTL=0).
 - For DataFLASH or EDataFLASH sector erasing/full erasing operation: The bus control bit during sector (full) erasing can be set arbitrarily (EFM_DFWMC.BUSHLDCTL).
- Step5.** Write any 32-bit value to any Flash address within the sector to be (full) erased.
- Step6.** Wait for the Flash operation to end and enter the idle state. (EFM_(D)FSR.OPTEND=1, EFM_(D)FSR.RDY=1)
- Step7.** Clear the operation end flag bit. (EFM_(D)FSR.OPTEND)
- Step8.** Check if the Flash is in the sector (full) erasing cancel state (EFM_(D)FSR.ERCAL=1):
If yes, clear the sector (full) erasing cancel flag bit and perform the sector erasing or full erasing operation again.

8.3.4.10 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 (SWD), and test interfaces.

Protection Level 0: No Protection

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

Protection Level 1

When the Flash addresses 0x01000B00-0x01000B03, 0x01000B04-0x01000B07, and 0x01000B08-0x01000B0B are all programmed with data of 0xAF180402, and simultaneously a 96-bit password is programmed into addresses 0x01001000-0x0100102F, protection level 1 is enabled.

- Flash addresses 0x01001000-0x01001003, 0x01001004-0x01001007, and 0x01001008-0x0100100B are programmed to contain the same data as the lower 32 bits of a 96-bit password.
- Flash addresses 0x01001010-0x01001013, 0x01001014-0x01001017, and 0x01001018-0x0100101B are programmed to contain the same data as the middle 32 bits of a 96-bit password.
- Flash addresses 0x01001020-0x01001023, 0x01001024-0x01001027, and 0x01001028-0x0100102B are programmed to contain the same data as the high 32 bits of a 96-bit password.

After enabling protection level 1:

- Debug interfaces and test interfaces cannot access MCU resources.
- User programs cannot program or perform sector erases in ICGA area.
- The data at addresses 0x01001000-0x0100102F 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 at OTPA data address 7, then a full erase of ProgramFLASH will be ineffective. Please consult the sales window for password authentication and full erase methods.

Protection Level 2

When the data written to Flash addresses 0x01000B10-0x01000B13, 0x01000B14-0x01000B17, and 0x01000B18-0x01000B1B is all 0xA85173AE, protection level 2 is enabled.

After enabling protection level 2:

- Debug interfaces and test interfaces cannot access MCU resources.
- User programs cannot program or perform sector erases in ICGA area.

After activating protection level 2, it can be reset to protection level 0 through full erase. If OTP is enabled at OTPA data address 7, then a full erase of ProgramFLASH will be ineffective.

Protection Level 3

Protection level 3 is enabled by programming data 0x42545048 into Flash addresses 0x01000B20-0x01000B23, 0x01000B24-0x01000B27, 0x01000B28-0x01000B2B.

After enabling protection level 3:

- Debug interfaces and test interfaces cannot rewrite Flash data.
- User programs cannot program or perform sector erases in ICGA area.

After activating protection level 3, it can be reset to protection level 0 through full erase. If the OTP enable bit at OTPA data address 7 is set, then a full chip erase of the ProgramFLASH is invalid.

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

8.3.4.11 BGO Function

The program executes on ProgramFLASH and performs programming, sector erasing, or full erasing on DataFLASH or EDataFLASH; or the program executes on RAM and performs programming, sector erasing, or full erasing on ProgramFLASH, DataFLASH, or EDataFLASH. You can set EFM_(D)FWMC.BUSHLDCTL to 1 (bus release during programming and erasing) to enable Flash programming or erasing while the program is executing.

The following are the steps for setting BGO functions:

- Step1.** Disable the Flash register write protection. (EFM_(D)FAPRT writes 0x0123 first, then writes 0x3210)
- Step2.** Unlock the EFM_(D)KEY1 lock. (EFM_(D)KEY1 writes 0x01234567 first, then writes 0xFEDCBA98)
- Step3.** Set the programming, sector erase or full erase mode. (EFM_(D)FWMC.PE-MOD=0b001/0b010/0b011/0b100/0b101)
- Step4.** Set the bus release during programming, sector erasing or full erasing (EFM_(D)FWMC.BUSHLDCTL=1)
- Step5.** Write data to the Flash address that needs to be programmed or erased.
- Step6.** Wait for the erase/write end flag bit to be set. (EFM_(D)FSR.OPTEND=1)

8.3.5 One-Time Programmable Byte (OTP)

This MCU provides two OTP areas: OTPA and OTPB. The address distribution is as follows:

- OTPA: Seven 64Bytes and one 16KBytes OTP data areas, plus eight 8Bytes latch areas.
- OTPB: Twenty-four 16Bytes OTP data areas, plus twenty-four 4Bytes latch areas.

Table 8-2 OTPA Address Distribution Table

Sector	OTPA Block Data Address	OTPA Block Latch Address
0	0x01000C00-0x01000C3F	0x01000DC0-0x01000DC7
1	0x01000C40-0x01000C7F	0x01000DC8-0x01000DCF
...	...	...
5	0x01000D40-0x01000D7F	0x01000DE8-0x01000DEF
6	0x01000D80-0x01000DBF	0x01000DF0-0x01000DF7
7	0x00000000-0x00003FFF	0x01000DF8-0x01000DFF

Table 8-3 OTPB Address Distribution Table

Sector	OTPB Block Data Address	OTPB Block Latch Address
0	0x40017A00-0x40017A0F	0x40017B80-0x40017B83
1	0x40017A10-0x40017A1F	0x40017B84-0x40017B87
...	...	...
22	0x40017B60-0x40017B6F	0x40017BD8-0x40017BDB
23	0x40017B70-0x40017B7F	0x40017BDC-0x40017BDF

When all latch address data of OTPA is all 1s, OTPA programming and sector erasing are permitted, OTPA latch address full erasing is prohibited, OTPA data addresses 0-6 full erasing is prohibited, and OTPA data address 7 full erasing is permitted.

When all latch address data of OTPB is all 1s, OTPB programming and sector erasing are permitted, and full erasing is prohibited.

When OTPA latch address data contains 0 (it is recommended to write all 0s), programming of the latch address is not possible, and OTPA latch address sector erasing and full erasing are prohibited. OTPA data addresses 0-6 sector erasing and full erasing are prohibited. When OTPA latch address 7 data contains 0 (it is recommended to write all 0s), OTPA data address 7 sector erasing and full erasing are prohibited.

When OTPB latch address data contains 0 (it is recommended to write all 0s), programming of the latch address is not possible, and sector erasing and full erasing of the entire OTPB area are prohibited.

The status flag bits of the OTPA area are consistent with those of ProgramFLASH, and the status flag bits of the OTPB area are consistent with those of DataFLASH.

OTPA data address 7 shares 16KBytes of physical space with ProgramFLASH addresses 0x00000000-0x00003FFF (sectors 0-31). If the latch address data of this space is all 1s, this area is ordinary Flash space and can be programmed and erased multiple times. If the corresponding latch address data contains 0, the corresponding sector becomes read-only, cannot be programmed or erased, and ProgramFLASH will be unable to perform a full erasing.

To program OTP latch addresses, unlock the EFM_(D)KEY2 register (EFM_(D)KEY2 is first written with 0x10325476, then with 0xEFCDA89). If an incorrect sequence is written to the EFM_(D)KEY2 register, a bus error will occur, and the EFM_(D)KEY2 register will lock until reset.

Please follow the steps below to operate the OTP area:

- Step1.** Disable the Flash register write protection. (EFM_(D)FAPRT: First write 0x0123, then write 0x3210).
- Step2.** Unlock the EFM_(D)KEY1 lock. (EFM_(D)KEY1: First write 0x01234567, then write 0xFEDCBA98).
- Step3.** Set programming mode. (EFM_(D)FWMC.PEMOD=0b001/0b010/0b011).
- Step4.** Write data to OTP addresses as needed.
- Step5.** Unlock the EFM_(D)KEY2 lock. (EFM_(D)KEY2: First write 0x10325476, then write 0xEFCDAB89).
- Step6.** Write 0 to any bit of OTP latch addresses. (It is recommended to write all 0)

After chip reset, the hardware will automatically load OTP latch address data into the circuit;

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

After writing to OTP latch addresses, to return to the EFM_(D)KEY2 locked state, write 1 to EFM_(D)FWMC.KEY2LOCK.

8.3.6 Boot Swap

Programming the boot swap enable address (0x01000E00) by writing the data 0x5A5A5A enables the boot swap function.

After programming the boot swap enable address and a reset, the addresses of ProgramFLASH sectors 0-7 and sectors 8-15 are swapped. The EFM_FSWP.FSWP bit can be queried to check if boot swap is enabled. After boot swap is enabled, the address transformation is shown in the following figure.

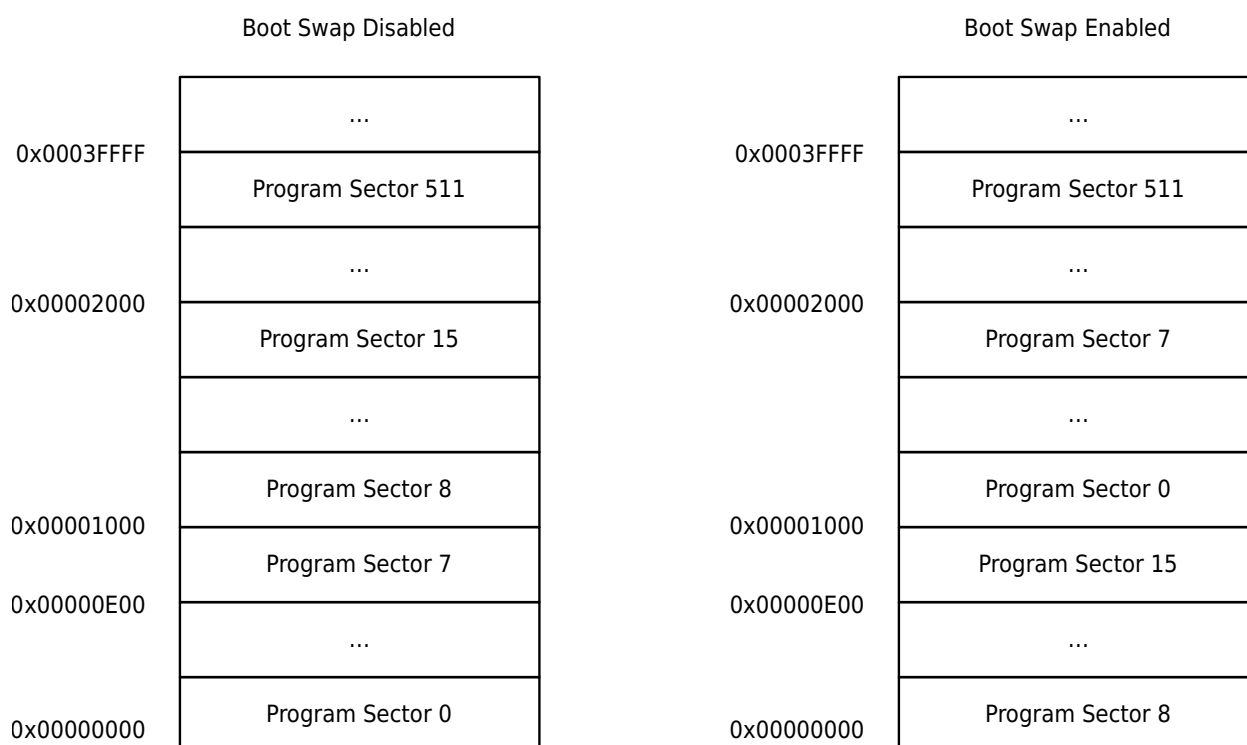


Figure 8-2 Address Transformation After Boot Swap

After boot swap is enabled, to revert to the initial state, a sector erase must be performed on address 0x01000E00.

The boot swap operation flow is shown in the following figure.

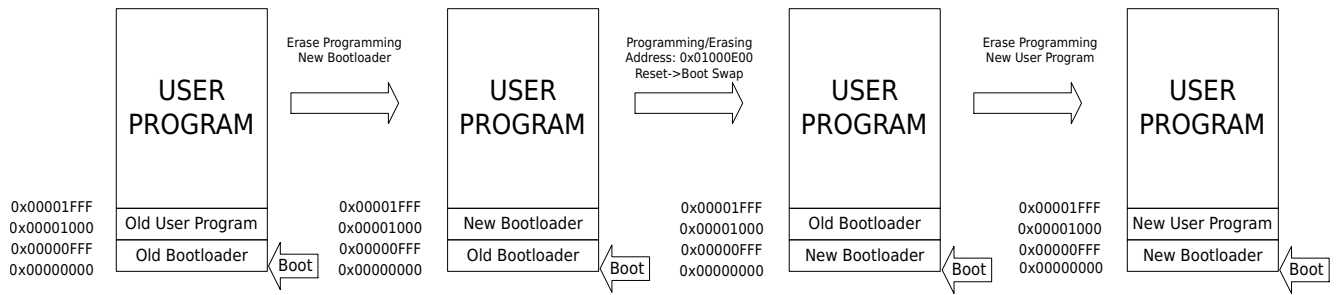


Figure 8-3 Boot Swap Operation Flow

If OTPA data address 7 is not latched (data at 0x01000DF8-0x01000DFF is all 1s), after boot swap becomes effective, the entire addresses of ProgramFLASH sectors 0-7 and sectors 8-15 are swapped; and the OTP latch address for OTPA data address 7 (0x01000DF8-0x01000DFF) cannot be programmed.

If OTPA data address 7 is already latched (data at 0x01000DF8-0x01000DFF is not all 1s), when attempting to set boot swap effective, boot swap will also be ineffective, and the addresses of ProgramFLASH sectors 0-7 and sectors 8-15 will not be swapped.

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 Data ECC Check

Flash data ECC can correct 1-bit errors and detect 2-bit errors.

- During programming, the Flash controller calculates the ECC code based on the bus data and writes it to the Flash along with the data.
- During erasure, the Flash data and the ECC code are refreshed to all 1s.

When performing a read operation on the Flash, the Flash controller determines whether ECC check is required based on whether the ECC check function is enabled:

- If the ECC check function is not enabled, no check occurs, and the ECC error flag will not be set.
- If the ECC check function is enabled, the check is performed. If a data error is detected, a Non-Maskable Interrupt (NMI) or a reset is generated according to the register bit EFM_(D)FEC setting. The interrupt flag bit can be queried via EFM_(D)FSR, and the reset flag bit can be queried via RMU_RSTF0.

8.3.7.2 Error Injection Read

This chip provides an error injection function, which can be used for ECC check circuit self-test. After enabling this function and performing a Flash read, the hardware circuit inverts the read data bits according to the bits set to 1 in the error injection data register, and then performs ECC check.

When the ECC circuit is functioning normally, if one bit is set to 1 in the error injection data register, the ECC circuit will detect a 1-bit data error. If two bits are set to 1 in the error injection data register, the ECC circuit will detect a 2-bit data error. If more than two bits are set to 1 in the error injection data register, the ECC circuit may fail to detect the error.

The steps to set the error injection read mode are as follows:

1. Disable the Flash register write protection. (Write 0x0123, then 0x3210 to EFM_(D)FAPRT)
2. Set the ECC check enable. (Example: EFM_(D)FEC.ECCMOD=0b11, EFM_(D)FEC.ECCOAD=0b0)
3. Set the ECC error injection read mode. (EFM_(D)FRMC.EIRD=0b1)
4. Set data to the ECC error injection data register. According to the ECC circuit principle, it is recommended to set at most two bits to 1.
 - ProgramFLASH:
 - ▶ Set 32-bit data to EFM_ERI0.
 - ▶ Set 32-bit data to EFM_ERI1.
 - ▶ Set 8-bit data to EFM_ERI2.
 - DataFLASH and EDataFLASH:
 - ▶ Set 32-bit data to EFM_DERI0.
 - ▶ Set 7-bit data to EFM_DERI2.
5. Read the already programmed Flash address.
6. Read the flag bit EFM_(D)FSR to determine if ECC1ERR and ECC2ERR are set.
Read the ECC check error address register EFM_(D)ECERA to determine if the address is the Flash read address.
7. Set back to Normal read mode. (EFM_(D)FRMC.EIRD=0b0)

The above program should be executed on a different Flash from the check address (program in ProgramFLASH, check address in DataFLASH or EDataFLASH), or executed in other memory (such as RAM).

8.3.7.3 MARGIN Read

This product provides the MARGIN read mode. MARGIN read data will produce errors before the Flash data retention capability reaches its end of life, thereby achieving the purpose of early prediction of Flash data degradation.

There are two types of MARGIN read: MARGIN0 read and MARGIN1 read:

- MARGIN0 read is a stricter method than normal read for reading data 0;
- MARGIN1 read is a stricter method than normal read for reading data 1.

Therefore, if Flash data 0 is read as 0 in normal read mode but read as 1 in MARGIN0 mode, it indicates that the probability of this Flash data being misread as 1 in normal read mode increases, and it needs to be erased and reprogrammed to ensure sufficient read margin. Similarly, if Flash data 1 is read as 1 in normal read mode but read as 0 in MARGIN1 mode, it indicates that the probability of this Flash data being misread as 0 in normal read mode increases, and it needs to be erased to ensure sufficient read margin.

MARGIN read mode setting example:

1. Disable the Flash register write protection. (Write 0x0123 then 0x3210 to EFM_(D)FAPRT)
2. Set MARGIN0 read mode. (Set EFM_(D)FRMC.M0RD to 1)
Set MARGIN1 read mode. (Set EFM_(D)FRMC.M1RD to 1)
3. Read the EFM_(D)FRMC register and verify that the register bits EFM_(D)FRMC.M0/1RD=1 are correctly set.
4. Set the Flash read wait cycle according to the bus clock frequency.
5. Read the EFM_(D)FRMC register and verify that the read wait cycle EFM_(D)FRMC.FLWT is correctly set.
6. Perform a read operation on the Flash address and compare it with the expected value.
7. Exit MARGIN0 read mode. (Set EFM_(D)FRMC.M0RD to 0)
Exit MARGIN1 read mode. (Set EFM_(D)FRMC.M1RD to 0)

The setting of MARGIN0 and MARGIN1 read modes and the setting of the read wait cycle must be done in two separate setting operations. They cannot be modified simultaneously in a single write operation. The above program must be executed in memory other than the object Flash (such as RAM).

8.3.8 Interrupts and Reset

The EFM module has a total of 8 NVIC interrupts, with 4 each for ProgramFLASH and DataFLASH. These are: PE (Program/Erase) error interrupt, operation end interrupt, read/write conflict interrupt, and blank check error interrupt. There is also 1 ECC error interrupt (non-maskable interrupt). The interrupt flag bits are queried by reading the EFM_(D)FSR register.

The EFM module has a total of 4 reset sources, with 2 each for ProgramFLASH and DataFLASH. These are: ECC 1-bit error reset source and ECC 2-bit error reset source.

The interrupts and reset sources for EDataFLASH are the same as those for DataFLASH.

Table 8-4 EFM Module Interrupt Source and Reset Source Comparison Table

Flash	Interrupt Source					Reset Source	
	PE (Program/Erase) Error Interrupt	Operation End Interrupt	Read/Write Conflict Interrupt	Blank Check Error Interrupt	ECC Error Interrupt (Non-Maskable Interrupt)	ECC 1-bit Error Reset Source	ECC 2-bit Error Reset Source
Program-FLASH	EFM_PPEERR	EFM_POP-TEND	EFM_PCOLERR	EFM_PBLKERR	EFM_EC-CERR	EFM_PECC1RST	EFM_PECC2RST
DataFLASH	EFM_DPEERR	EFM_DOP-TEND	EFM_DCOLERR	EFM_DBLKERR	EFM_EC-CERR	EFM_DECC1RST	EFM_DECC2RST
EData-FLASH	EFM_DPEERR	EFM_DOP-TEND	EFM_DCOLERR	EFM_DBLKERR	EFM_EC-CERR	EFM_DECC1RST	EFM_DECC2RST

The conditions for setting and clearing interrupt sources are as follows:

1. PE Error Interrupt EFM_P(D)PEERR:

● **Set Condition:**

- ▶ Issuing a programming, sector erasing, or full erasing operation to a latched OTP address (OTPWERR=1).
- ▶ Issuing a programming or sector erasing operation to a write-protected sector (PRTWERR=1).
- ▶ Program data size is not 32 bits (PGSZERR=1).

During ProgramFLASH programming, the first write address does not end with 0 or 8 (PGSZERR=1).

The second write address does not end with 4 or C (PGSZERR=1).

The two write addresses are not consecutive (PGSZERR=1). (When the Cortex-M0+ core attempts to execute a non-4-byte-aligned access, a hardware exception will be triggered.)

- ▶ In single program read-back mode, the hardware automatically read value from the program address does not match the written value (MISMTCH=1).

● **Clear Condition:**

Write 1 to the corresponding flag clear bit in the EFM_(D)FSCLR register to clear the status bit.

2. Operation End Interrupt EFM_P(D)OPTEND:

● **Set Condition:**

- ▶ Programming mode: Single address programming ends (OPTEND=1).
- ▶ Erasing mode: Sector erasing or full erasing ends (OPTEND=1).

● **Clear Condition:**

Write 1 to the corresponding clear bit in the EFM_(D)FSCLR register to clear the status bit.

3. Read/Write Collision Interrupt EFM_P(D)COLERR:

● **Set Condition:**

Read or write operation to the Flash block while the Flash is busy (RDY=0). (Excluding continuous writes to Flash in continuous programming mode.)

● **Clear Condition:**

Write 1 to the corresponding clear bit in the EFM_(D)FSCLR register to clear the status bit.

4. Blank Check Error Interrupt EFM_P(D)BLKERR:

- **Set Condition:**

When Blank read mode is enabled, the read Flash data and ECC code are not all 1s.

- **Clear Condition:**

Write 1 to the corresponding flag clear bit in the EFM_(D)FSCLR register to clear the status bit.

5. ECC Error Interrupt EFM_ECCERR (Non-Maskable Interrupt):

- **Set Condition:**

- ▶ EFM_FEC.ECCMOD[1]=1, EFM_FEC.ECCOAD=0 setting, and a 1-bit error is detected in the data when reading ProgramFLASH. (ECC1ERR)
- ▶ EFM_DFEC.ECCMOD[1]=1, EFM_DFEC.ECCOAD=0 setting, and a 1-bit error is detected in the data when reading DataFLASH or EDataFLASH. (ECC1ERR)
- ▶ EFM_FEC.ECCMOD[1:0] is not 0b00, EFM_FEC.ECCOAD=0 setting, and a 2-bit error is detected in the data when reading ProgramFLASH. (ECC2ERR)
- ▶ EFM_DFEC.ECCMOD[1:0] is not 0b00, EFM_DFEC.ECCOAD=0 setting, and a 2-bit error is detected in the data when reading DataFLASH or EDataFLASH. (ECC2ERR)

- **Clear Condition:**

Write 1 to the corresponding flag clear bit in the EFM_(D)FSCLR register to clear the status bit.

The reset source set and clear conditions are as follows:

1. ECC 1-bit error reset source EFM_P(D)ECC1RST (flag bit RMU_RSTF0.FLECRF):

- **Set condition:**

- ▶ When EFM_FEC.ECCMOD[1:0]=0b11 and EFM_FEC.ECCOAD=1 is set, a 1-bit error occurs in the data when reading ProgramFLASH.
- ▶ When EFM_DFEC.ECCMOD[1:0]=0b11 and EFM_DFEC.ECCOAD=1 is set, a 1-bit error occurs in the data when reading DataFLASH or EDataFLASH.

- **Clear condition:**

Set the RMU_RSTF0.CLRF bit to 1.

2. ECC 2-bit error reset source EFM_P(D)ECC2RST (flag bit RMU_RSTF0.FLECRF):

- **Set condition:**

- ▶ When EFM_FEC.ECCMOD[1:0] is not 0b00 and EFM_FEC.ECCOAD=1 is set, a 2-bit error occurs in the data when reading ProgramFLASH.
- ▶ When EFM_DFEC.ECCMOD[1:0] is not 0b00 and EFM_DFEC.ECCOAD=1 is set, a 2-bit error occurs in the data when reading DataFLASH or EDataFLASH.

- **Clear condition:**

Set the RMU_RSTF0.CLRF bit to 1.

8.4 Register Description

8.4.1 Register List

(ProgramFLASH) EFM Base Address: 0x40000800

Table 8-5 ProgramFLASH Register Offset

Offset	Register	Description	Width
0x0000	EFM_FAPRT	ProgramFLASH Access Protect Register	32
0x0004	EFM_KEY1	ProgramFLASH Key 1 Register	32
0x0008	EFM_KEY2	ProgramFLASH Key 2 Register	32
0x0014	EFM_FSTP	ProgramFLASH Stop Register	32
0x0018	EFM_FRMC	ProgramFLASH Read Mode Register	32
0x001C	EFM_FWMC	ProgramFLASH Write Mode Register	32
0x0020	EFM_FSR	ProgramFLASH State Register	32
0x0024	EFM_FSCLR	ProgramFLASH State Clear Register	32
0x0028	EFM_FITE	ProgramFLASH Interrupt Enable Register	32
0x002C	EFM_FSWP	ProgramFLASH Boot Swap Status Register	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
0x00B0	EFM_FEC	ECC Control Register	32
0x00B4	EFM_FECERA	ECC Error Address Register	32
0x00B8	EFM_ERI0	Error Injection Register 0	32
0x00BC	EFM_ERI1	Error Injection Register 1	32
0x00C0	EFM_ERI2	Error Injection Register 2	32
0x0120	EFM_FNWPRT0	ProgramFLASH Write Protection Register 0	32
0x0124	EFM_FNWPRT1	ProgramFLASH Write Protection Register 1	32

(DataFLASH and EDataFLASH) EFMD Base Address: 0x40000400

Table 8-6 DataFLASH and EDataFLASH Register Offset

Offset	Register	Description	Width
0x0000	EFM_DFAPRT	DataFLASH Access Protect Register	32
0x0004	EFM_DKEY1	DataFLASH Key 1 Register	32

Offset	Register	Description	Width
0x0008	EFM_DKEY2	DataFLASH Key 2 Register	32
0x0014	EFM_DFSTP	DataFLASH Stop Register	32
0x0018	EFM_DFRMC	DataFLASH Read Mode Register	32
0x001C	EFM_DFWMC	DataFLASH Write Mode Register	32
0x0020	EFM_DFSR	DataFLASH State Register	32
0x0024	EFM_DFSCLR	DataFLASH State Clear Register	32
0x0028	EFM_DFITE	DataFLASH Interrupt Enable Register	32
0x00B0	EFM_DFEC	ECC Control Register	32
0x00B4	EFM_DFECERA	ECC Error Address Register	32
0x00B8	EFM_DERI0	Error Injection Register 0	32
0x00C0	EFM_DERI2	Error Injection Register 2	32
0x0120	EFM_DFNWPRT0	DataFLASH Write Protection Register 0	32
0x0124	EFM_DFNWPRT1	EDataFLASH Write Protection Register 1	32

8.4.2 Flash Access Protect Register (EFM_(D)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. Release method: FAPRT writes 16-bit data "0x0123" first, then writes 16-bit data "0x3210". In the unprotected state, write any data, and the EFM register re-enters the protected state. When EFM register access protection is enabled, the read value of this register is 0x00000000. When EFM register access protection is disabled, the register reads as 0x00000001.	RW

8.4.3 Flash Key 1 Register (EFM_(D)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_(D)FWMC write protection register. EFM_(D)KEY1 writes 0x01234567 first, then writes 0xFEDCBA98 to release the write protection of EFM_(D)FWMC. The register reads as 0x00000000.	RW

8.4.4 Flash Key 2 Register (EFM_(D)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. EFM_(D)KEY2 writes 0x10325476 first, then writes 0xEFCDAB89 to release the write protection of the OTP latch address. The register reads as 0x00000000.	RW

8.4.5 Flash Stop Register (EFM_(D)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																															FSTP
Access																																RW

Bit	Name	Function	Description	Access
31:1	Reserved	Reserved	-	-
0	FSTP	ProgramFLASH/ DataFLASH Stop Mode Control	0b0: Flash in active state 0b1: Flash in stop mode  Note After EFM_DFSTP.FSTP is set to 1, the EDataFLASH enters stop mode. After this bit is changed from 1 to 0, wait until the EFM_(D)FSR.RDY bit is confirmed as 1 before accessing ProgramFLASH/DataFLASH/EDataFLASH.	RW

8.4.6 Flash Read Mode Register (EFM_(D)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	0x00000003																															
Name	Reserved													PREFE	Reserved							EIRD	BLKDE	MIRD	MORD	Reserved					FLWT	
Access														RW								RW	RW	RW	RW						RW	

Bit	Name	Function	Description	Access
31:19	Reserved	Reserved	-	-
18	PREFE	Prefetch Per- mission Bit	0b0: Disable prefetch function 0b1: Enable prefetch function  Note This bit does not exist in EFM_DFRMC.	RW
17:12	Reserved	Reserved	-	-
11	EIRD	ECC Error Injec- tion Read Mode	0b0: Normal read mode 0b1: Error injection read mode	RW
10	BLKDE	Blank Read Mode	0b0: Normal read mode 0b1: Blank read mode	RW
9	M1RD	MARGIN1 Read Mode	0b0: Normal read mode 0b1: MARGIN1 read mode	RW
8	M0RD	MARGIN0 Read Mode	0b0: Normal read mode 0b1: MARGIN0 read mode	RW
7:4	Reserved	Reserved	-	-
3:0	FLWT	Flash Read In- serted Wait Pe- riod	0b0000: No read wait period is inserted 0b0001: Insert 1 read wait period 0b0010: Insert 2 read wait periods 0b1110: Insert 14 read wait periods 0b1111: Insert 15 read wait periods	RW

8.4.7 Flash Write Mode Register (EFM_(D)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			ERCALE	Reserved	PEMOD																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																			
Access																	RW	RW																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																		

Bit	Name	Function	Description	Access
31:18	Reserved	Reserved	-	-
17	KEY2LOCK	Flash Key 2 Protection Status Bit	Write 1 to lock the EFM_(D)KEY2 register, write 0 has no effect Read: 0b0: EFM_(D)KEY2 sequence unlocked 0b1: EFM_(D)KEY2 sequence locked  Note After writing an incorrect sequence to EFM_(D)KEY2, this bit remains 1 until reset.	RW
16	KEY1LOCK	Flash Key 1 Protection Status Bit	Write 1 to lock the EFM_(D)KEY1 register, write 0 has no effect Read: 0b0: EFM_(D)KEY1 sequence unlocked 0b1: EFM_(D)KEY1 sequence locked  Note After writing an incorrect sequence to EFM_(D)KEY1, this bit remains 1 until reset.	RW
15:11	Reserved	Reserved	-	-
10	BLKECCRSEL	Program-FLASH Read 64-bit Data and 8-bit ECC All Ones ECC Selection Bit	0b0: No ECC error 0b1: ECC error determined by algorithm	RW
		ECC Selection Bit for Data-FLASH/EData-FLASH Reading 32-bit Data and 7-bit ECC All 1	0b0: No ECC error 0b1: ECC error determined by algorithm	RW

Bit	Name	Function	Description	Access
9	BLKECCWSEL	Program-FLASH Programming 64-bit All-1 Data ECC Selection Bit	● 0b0: Data automatically generated during programming ● 0b1: 0xFF	RW
		DataFLASH/EDataFLASH Programming 32-bit All-1 Data ECC Selection Bit	● 0b0: Data automatically generated during programming ● 0b1: 0x7F	RW
8	BUSHLDCTL	Bus Control During Flash Programming/Erasing	● 0b0: During Flash programming and erasing, the bus is occupied ● 0b1: During Flash programming and erasing, the bus is released	RW
7:5	<i>Reserved</i>	<i>Reserved</i>	-	-
4	ERCALE	Flash Sector (Full) Erasing Cancel Permission Bit	● 0b0: Flash sector (full) erasing cancel function is not permitted ● 0b1: Flash sector (full) erasing cancel function is permitted	RW
3	<i>Reserved</i>	<i>Reserved</i>	-	-
2:0	PEMOD	Flash Programming/Erasing Mode	● 0b000: Read-only Mode ● 0b001: Single Programming without Read-back Mode ● 0b010: Single Programming with Read-back Mode ● 0b011: Continuous Programming Mode ● 0b100: Sector Erasing Mode ● 0b101: Flash Full Erasing Mode ● 0b110/0b111: Read-only Mode	RW

8.4.8 Flash State Register (EFM_(D)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	0x00000100																																	
Name	Reserved																				BLKERR	ECC2ERR	ECC1ERR	FSTWRST	RDY	Reserved	ERCAL	COLERR	OPTEND	MISMTCH	PGSZERR	PRTWERR	OTPWERR	
Access																					RO	RO	RO	RO	RO		RO	RO	RO	RO	RO	RO	RO	RO

Bit	Name	Function	Description	Access
31:13	Reserved	Reserved	-	-
12	BLKERR	Flash Data Blank Check Error Flag Bit	0b0: Flash data together with ECC code are all ones 0b1: Flash data together with ECC code are not all ones	RO
11	ECC2ERR	Flash Data Has a 2-Bit Error	0b0: Flash data has no 2-bit error 0b1: Flash data has a 2-bit error	RO
10	ECC1ERR	Flash Data Has a 1-Bit Error	0b0: Flash data has no 1-bit error 0b1: Flash data has a 1-bit error	RO
9	FSTWRST	ProgramFLASH First 32-Bit Data Write Flag Bit	0b0: Flag not generated 0b1: Flag generated Setting conditions: - First 32-bit data write to ProgramFLASH Clearing conditions: - ProgramFLASH second write operation flag bit - Write 1 to EFM_FSCLR.FSTWRSTCLR  Note This bit does not exist in EFM_DFSR.	RO
8	RDY	Flash Busy/Idle State	0b0: Flash busy 0b1: Flash idle	RO
7	Reserved	Reserved	-	-
6	ERCAL	Flash Sector (Full) Erasing Cancel Flag Bit	0b0: Flash sector (full) erasing cancel has not occurred 0b1: Flash sector (full) erasing cancel has occurred	RO

Bit	Name	Function	Description	Access
5	COLERR	Flash Read and Write Conflict Error Flag Bit	● 0b0: No flag generated ● 0b1: Generate flag Setting conditions: ● Read and write operations on Flash when RDY is not set ● In continuous programming mode, writing to Flash 0 without setting Clearing conditions: ● Write 1 to the corresponding clear bit of EFM_(D)FSCLR	RO
4	OPTEND	Flash Space Programming/Erasing/Full Erasing Operation End Flag Bit	● 0b0: No flag generated ● 0b1: Generate flag Setting conditions: ● End of programming/erasing/full erasing operation Clearing conditions: ● Write 1 to the corresponding clear bit of EFM_(D)FSCLR	RO
3	MISMTCH	Flash Space Single Programming Read-Back Value Inconsistency Flag Bit	● 0b0: No flag generated ● 0b1: Generate flag Setting conditions: ● In single programming read-back mode, the read data from Flash after programming is inconsistent with the written value Clearing conditions: ● Write 1 to the corresponding clear bit of EFM_(D)FSCLR	RO
2	PGSZERR	Flash Space Programming Address and Size Misalignment Flag Bit	● 0b0: No flag generated ● 0b1: Generate flag Setting conditions: ● A non-32-bit write operation occurs during programming ● The first write address for ProgramFLASH is not aligned to 8, and the second write address is not consecutive with the first Clearing conditions: ● Write 1 to the corresponding clear bit of EFM_(D)FSCLR	RO
1	PRTWERR	Flash Write Protection Error Flag Bit	● 0b0: No flag generated ● 0b1: Generate flag Setting conditions: ● Programming, erasing on the write-protected sector Clearing conditions: ● Write 1 to the corresponding clear bit of EFM_(D)FSCLR	RO

Bit	Name	Function	Description	Access
0	OTPWERR	OTP Erase Error Flag of Flash	● 0b0: No flag generated ● 0b1: Generate flag Setting conditions: ● Programming, erasing, and full erasing on the OTP latch area Clearing conditions: ● Write 1 to the corresponding clear bit of EFM_(D)FSCLR	RO

**Note**

When the Cortex-M0+ core attempts to execute a non-4-byte-aligned access, a hardware exception will be triggered.

8.4.9 Flash State Clear Register (EFM_(D)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																				BLKERRCLR	ECC2ERRCLR	ECC1ERRCLR	FSTWRSTCLR	Reserved	ERCALCLR	COLERRCLR	OPTENDCLR	MISMTCHCLR	PGSZERRCLR	PRTWERRCLR	OTPWERRCLR	
Access																					RW	RW	RW	RW		RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	Function	Description	Access
31:13	Reserved	Reserved	-	-
12	BLKERRCLR	Clear the Flash Data Blank Check Error Flag	Write 1 to clear (D)FSR.BLKERR bit, write 0 has no effect; read always returns 0	RW
11	ECC2ERRCLR	Clear the Flash Data 2-Bit Error Flag	Write 1 to clear (D)FSR.ECC2ERR bit, write 0 has no effect; read always returns 0	RW
10	ECC1ERRCLR	Clear the Flash Data 1-Bit Error Flag	Write 1 to clear (D)FSR.ECC1ERR bit, write 0 has no effect; read always returns 0	RW
9	FSTWRSTCLR	Clear the First 32-bit Data Write Flag for ProgramFLASH	Write 1 to clear FSR.FSTWRST bit, write 0 has no effect; read always returns 0  Note This bit is invalid in EFM_DFSCCLR.	RW
8:7	Reserved	Reserved	-	-
6	ERCALCLR	Flash Sector (Full) Erase Cancel Flag Clear	Writing 1 clears the (D)FSR.ERCAL bit, writing 0 has no effect; read value is always 0	RW
5	COLERRCLR	Clear the Read and Write Conflict Error Flag of Flash	Write 1 to clear the (D)FSR.COLERR bit, write 0 has no effect; read always returns 0	RW
4	OPTENDCLR	Clear the Space Programming/Erasing/Full Erasing End Flag of Flash	Write 1 to clear the (D)FSR.OPTEND bit, write 0 has no effect; read always returns 0	RW

Bit	Name	Function	Description	Access
3	MISMTCHCLR	Clear the Space Single Programming Read-back Value Inconsistency Flag of Flash	Write 1 to clear the (D)FSR.MISMTCH bit, write 0 has no effect; read always returns 0	RW
2	PGSZERRCLR	Clear the Space Programming Address and Size Misalignment Flag of Flash	Write 1 to clear the (D)FSR.PGSZERR bit, write 0 has no effect; read always returns 0	RW
1	PRTWERRCLR	Clear the Write Protection Error Flag of Flash	Write 1 to clear the (D)FSR.PRTWERR bit, write 0 has no effect; read always returns 0	RW
0	OTPWERRCLR	Flash OTP Erasing Error Flag Cleared	Write 1 to clear the (D)FSR.OTPWERR bit; write 0 has no effect; read always returns 0	RW

8.4.10 Flash Interrupt Enable Register (EFM_(D)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																																BLKERRITE	COLERRITE	OPTENDITE	PEERRITE
Access																																	RW	RW	RW	RW

Bit	Name	Function	Description	Access
31:4	Reserved	Reserved	-	-
3	BLKERRITE	Blank Check Error Interrupt Enable	0b0: Interrupt Disable 0b1: Interrupt Enable	RW
2	COLERRITE	Read and Write Conflict Error Interrupt Enabled	0b0: Interrupt disabled 0b1: Interrupt enabled	RW
1	OPTENDITE	Operation End Interrupt Enabled	0b0: Interrupt disabled 0b1: Interrupt enabled	RW
0	PEERRITE	Programming/Erasing Error Interrupt Enabled	0b0: Interrupt disabled 0b1: Interrupt enabled	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	0xFFFFFFFF																															
Name	Reserved																															FSWP
Access																																RO

Bit	Name	Function	Description	Access
31:1	Reserved	Reserved	-	-
0	FSWP	Boot Swap Flag	0b0: Flash sector 0-7 and sector 8-15 addresses are not swapped 0b1: Flash sector 0-7 and sector 8-15 addresses are swapped  Note The initial value of this register is determined by the value at Flash address 0x01000E00-0x01000E02. If the data is 0x5A5A5A, the initial value after reset is 1; otherwise, it is 0.	RO

8.4.12 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	0x58480118																															
Name	CHIPID																															
Access	RO																															

Bit	Name	Function	Description	Access
31:0	CHIPID	Chip Specific Flag	CHIPID[31:16]: ASCII code 0x5848 of XH CHIPID[15:0]: Product model, for example, 0x0118	RO

8.4.13 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

8.4.14 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									RO								RO															

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

8.4.15 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	PTNMFG										Y_LOCATION						X_LOCATION															
Access	RO										RO						RO															

Bit	Name	Function	Description	Access
31:16	PTNMFG	Product Name Mark	Product PART NAME identification mark For example, 0xFFFF indicates K118, 0x0116 indicates K116	RO
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

8.4.16 ECC Control Register (EFM_(D)FEC)

Offset	Bit Position																																
0x00B0	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Reset	Determined by ICGA reset loading(ProgramFLASH);0x00000006(DataFLASH)																																
Name	Reserved																														ECCMOD		ECCOAD
Access																															RW		RW

Bit	Name	Function	Description	Access
31:3	Reserved	Reserved	-	-
2:1	ECCMOD	Flash ECC Check Permission Bit	0b00: ECC check function disabled 0b01: - If 1-bit error occurs, ECC will correct the error without generating a 1-bit error flag or causing an interrupt/reset. - If 2-bit error occurs, ECC will detect the error, generate a 2-bit error flag, and trigger an interrupt/reset. 0b10: - If 1-bit error occurs, ECC will correct the error and generate a 1-bit error flag, but without causing an interrupt/reset. - If 2-bit error occurs, ECC will detect the error, generate a 2-bit error flag, and trigger an interrupt/reset. 0b11: - If 1-bit error occurs, ECC will correct the error, generate a 1-bit error flag, and trigger an interrupt/reset. - If 2-bit error occurs, ECC will detect the error, generate a 2-bit error flag, and trigger an interrupt/reset.	RW
0	ECCOAD	Operate after a Flash ECC Check Error Occurred	0b0: Non-maskable interrupt 0b1: Reset	RW

8.4.17 ECC Error Address Register (EFM_(D)FECERA)

Offset	Bit Position																															
0x00B4	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	ECERA																															
Access	RO																															

Bit	Name	Function	Description	Access
31:0	ECERA	Flash ECC Error Address	Specifies the ECC error address. This register value is valid only when the flag bit EFM_(D)FSR.ECC1ERR or ECC2ERR = 1.  Note EFM_DECERA.ECERA[31:16] is fixed at 0x0000.	RO

8.4.18 Error Injection Register 0 (EFM_(D)ERI0)

Offset	Bit Position																															
0x00B8	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	EIR0[31:0]																															
Access	RW																															

Bit	Name	Function	Description	Access
31:0	ERI0[i] (i=31-0)	Error Injection for Flash Read Data Bit[i]	0b0: No error injection for Flash read data bit[i] 0b1: Error injection for Flash read data bit[i]	RW

8.4.19 Error Injection Register 1 (EFM_ERI1)

Offset	Bit Position																															
0x00BC	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	EIR1[31:0]																															
Access	RW																															

Bit	Name	Function	Description	Access
31:0	EIR1[i] (i=31-0)	Error Injection for Flash Read Data Bit [32+i]	0b0: No error injection for Flash read data bit [32+i] 0b1: Error injection for Flash read data bit [32+i]	RW

8.4.20 Error Injection Register 2 (EFM_(D)ERI2)

Offset	Bit Position																															
0x00C0	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved												ERI2[7:0]																			
Access													RW																			

Bit	Name	Function	Description	Access
31:8	Reserved	Reserved	-	-
7	EIR2[7]	Error Injection for Flash Read Data Bit 71	ProgramFLASH: 0b0: No error injection for Flash read data bit 71 0b1: Error injection for Flash read data bit 71 DataFLASH/EDataFLASH: This bit is meaningless	RW
6:0	EIR2[i] (i=6-0)	Error Injection for Program-FLASH Read Data Bit [64+i]	ProgramFLASH: 0b0: No error injection for Flash read data bit [64+i] 0b1: Error injection for Flash read data bit [64+i]	RW
		Error Injection for DataFLASH/EDataFLASH Read Data Bit [32+i]	DataFLASH/EDataFLASH: 0b0: No error injection for Flash read data bit [32+i] 0b1: Error injection for Flash read data bit [32+i]	RW

8.4.21 Flash Write Protection Register 0 (EFM_(D)FNWPRT0)

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	Consistent with ICGA3 (EFM_FNWPRT0); Consistent with ICGB0 (EFM_DFNWPRT0)																																
Name	FNWPRT31	FNWPRT30	FNWPRT29	FNWPRT28	FNWPRT27	FNWPRT26	FNWPRT25	FNWPRT24	FNWPRT23	FNWPRT22	FNWPRT21	FNWPRT20	FNWPRT19	FNWPRT18	FNWPRT17	FNWPRT16	FNWPRT15	FNWPRT14	FNWPRT13	FNWPRT12	FNWPRT11	FNWPRT10	FNWPRT9	FNWPRT8	FNWPRT7	FNWPRT6	FNWPRT5	FNWPRT4	FNWPRT3	FNWPRT2	FNWPRT1	FNWPRT0	
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	FNWPRT31	Flash Sector Write Protection Bit 31	ProgramFLASH: 0b0: Sector 248-255 programming and sector erasing disabled 0b1: Sector 248-255 programming and sector erasing enabled DataFLASH: 0b0: Sector 62-63 programming and sector erasing disabled 0b1: Sector 62-63 programming and sector erasing enabled  Note Writing 0 is valid after reset; writing 1 has no effect.	RW
30:2	...	...	...	RW
1	FNWPRT1	Flash Sector Write Protection Bit 1	ProgramFLASH: 0b0: Sector 8-15 programming and sector erasing disabled 0b1: Sector 8-15 programming and sector erasing enabled DataFLASH: 0b0: Sector 2-3 programming and sector erasing disabled 0b1: Sector 2-3 programming and sector erasing enabled  Note Writing 0 is valid after reset; writing 1 has no effect.	RW

Bit	Name	Function	Description	Access
0	FNWPRT0	Flash Sector Write Protection Bit 0	ProgramFLASH: ● 0b0: Sector 0-7 programming and sector erasing disabled ● 0b1: Sector 0-7 programming and sector erasing enabled DataFLASH: ● 0b0: Sector 0-1 programming and sector erasing disabled ● 0b1: Sector 0-1 programming and sector erasing enabled  Note Writing 0 is valid after reset; writing 1 has no effect.	RW

8.4.22 ProgramFLASH Write Protection Register 1 (EFM_FNWPRT1)

Offset	Bit Position																															
0x0124	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	Consistent with ICGA4																															
Name	FNWPRT63	FNWPRT62	FNWPRT61	FNWPRT60	FNWPRT59	FNWPRT58	FNWPRT57	FNWPRT56	FNWPRT55	FNWPRT54	FNWPRT53	FNWPRT52	FNWPRT51	FNWPRT50	FNWPRT49	FNWPRT48	FNWPRT47	FNWPRT46	FNWPRT45	FNWPRT44	FNWPRT43	FNWPRT42	FNWPRT41	FNWPRT40	FNWPRT39	FNWPRT38	FNWPRT37	FNWPRT36	FNWPRT35	FNWPRT34	FNWPRT33	FNWPRT32
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	FNWPRT63	Flash Sector Write Protection Bit 63	ProgramFLASH: 0b0: Sector 504-511 programming and sector erasing disabled 0b1: Sector 504-511 programming and sector erasing enabled  Note Writing 0 is valid after reset, writing 1 has no effect.	RW
30:2	...	...	...	RW
1	FNWPRT33	Flash Sector Write Protection Bit 33	ProgramFLASH: 0b0: Sector 264-271 programming and sector erasing disabled 0b1: Sector 264-271 programming and sector erasing enabled  Note Writing 0 is valid after reset, writing 1 has no effect.	RW
0	FNWPRT32	Flash Sector Write Protection Bit 32	ProgramFLASH: 0b0: Sector 256-263 programming and sector erasing disabled 0b1: Sector 256-263 programming and sector erasing enabled  Note Writing 0 is valid after reset; writing 1 has no effect.	RW

8.4.23 EDataFLASH Write Protection Register 1 (EFM_DFNWPRT1)

Offset	Bit Position																															
0x0124	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	Consistent with ICGB1																															
Name	ENWPRT31	ENWPRT30	ENWPRT29	ENWPRT28	ENWPRT27	ENWPRT26	ENWPRT25	ENWPRT24	ENWPRT23	ENWPRT22	ENWPRT21	ENWPRT20	ENWPRT19	ENWPRT18	ENWPRT17	ENWPRT16	ENWPRT15	ENWPRT14	ENWPRT13	ENWPRT12	ENWPRT11	ENWPRT10	ENWPRT9	ENWPRT8	ENWPRT7	ENWPRT6	ENWPRT5	ENWPRT4	ENWPRT3	ENWPRT2	ENWPRT1	ENWPRT0
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	ENWPRT31	EDataFLASH Sector Write Protection Bit 31	EDataFLASH: 0b0: Sector 62-63 programming and sector erasing disabled 0b1: Sector 62-63 programming and sector erasing enabled  Note Writing 0 is valid after reset; writing 1 has no effect.	RW
30:2	...	...	...	RW
1	ENWPRT1	EDataFLASH Sector Write Protection Bit 1	EDataFLASH: 0b0: Sector 2-3 programming and sector erasing are disabled. 0b1: Sector 2-3 programming and sector erasing are enabled.  Note Writing 0 is valid after reset; writing 1 has no effect.	RW
0	ENWPRT0	EDataFLASH Sector Write Protection Bit 0	EDataFLASH: 0b0: Sector 0-1 programming and sector erasing are disabled. 0b1: Sector 0-1 programming and sector erasing are enabled.  Note Writing 0 is valid after reset; writing 1 has no effect.	RW

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. Repeated programming on the same address cannot ensure the correctness of the data.
3. 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) in continuous programming mode.
In continuous programming mode, if EFM_(D)FSR.PRTWERR/OTPWERR/PGSZERR is set, exit continuous programming mode immediately.
4. When the bus is in release state (EFM_(D)FWMC.BUSHLCTL=1) during programming and erasing, set the vectors and subroutines to RAM or Flash blocks other than those being programmed or erased.
5. When ECC check is enabled, the 64-bit data and 8-bit ECC of unprogrammed addresses in Program-FLASH are all 1s, and the 32-bit data and 7-bit ECC of unprogrammed addresses in DataFLASH/EData-FLASH are all 1s. If an error occurs, ECC correction and detection are not guaranteed.

9 Built-in SRAM (SRAM)

9.1 Introduction

This product has 32KB system SRAM.

SRAM can be accessed as a byte, half-word (16 bits), or full-word (32 bits). Read and write operations can be performed at the CPU speed.

SRAM has ECC check. The ECC check is a one-correction-two-check code, which can correct one-bit error and check 2-bit errors. When an ECC error occurs during the reading of SRAM data, an SRAM ECC check reset will be triggered.

9.2 Functional Description

9.2.1 SRAM Space Allocation

Table 9-1 SRAM Space Allocation

Name	Capacity	Address Range	Check Mode
SRAM1	32KB	0x20000000-0x20007FFF	ECC check

9.2.2 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.

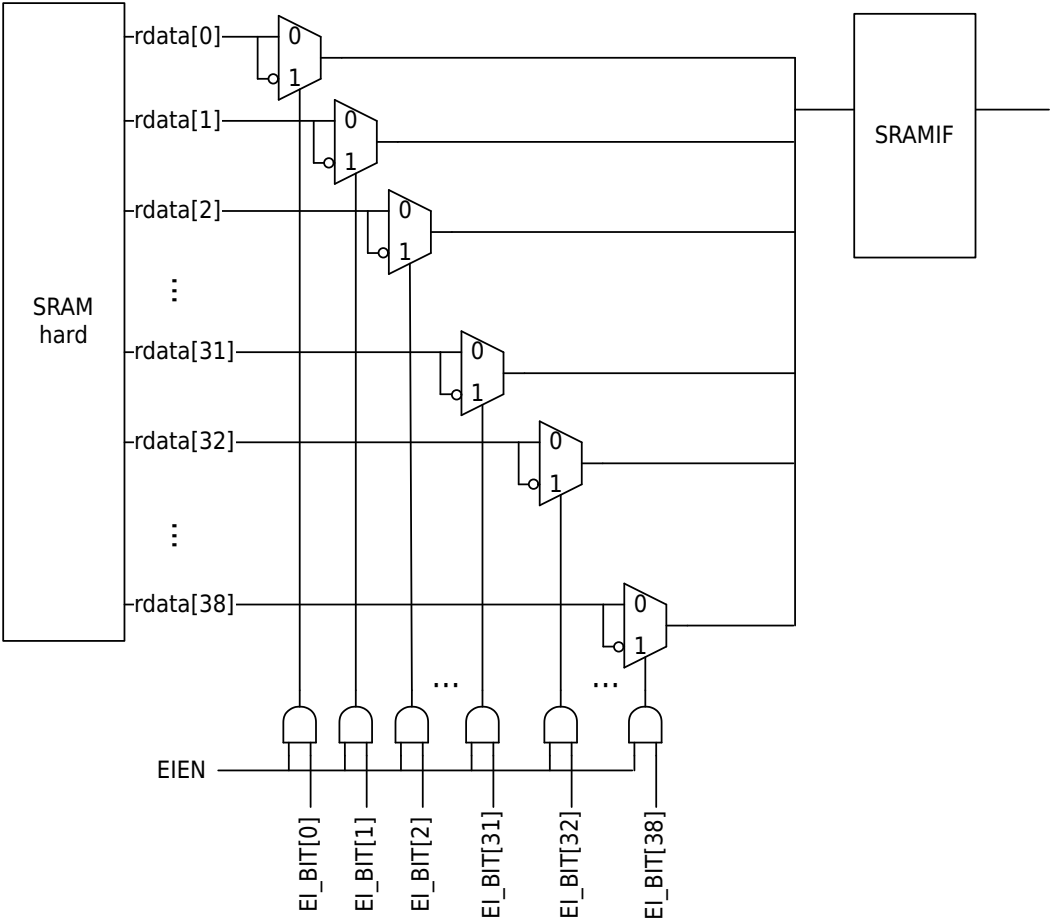


Figure 9-1 Error Injection Diagram for SRAM 39-bit

9.3 Register Description

9.3.1 Register List

Base address: 0x40010C00

Table 9-2 Built-in SRAM (SRAM) Register Offset

Offset	Register	Description	Width
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

9.3.2 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	Reserved																				SRAM1_ECCMOD	Reserved										SRAM1_ECCOAD	Reserved
Access																					RW											RW	

Bit	Name	Function	Description	Access
31:12	Reserved	Reserved	-	-
11:10	SRAM1_ECCMOD	SRAM1 ECC Check Enable Bit	0b00: Disable ECC check function. 0b01: - If a 1-bit error occurs, ECC will correct the error without generating a 1-bit error flag or causing an interrupt/reset. - If a 2-bit error occurs, ECC will detect the error, generate a 2-bit error flag, and trigger an interrupt/reset. 0b10: - If a 1-bit error occurs, ECC will correct the error and generate a 1-bit error flag, but without causing an interrupt/reset. - If a 2-bit error occurs, ECC will detect the error, generate a 2-bit error flag, and trigger an interrupt/reset. 0b11: - If a 1-bit error occurs, ECC will correct the error, generate a 1-bit error flag, and trigger an interrupt/reset. - If a 2-bit error occurs, ECC will detect the error, generate a 2-bit error flag, and trigger an interrupt/reset.	RW
9:2	Reserved	Reserved	-	-
1	SRAM1_ECCOAD	Operation After SRAM1 ECC Check Error	0b0: Non-maskable interrupt 0b1: Reset	RW
0	Reserved	Reserved	-	-

9.3.3 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	0b0: SRAM check control register write disabled 0b1: SRAM check control register write enabled  Note 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.	RW

**Note**

Writing 0x00000077 to this register sets CKPRC to 1; writing 0x00000076 clears CKPRC to 0.

9.3.4 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																												SRAM1_2ERR	SRAM1_1ERR	Reserved	
Access																													RW	RW		

Bit	Name	Function	Description	Access
31:4	Reserved	Reserved	-	-
3	SRAM1_2ERR	SRAM1 Occurred ECC 2-bit Error Flag	Write 1 to clear the flag, write 0 has no effect Read: 0b0: No 2-bit ECC errors 0b1: 2-bit ECC error occurs	RW
2	SRAM1_1ERR	SRAM1 Occurred ECC 1-bit Error Flag	Write 1 to clear the flag, write 0 has no effect Read: 0b0: No 1-bit ECC errors 0b1: 1-bit ECC error occurs	RW
1:0	Reserved	Reserved	-	-

9.3.5 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	0b0: Error injection disabled 0b1: Error injection enabled	RW

9.3.6 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: 0b0: No inversion, no ECC error injection for this bit. 0b1: Invert, inject ECC error for this bit.  Note Each bit corresponds to an output data bit.	RW

9.3.7 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: 0b0: No inversion, no ECC error injection for this bit. 0b1: Invert, inject ECC error for this bit.  Note Each bit corresponds to an output data bit.	RW

**Note**

Bits 38-32 of the SRAM1 output data are ECC bits.

9.3.8 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

9.4 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.
2. 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.
3. 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 E) represents a group of ports, such as PA represents a group of 14 I/O ports PA0-PA13.
- Pxy (x=A to E, y=0-17, the same below) represents a single I/O port, such as the PB7 port represents the 8th 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 18 I/O pins, which may be less than 18 depending on actual configuration
- Support pull-up and pull-down input
- Support push-pull, open-drain output mode
- Support normal and high 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 17 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)
- Some registers support the FASTIO interface, enabling single-cycle access

10.3 Functional Description

10.3.1 Function Block Diagram

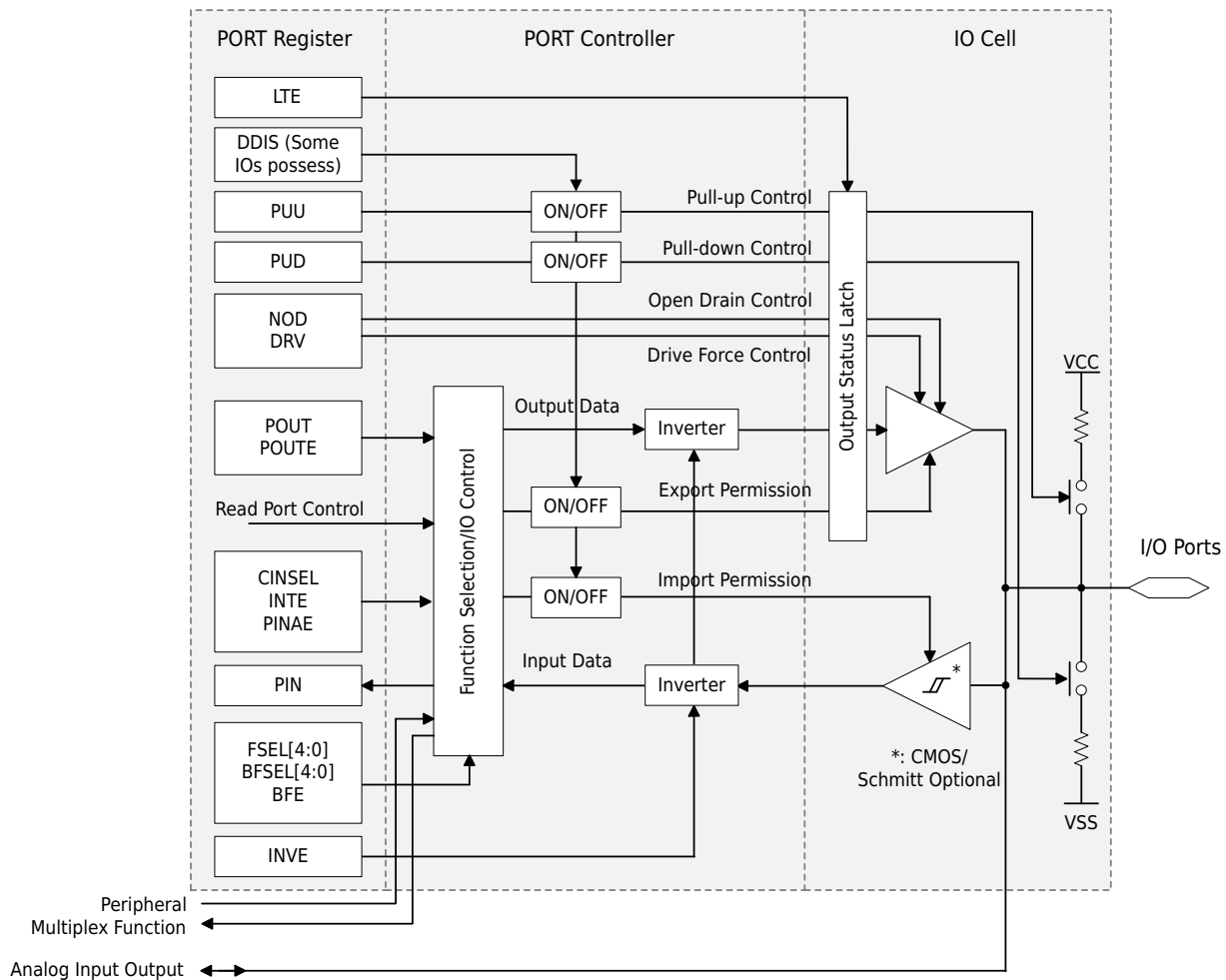


Figure 10-1 Port Basic Structure Diagram

For detailed GPIO port number, and drive capability configuration, please refer to the "Pin Definitions" chapter in the "Datasheet".

DDIS: Digital functions disabled. For details, please refer to the chapter "[Analog Function](#)".

10.3.2 General Input/Output GPIO Function

General Input Function GPI

Each I/O possesses a General Purpose Input (GPI) function. For I/Os with analog capabilities, the GPI function becomes inactive when the analog function is selected. When functions other than the analog function are chosen, the GPI function remains active at all times. For I/Os without analog capabilities, the GPI function is always active, regardless of the setting in the register 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.

General Output Function GPO

All 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 18 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.POUTEy. The output value of I/O can be controlled by setting PCRxy.POUT, where the PCRxy.POUT register bit is equivalent to PODRx.POUTy. PCRxy is suitable for controlling a single PORT and POERx/PODRx is suitable for controlling an 18-bit set of PORTs.

After the system is reset, except for the SDW multiplexing ports PA4/PC4, and the NRST terminal PA5, the initial functions of other ports are GPO (PFSRxy.FSEL=0x0) and are in high-impedance state (output disabled POUTE=0).

10.3.3 Peripheral Function

Through the register PFSRxy.FSEL, each port can configure up to 18 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".

SWD debug function, use the register PSPCR to select. PSPCR.SPFE[z], when z=0-1 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 0x03, and the SWD function is enable. If you want to set these ports as functions other than 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 8 ports. Event Port1 includes EVNTP10-EVNTP17, Event Port2 includes EVNTP20-EVNTP27, and so on. The EVNTPmn (m=1-4, n=0-7) 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 register (CMU_FCG) to active.

10.3.6 External Interrupt EIRQ Input Function

Each I/O port has an external interrupt input function. When the PCRxy.INTE bit is set to 1 or the PFSRxy.FSEL bit is set to 9-16, this I/O will be enabled as the external interrupt source EIRQy input. 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.



Notice

The PCRxy.INTE bit cannot be set to 1 simultaneously while PFSRxy.FSEL bit is set to 9-16.

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 PFSRxy.FSEL to disable the digital function of the current port.

10.3.8 Reset Terminal Multiplexing Configuration

The reset pin (PA5) on this product can be configured for other functions via PFSR50.FSEL (refer to the "Pin Function Table" section in the Datasheet). Prior to this configuration, the ICGA1x.NRSTEN bit must be written to 0 (where x = A/B/C) to disable the terminal reset function.

10.3.9 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/ Down 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.

Each I/O port has an internal pull down resistor. You can set the register PCRxy.PUD bit to enable this feature, which is in a weak pull down state when I/O ports do not have inputs. When the I/O port is in output state, the pull-down function will be automatically disabled.

**Note**

When the I/O port selects the I2Cx_SCL/I2Cx_SDA function, the setting of the register PUD will be ignored, and the internal pull-down function will be forced to be disable.

PCRxy.PUU and PCRxy.PUD can be 1 at the same time, but when the pull-up permission active, the pull-down permission will be blocked and the pull-down will be disable.

Drive Capability Control

Each I/O port has 2 levels of normal and high 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

The registers PIDRx, PODRx, POERx, POSRx, PORRx, POTRx, and PINAER in this section support 8-bit, 16-bit, and 32-bit access; registers PSPCR, PCCR, and PCRxy support 16-bit and 32-bit access; PWPR only supports 16-bit access; Event Port related registers only support 32-bit access.



Note

Unless otherwise specified, all register offset calculation formulas in this section are:

- x=A-E corresponds to n=0-4, y=0-17.

10.4.1 Register List

Base address: 0x40013800

Table 10-1 General-Purpose IO (GPIO) Register Offset

Offset	Register	Description	Width
0x00+0x4*n	PIDRx	General Input Data Register (x=A to E)	32
0x20+0x4*n	PODRx	General Output Data Register (x=A to E)	32
0x40+0x4*n	POERx	General Output Enable Register (x=A to E)	32
0x60+0x4*n	POSRx	General Output Set Register (x=A to E)	32
0x80+0x4*n	PORRx	General Output Reset Register (x=A to E)	32
0xA0+0x4*n	POTRx	General Output Toggle Register (x=A to E)	32
0x700	PSPCR	Special Control Register	16
0x704	PCCR	Common Control Register	16
0x708	PINAER	Input Control Register	16
0x70C	PWPR	Write Protection Register	16
0x400+0x80*n+0x4*y	PCRxy	General Control Register (x=A to E)	16
0x402+0x80*n+0x4*y	PFSRxy	Function Select Register (x=A to E)	16



Note

The access interface for registers PIDRx, PODRx, POERx, POSRx, PORRx, and POTRx is FASTIO; whereas the access interface for registers PSPCR, PCCR, PINAER, PWPR, PCRxy, and PFSRxy is AHB.

Event Port base address: 0x40000C00

Table 10-2 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

Offset	Register	Description	Width
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+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	0x000XXXXX																															
Name	Reserved																PIN[17:0]															
Access																	RO															

Bit	Name	Function	Description	Access
31:18	Reserved	Reserved	-	-
17:0	PINy (y=17-0)	Input Status	Corresponding to Px17-Px0 ports respectively: 0b0: I/O port input state is low level 0b1: I/O port input state is high level	RO



Note

When the port digital function is not disabled, 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, 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: 0x20+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	0x00000000																															
Name	Reserved																POUT[17:0]															
Access																	RW															

Bit	Name	Function	Description	Access
31:18	Reserved	Reserved	-	-
17:0	POUTy (y=17-0)	Output Data	Corresponding to Px17-Px0 ports respectively: 0b0: Output low level 0b1: Output high level	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: 0x40+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	0x00000000																															
Name	Reserved																POUTE[17:0]															
Access																	RW															

Bit	Name	Function	Description	Access
31:18	Reserved	Reserved	-	-
17:0	POUTE _y (y=17-0)	Output Enable	Corresponding to Px17-Px0 ports respectively: 0b0: Output disabled 0b1: Output enabled	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: 0x60+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	0x00000000																															
Name	Reserved																POS[17:0]															
Access																	WO															

Bit	Name	Function	Description	Access
31:18	Reserved	Reserved	-	-
17:0	POS _y (y=17-0)	Output Height	Corresponding to Px17-Px0 ports respectively: 0b0: The corresponding PODRx.POUT_y does not change 0b1: The corresponding PODRx.POUT_y is set to 1	WO

**Note**

The read value for this register is always 0. In 32-bit access, when PORy and POSy of the same I/O are written to 1 at the same time, PORy has a higher priority, that is, the corresponding POUTy is cleared.

10.4.6 General Output Reset Register (PORRx)

Offset: 0x80+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	0x00000000																															
Name	Reserved																POR[17:0]															
Access																	WO															

Bit	Name	Function	Description	Access
31:18	Reserved	Reserved	-	-
17:0	PORy (y=17-0)	Output Low	Corresponding to Px17-Px0 ports respectively: 0b0: The corresponding PODRx.POUTy does not change 0b1: The corresponding PODRx.POUTy is cleared	WO



Note

The read value for this register is always 0.

10.4.7 General Output Toggle Register (POTRx)

Offset: 0xA0+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	0x00000000																															
Name	Reserved																POT[17:0]															
Access																	WO															

Bit	Name	Function	Description	Access
31:18	Reserved	Reserved	-	-
17:0	POTy (y=17-0)	Output Inversion	Corresponding to Px17-Px0 ports respectively: 0b0: The corresponding PODRx.POUTy does not change 0b1: The corresponding PODRx.POUTy is inverted	WO

**Note**

The read value for this register is always 0.

10.4.8 Special Control Register (PSPCR)

Offset	Bit Position															
0x700	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0103															
Name	Reserved														SPFE[1:0]	
Access															RW	

Bit	Name	Function	Description	Access
15:9	Reserved	Reserved	-	-
8	Reserved	Reserved	Read as 1, Write as 1	-
7:2	Reserved	Reserved	-	-
1	SPFE[1]	Special Function Selection	0b0: SWDIO function disabled 0b1: SWDIO function enabled	RW
0	SPFE[0]	Special Function Selection	0b0: SWCLK function disabled 0b1: SWCLK function enabled	RW



Note

The PSPCR.SPFE have higher priority than the PFSRxy.FSEL.

10.4.9 Common Control Register (PCCR)

Offset	Bit Position															
0x704	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x4000															
Name	RDWT		Reserved										BFSEL			
Access	RW												RW			

Bit	Name	Function	Description	Access															
15:14	RDWT	Read Port Waiting	Set the number of wait periods inserted when reading register PCRxy	RW															
			<table><tr><th>Operating Frequency</th><th>Minimum Waiting Period</th><th>Setting Value</th></tr><tr><td>0<HCLK≤32MHz</td><td>No wait</td><td>0b00</td></tr><tr><td>32MHz<HCLK≤48MHz</td><td>1 period</td><td>0b01 (Default)</td></tr><tr><td>48MHz<HCLK≤64MHz</td><td>2 periods</td><td>0b10</td></tr><tr><td>48MHz<HCLK≤64MHz</td><td>3 periods</td><td>0b11</td></tr></table>		Operating Frequency	Minimum Waiting Period	Setting Value	0<HCLK≤32MHz	No wait	0b00	32MHz<HCLK≤48MHz	1 period	0b01 (Default)	48MHz<HCLK≤64MHz	2 periods	0b10	48MHz<HCLK≤64MHz	3 periods	0b11
			Operating Frequency		Minimum Waiting Period	Setting Value													
			0<HCLK≤32MHz		No wait	0b00													
			32MHz<HCLK≤48MHz		1 period	0b01 (Default)													
			48MHz<HCLK≤64MHz		2 periods	0b10													
48MHz<HCLK≤64MHz	3 periods	0b11																	
13:5	Reserved	Reserved	-	-															
4:0	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															
0x708	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
15:5	Reserved	Reserved	-	-
4:0	PINAER	Input Normally-Open	0b0: Input MOS normally-open disabled 0b1: Input MOS normally-open enabled The correspondence between each bit of PINAER and the I/O port is: - PINAER[0] Control PA0-PA17 - PINAER[1] Control PB0-PB17 - PINAER[2] Control PC0-PC17 - PINAER[3] Control PD0-PD17 - PINAER[4] Control PE0-PE17	RW

10.4.11 Write Protection Register (PWPR)

Offset	Bit Position																
0x70C	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	0b0: PSPCR, PCCR, PCRxy, PFSRxy register write disabled 0b1: PSPCR, PCCR, PCRxy, PFSRxy register write enabled	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: $0x400+0x80*n+0x4*y$

Offset	Bit Position															
-	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0X00															
Name	LK	Reserved			INTE	CINSEL	INVE	PIN	PUD	PUU	LTE	DRV	Reserved	NOD	POUTE	POUT
Access	RW				RW	RW	RW	RO	RW	RW	RW	RW		RW	RW	RW

Bit	Name	Function	Description	Access																						
15	LK	Lock Bit	0b0: The relevant bits of registers PCRxy and PFSRxy protected by LK are writable 0b1: The relevant bits of registers PCRxy and PFSRxy protected by LK are not writable The relevant bits of registers PCRxy and PFSRxy protected by LK: <table><tr><th>Name</th><th>Function</th></tr><tr><td>LK</td><td>Lock bit</td></tr><tr><td>CINSEL</td><td>Input mode selection</td></tr><tr><td>PUD</td><td>Pull-down enabled</td></tr><tr><td>PUU</td><td>Pull-up enabled</td></tr><tr><td>DRV</td><td>Drive mode selection</td></tr><tr><td>NOD</td><td>NMOS open-drain</td></tr><tr><td>FSEL</td><td>Function selection</td></tr><tr><td>BFE</td><td>Secondary function enabled</td></tr><tr><td>LTE</td><td>Latch output</td></tr><tr><td>INVE</td><td>Inverse enabled</td></tr></table>	Name	Function	LK	Lock bit	CINSEL	Input mode selection	PUD	Pull-down enabled	PUU	Pull-up enabled	DRV	Drive mode selection	NOD	NMOS open-drain	FSEL	Function selection	BFE	Secondary function enabled	LTE	Latch output	INVE	Inverse enabled	-
Name	Function																									
LK	Lock bit																									
CINSEL	Input mode selection																									
PUD	Pull-down enabled																									
PUU	Pull-up enabled																									
DRV	Drive mode selection																									
NOD	NMOS open-drain																									
FSEL	Function selection																									
BFE	Secondary function enabled																									
LTE	Latch output																									
INVE	Inverse enabled																									
14:12	Reserved	Reserved	-	-																						
11	INTE	External Interrupt Enabled	0b0: External interrupt input disabled 0b1: External interrupt input enabled  Note For I/O without external interrupt functionality, this bit is a reserved bit: read as 0, write as 0.	RW																						
10	CINSEL	Input Modes Selection	0b0: Schmitt input 0b1: CMOS input	RW																						

Bit	Name	Function	Description	Access
9	INVE	Inverse Enabled	0b0: Input and output data not inverted 0b1: Input and output data inverted	RW
8	PIN	Input Status	0b0: I/O port input state is low level 0b1: I/O port input state is high level The function is consistent with the register PIDRx.PINy	RO
7	PUD	Pull-down Enabled	0b0: Internal pull-down resistor disabled 0b1: Internal pull-down resistor enabled	RW
6	PUU	Pull-up Enabled	0b0: Internal pull-up resistor disabled 0b1: Internal pull-up resistor enabled	RW
5	LTE	Latch Output	0b0: Output latch invalid 0b1: Output latch valid	RW
4	DRV	Drive Mode Selection	0b0: Normal drive force mode 0b1: High drive force mode	RW
3	<i>Reserved</i>	<i>Reserved</i>	-	-
2	NOD	NMOS Open-drain	0b0: Normal CMOS output mode 0b1: NMOS open-drain output mode	RW
1	POUTE	Output Permission	0b0: Output disabled 0b1: Output enabled The function is consistent with the register PO-ERx.POUTEy	RW
0	POUT	Output Data	0b0: Output low level 0b1: Output high level The function is consistent with the register PODRx.POUTy	RW

**Note**

1. The lock bit (LK) is a security measure applied to prevent accidental writes to some key functions of the PCR and PFSR after they have been configured by software. The initial value of LK is 0, and the protected PCR and PFSR function bits are writable. When the LK bit is set to 1 by software, the protected PCR and PFSR function bits, including the LK bit itself, become non-writable. Writing 0 to LK when it is 1 has no effect. LK can be reset to 0 by a system reset.
2. 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.
3. 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.
4. 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: $0x402+0x80*n+0x4*y$

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 ● 0b0: Secondary function disabled ● 0b1: Secondary function enabled	RW
7:5	Reserved	Reserved	-	-
4:0	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.
- After reset, the initial state of the PA4 and PC4 ports is that the SWD function is enabled. When configuring the FSEL to select a function, you must first write 0 to the corresponding bit in the PSPCR register to disable the SWD function.

10.4.14 Event Port Direction Selection Register (PEVNTDIRRm)

Offset: $0x100 + 0x1C \times (m-1)$

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[7:0]							
Access																									RW							

Bit	Name	Function	Description	Access
31:8	Reserved	Reserved	-	-
7:0	PDIR[7:0]	Direction Selection	0b0: Event Port Input Function 0b1: Event Port Output Function	RW

10.4.15 Event Port Input Data Register (PEVNTIDRm)

Offset: $0x104 + 0x1C \times (m-1)$

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[7:0]							
Access																									RO							

Bit	Name	Function	Description	Access
31:8	Reserved	Reserved	-	-
7:0	PIN[7:0]	Port Input State	0b0: Event Port input state is low level when the event is triggered 0b1: Event Port input state is high level when the event is triggered	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: $0x108 + 0x1C \times (m-1)$

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[7:0]							
Access																									RW							

Bit	Name	Function	Description	Access
31:8	Reserved	Reserved	-	-
7:0	POUT[7:0]	Port Output Value	0b0: Event Port output low level 0b1: Event Port output high level	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[7:0]							
Access																									RW							

Bit	Name	Function	Description	Access
31:8	Reserved	Reserved	-	-
7:0	POR[7:0]	Output Value Reset	0b0: The corresponding PEVNTODRm.POUT does not change when the event is triggered 0b1: The corresponding PEVNTODRm.POUT is reset when the event is triggered	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[7:0]							
Access																									RW							

Bit	Name	Function	Description	Access
31:8	Reserved	Reserved	-	-
7:0	POS[7:0]	Output Value Set	0b0: The corresponding PEVNTODRm.POUT does not change when the event is triggered 0b1: When the event is triggered, the corresponding PEVNTODRm.POUT is set	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: $0x114 + 0x1C \times (m-1)$

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[7:0]							
Access																									RW							

Bit	Name	Function	Description	Access
31:8	Reserved	Reserved	-	-
7:0	RIS[7:0]	Rising Edge Detection Permission	0b0: EVNTPmn rising edge event detection disabled 0b1: EVNTPmn rising edge event detection enabled 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 EVNTPm0-7 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[7:0]							
Access																									RW							

Bit	Name	Function	Description	Access
31:8	Reserved	Reserved	-	-
7:0	FAL[7:0]	Falling Edge Detection Permission	0b0: EVNTPmn falling edge event detection disabled 0b1: EVNTPmn falling edge event detection enabled 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 EVNTPm0-7 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	0b00: HCLK 0b01: HCLK/8 0b10: HCLK/32 0b11: HCLK/64	RW
24	NFEN4	Event Port4 Digital Filter Permission	0b0: Digital filter disabled 0b1: Digital filter enabled	RW
23:19	Reserved	Reserved	-	-
18:17	DIVS3	Event Port3 Digital Filter Sampling Clock Selection	0b00: HCLK 0b01: HCLK/8 0b10: HCLK/32 0b11: HCLK/64	RW
16	NFEN3	Event Port3 Digital Filter Permission	0b0: Digital filter disabled 0b1: Digital filter enabled	RW
15:11	Reserved	Reserved	-	-
10:9	DIVS2	Event Port2 Digital Filter Sampling Clock Selection	0b00: HCLK 0b01: HCLK/8 0b10: HCLK/32 0b11: HCLK/64	RW
8	NFEN2	Event Port2 Digital Filter Permission	0b0: Digital filter disabled 0b1: Digital filter enabled	RW
7:3	Reserved	Reserved	-	-
2:1	DIVS1	Event Port1 Digital Filter Sampling Clock Selection	0b00: HCLK 0b01: HCLK/8 0b10: HCLK/32 0b11: HCLK/64	RW
0	NFEN1	Event Port1 Digital Filter Permission	0b0: Digital filter disabled 0b1: Digital filter enabled	RW

10.5 Precautions

1. Do not set the same feature to multiple ports.
2. Please switch the port function when the output latch enabled (PCRxy.LTE=0b1) to avoid unexpected glitches on the port output during switching.
3. When using the PIDRx register to read the port status, the system clock must be maintained at 20MHz or below; if you wish to read the port status at 64MHz, set the corresponding PORT group to 0b1 in the PINAER register.
4. Do not allow multiple masters to access the registers in the FASTIO domain simultaneously.

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 ; interrupt or event control functions for external pins NMI and INTP.

The main features of INTC are as follows:

1. NVIC interrupt vector: The actual number of interrupt vectors used, please refer to the interrupt vector table chapter (excluding the 16 interrupt lines of Cortex™-M0+).



Note

For more instructions on exception and NVIC programming, please refer to Chapter 5: Nested Vectored Interrupt Controller in "ARM Cortex™-M0+ Technical Reference Manual".

2. Programmable priority of NVIC: 4 programmable priorities .
3. Non-maskable interrupts: In addition to the NMI pin serving as a non-maskable interrupt source, various system interrupt event requests can also be independently selected as non-maskable interrupts. Each interrupt event request is equipped with independent enable selection, flag, and flag clear registers.
4. Equipped with 8 external EIRQ pins (INTP) for interrupts.
5. Equipped with multiple interrupt events, please refer to the "Interrupt Event List" for the specific number.
6. Interrupt wakes up system Sleep mode and Stop mode.

Table 11-1 External Interrupt Input Pin

Pin Name	I/O	Description
NMI	Input	Non-Maskable Interrupt Request Pin
INTP0	Input	External EIRQ Interrupt Request Pin 0
INTP1	Input	External EIRQ Interrupt Request Pin 1
INTP2	Input	External EIRQ Interrupt Request Pin 2
INTP3	Input	External EIRQ Interrupt Request Pin 3
INTP4	Input	External EIRQ Interrupt Request Pin 4
INTP5	Input	External EIRQ Interrupt Request Pin 5
INTP6	Input	External EIRQ Interrupt Request Pin 6
INTP7	Input	External EIRQ Interrupt Request Pin 7

11.2 Vector Digraph

11.2.1 Interrupt Vector Digraph

Table 11-2 Interrupt Vector Table

Vector Address	Vector No.	IRQ No.	Interrupt Source	Description
ARM Core Interrupt				
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	Reserved
0x00000014	5	-	ARM core	Reserved
0x00000018	6	-	ARM core	Reserved
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	Reserved
0x00000034	13	-	ARM core	Reserved
0x00000038	14	-	ARM core	Pendable request for system service (PendableSrvReq)
0x0000003C	15	-	ARM core	System tick timer (SysTick)
Non-ARM Core Interrupt				
0x00000040	16	0	INTC_INTEN.INTEN0	Interrupt Request from External EIRQ Interrupt Request Pin (INTP0/INTP1)
0x00000044	17	1	INTC_INTEN.INTEN1	Interrupt Request from External EIRQ Interrupt Request Pin (INTP2/INTP3)
0x00000048	18	2	INTC_INTEN.INTEN2	Interrupt Request from External EIRQ Interrupt Request Pin (INTP4/INTP5)
0x0000004C	19	3	INTC_INTEN.INTEN3	Interrupt Request from External EIRQ Interrupt Request Pin (INTP6/INTP7)
0x00000050	20	4	INTC_INTEN.INTEN4	Interrupt Request from DMA
0x00000054	21	5	INTC_INTEN.INTEN5	Interrupt Request from EFM
0x00000058	22	6	INTC_INTEN.INTEN6	Interrupt Request from Timer0
0x0000005C	23	7	INTC_INTEN.INTEN7	Interrupt Request from Timer2
0x00000060	24	8	INTC_INTEN.INTEN8	Interrupt Request from Timer4
0x00000064	25	9	INTC_INTEN.INTEN9	Interrupt Request from Timer4
0x00000068	26	10	INTC_INTEN.INTEN10	Interrupt Request from TimerA
0x0000006C	27	11	INTC_INTEN.INTEN11	Interrupt Request from USART_1
0x00000070	28	12	INTC_INTEN.INTEN12	Interrupt Request from USART_1
0x00000074	29	13	INTC_INTEN.INTEN13	Interrupt Request from USART_2
0x00000078	30	14	INTC_INTEN.INTEN14	Interrupt Request from USART_2
0x0000007C	31	15	INTC_INTEN.INTEN15	Interrupt Request from USART_3
0x00000080	32	16	INTC_INTEN.INTEN16	Interrupt Request from USART_3
0x00000084	33	17	INTC_INTEN.INTEN17	Interrupt Request for USART_4
0x00000088	34	18	INTC_INTEN.INTEN18	Interrupt Request for USART_4
0x0000008C	35	19	INTC_INTEN.INTEN19	Interrupt Request for SPI_1

Vector Address	Vector No.	IRQ No.	Interrupt Source	Description
0x00000090	36	20	INTC_INTEN.INTEN20	Interrupt Request for SPI_1
0x00000094	37	21	INTC_INTEN.INTEN21	Interrupt Request for SPI_2
0x00000098	38	22	INTC_INTEN.INTEN22	Interrupt Request for SPI_2
0x0000009C	39	23	INTC_INTEN.INTEN23	Interrupt Request for I2C
0x000000A0	40	24	INTC_INTEN.INTEN24	Interrupt Request for I2C
0x000000A4	41	25	INTC_INTEN.INTEN25	Interrupt Request for MCAN
0x000000A8	42	26	INTC_INTEN.INTEN26	Interrupt Request for FCM, CMU
0x000000AC	43	27	INTC_INTEN.INTEN27	Interrupt Request for EMB
0x000000B4	45	29	INTC_INTEN.INTEN29	Interrupt Request for CMP
0x000000B8	46	30	INTC_INTEN.INTEN30	Interrupt Request for ADC
0x000000BC	47	31	INTC_INTEN.INTEN31	Interrupt Request for EVENT, RTC, CTC, SWDT Stop Mode Wake-up Interrupt Request for USART, SPI

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	Interrupt Event No.	Modules	Interrupt Name	Interrupt Type ⁽¹⁾	Interrupt Source	Trigger Source	NVIC Vectors 0-31
0	0x000	PORT	PORT_EIRQ0	P/L	√	√	0
1	0x001	PORT	PORT_EIRQ1	P/L	√	√	0
2	0x002	PORT	PORT_EIRQ2	P/L	√	√	1
3	0x003	PORT	PORT_EIRQ3	P/L	√	√	1
4	0x004	PORT	PORT_EIRQ4	P/L	√	√	2
5	0x005	PORT	PORT_EIRQ5	P/L	√	√	2
6	0x006	PORT	PORT_EIRQ6	P/L	√	√	3
7	0x007	PORT	PORT_EIRQ7	P/L	√	√	3
8	0x008	DMA	DMA_TC0	P	√	√	4
9	0x009	DMA	DMA_BTC0	P	√	√	4
10	0x00A	DMA	DMA_TC1	P	√	√	4
11	0x00B	DMA	DMA_BTC1	P	√	√	4
12	0x00C	DMA	DMA_TC2	P	√	√	4
13	0x00D	DMA	DMA_BTC2	P	√	√	4
14	0x00E	DMA	DMA_TC3	P	√	√	4
15	0x00F	DMA	DMA_BTC3	P	√	√	4
16	0x010	DMA	DMA_ERR	P	√	-	4
17	0x011	EFM	EFM_PPEERR	P	√	-	5
18	0x012	EFM	EFM_PCOLERR	P	√	-	5
19	0x013	EFM	EFM_PBLKERR	P	√	-	5
20	0x014	EFM	EFM_POPTEND	P	√	√	5
21	0x015	EFM	EFM_DPEERR	P	√	-	5
22	0x016	EFM	EFM_DCOLERR	P	√	-	5
23	0x017	EFM	EFM_DBLKERR	P	√	-	5
24	0x018	EFM	EFM_DOPTEND	P	√	√	5

SN	Interrupt Event No.	Modules	Interrupt Name	Interrupt Type ⁽¹⁾	Interrupt Source	Trigger Source	NVIC Vectors 0-31
25	0x019	-	-	-	-	-	-
26	0x01A	-	-	-	-	-	-
27	0x01B	Timer0	TMR0_GCMP	P	√	√	6
28	0x01C	Timer0	TMR0_OVF	P	√	√	6
29	0x01D	Timer2	TMR2_1_CMPA	P	√	√	7
30	0x01E	Timer2	TMR2_1_CMPB	P	√	√	7
31	0x01F	Timer2	TMR2_1_OVFA	P	√	√	7
32	0x020	Timer2	TMR2_1_OVFB	P	√	√	7
33	0x021	Timer2	TMR2_2_CMPA	P	√	√	7
34	0x022	Timer2	TMR2_2_CMPB	P	√	√	7
35	0x023	Timer2	TMR2_2_OVFA	P	√	√	7
36	0x024	Timer2	TMR2_2_OVFB	P	√	√	7
37	0x025	Timer4	TMR4_GCMUH	L	√	√	8
38	0x026	Timer4	TMR4_GCMUL	L	√	√	8
39	0x027	Timer4	TMR4_GCMVH	L	√	√	8
40	0x028	Timer4	TMR4_GCMVL	L	√	√	8
41	0x029	Timer4	TMR4_GCMWH	L	√	√	8
42	0x02A	Timer4	TMR4_GCMWL	L	√	√	8
43	0x02B	Timer4	TMR4_GOVF	L	√	√	8
44	0x02C	Timer4	TMR4_GUDF	L	√	√	8
45	0x02D	Timer4	TMR4_GRLU	L	√	√	8
46	0x02E	Timer4	TMR4_GRLV	L	√	√	8
47	0x02F	Timer4	TMR4_GRLW	L	√	√	8
48	0x030	Timer4	TMR4_SCM0	P	√	√	9
49	0x031	Timer4	TMR4_SCM1	P	√	√	9
50	0x032	Timer4	TMR4_SCM2	P	√	√	9
51	0x033	Timer4	TMR4_SCM3	P	√	√	9
52	0x034	Timer4	TMR4_SCM4	P	√	√	9
53	0x035	Timer4	TMR4_SCM5	P	√	√	9
54	0x036	TimerA	TMRA_OVF	P	√	√	10
55	0x037	TimerA	TMRA_UDF	P	√	√	10
56	0x038	TimerA	TMRA_CMP	P	√	√	10
57	0x039	-	-	-	-	-	-
58	0x03A	-	-	-	-	-	-
59	0x03B	-	-	-	-	-	-
60	0x03C	USART	USART_1_RI	P	√	√	11
61	0x03D	USART	USART_1_TI	P	√	√	11
62	0x03E	USART	USART_1_EI	L	√	√	12
63	0x03F	USART	USART_1_TENDI	L	√	√	12
64	0x040	USART	USART_1_TCI	L	√	√	12
65	0x041	USART	USART_1_RTO	L	√	√	12
66	0x042	USART	USART_2_RI	P	√	√	13
67	0x043	USART	USART_2_TI	P	√	√	13
68	0x044	USART	USART_2_EI	L	√	√	14
69	0x045	USART	USART_2_TENDI	L	√	√	14

SN	Interrupt Event No.	Modules	Interrupt Name	Interrupt Type ⁽¹⁾	Interrupt Source	Trigger Source	NVIC Vectors 0-31
70	0x046	USART	USART_2_TCI	L	√	√	14
71	0x047	USART	USART_2_RTO	L	√	√	14
72	0x048	USART	USART_3_RI	P	√	√	15
73	0x049	USART	USART_3_TI	P	√	√	15
74	0x04A	USART	USART_3_EI	L	√	√	16
75	0x04B	USART	USART_3_TENDI	L	√	√	16
76	0x04C	USART	USART_3_TCI	L	√	√	16
77	0x04D	USART	USART_3_RTO	L	√	√	16
78	0x04E	USART	USART_3_BRKWKPI	L	√	√	16
79	0x04F	USART	USART_4_RI	P	√	√	17
80	0x050	USART	USART_4_TI	P	√	√	17
81	0x051	USART	USART_4_EI	L	√	√	18
82	0x052	USART	USART_4_TENDI	L	√	√	18
83	0x053	USART	USART_4_TCI	L	√	√	18
84	0x054	USART	USART_4_RTO	L	√	√	18
85	0x055	USART	USART_4_BRKWKPI	L	√	√	18
86	0x056	SPI	SPI_1_SPRI	P	√	√	19
87	0x057	SPI	SPI_1_SPTI	P	√	√	19
88	0x058	SPI	SPI_1_SPEND	-	-	√	-
89	0x059	SPI	SPI_1_SPII	L	√	√	20
90	0x05A	SPI	SPI_1_SPEI	L	√	√	20
91	0x05B	SPI	SPI_2_SPRI	P	√	√	21
92	0x05C	SPI	SPI_2_SPTI	P	√	√	21
93	0x05D	SPI	SPI_2_SPEND	-	-	√	-
94	0x05E	SPI	SPI_2_SPII	L	√	√	22
95	0x05F	SPI	SPI_2_SPEI	L	√	√	22
96	0x060	I2C	I2C_RXI	P	√	√	23
97	0x061	I2C	I2C_TXI	P	√	√	23
98	0x062	I2C	I2C_TEI	L	√	√	24
99	0x063	I2C	I2C_EEI	L	√	√	24
100	0x064	-	-	-	-	-	-
101	0x065	-	-	-	-	-	-
102	0x066	CANRAM	CANRAM_ECCER	L	√	-	25
103	0x067	MCAN	MCAN_INT0	L	√	-	25
104	0x068	MCAN	MCAN_INT1	L	√	-	25
105	0x069	FCM	FCM_FERRI	L	√	-	26
106	0x06A	FCM	FCM_MENDI	L	√	-	26
107	0x06B	FCM	FCM_COVFI	L	√	-	26
108	0x06C	XTAL	XTAL_STOP	L	√	-	26
109	0x06D	EMB	EMB_GR	L	√	-	27
110	0x06E	-	-	-	-	-	-
111	0x06F	-	-	-	-	-	-
112	0x070	CMP	CMP_IRQ	P	√	√	29
113	0x071	ADC	ADC_EOCA	P	√	√	30
114	0x072	ADC	ADC_EOCB	P	√	√	30

SN	Interrupt Event No.	Modules	Interrupt Name	Interrupt Type ⁽¹⁾	Interrupt Source	Trigger Source	NVIC Vectors 0-31
115	0x073	ADC	ADC_CMP0	P	√	√	30
116	0x074	ADC	ADC_CMP1	P	√	√	30
117	0x075	EVENT	EVENT_PORT1	P	√	√	31
118	0x076	EVENT	EVENT_PORT2	P	√	√	31
119	0x077	EVENT	EVENT_PORT3	P	√	√	31
120	0x078	EVENT	EVENT_PORT4	P	√	√	31
121	0x079	RTC	RTC_ALM	P	√	√	31
122	0x07A	RTC	RTC_PRD	P	√	√	31
123	0x07B	RTC	RTC_IT	P	√	-	31
124	0x07C	AOS	AOS_STRG	-	-	√	-
125	0x07D	CTC	CTC_ERR	P	√	-	31
126	0x07E	SWDT	SWDT_REFUDF	P	√	-	31
127	0x07F	USART SPI	USART_3_WUPI USART_4_WUPI SPI_1_WUPI	Edge	√	-	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.

11.3 Functional Description

11.3.1 Function Block Diagram

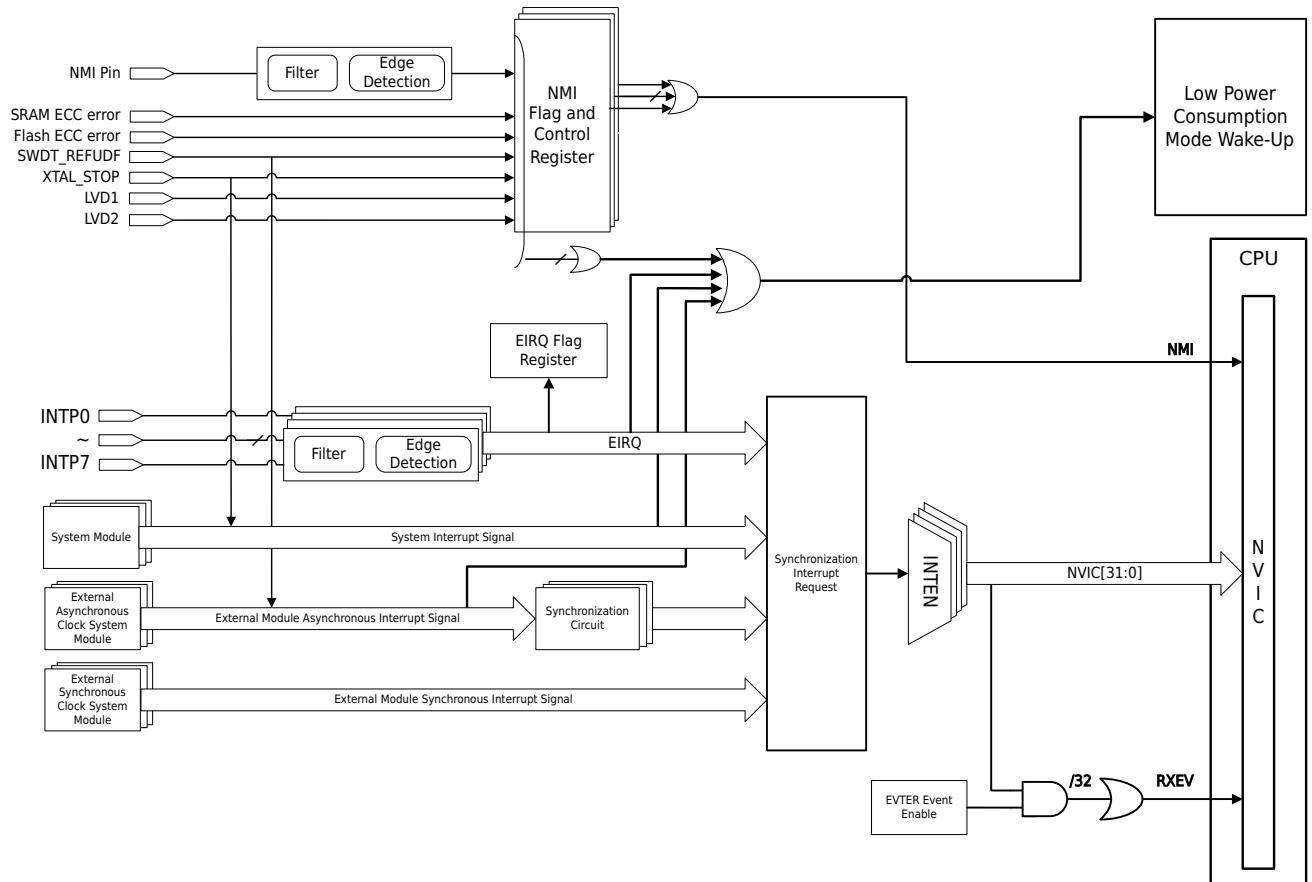


Figure 11-1 Interrupt System Block Diagram

11.3.2 Non-maskable Interrupt

The non-maskable interrupt events are as follows:

- NMI pin interrupt
- XTAL oscillation fault interrupt
- SWDT underflow/refresh error interrupt
- Low voltage detection LVD1 interrupt
- Low voltage detection LVD2 interrupt
- Flash ECC check error interrupt
- SRAM ECC parity 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 EIRQ Pin Interrupt

To use an external EIRQ pin (INTP) interrupt request, configure it as follows:

1. Clear the EIRQCRx.EIRQFEN bit (x=0-7) to disable the digital filter.
2. Set the EIRQCRx.EIRQTRG bit to select the trigger edge or level; set the EIRQCRx.EIRQFCLK bit to select the digital filter sampling clock; set the EIRQCRx.EIRQFEN bit (x=0-7) to enable the digital filter.

11.3.4 Software Trigger Event

The AOS_STRG event, numbered 124 in the interrupt event request list, can be triggered by setting the STRG bit in the Software Trigger Event Control Register (AOS_INTSFTTRG). This event can be used to trigger other peripheral modules, such as ADC, Timer, etc. For details, see the relevant sections.

**Note**

This register is not protected by the Function Protection Register (INTC_FPRCR). When using the software trigger event function, you must first enable the peripheral circuit trigger function. The enabling method is to clear the FCG.AOS bit in the Function Clock Control Register.

11.3.5 Interrupt Source Selection

One or multiple peripheral interrupt event requests share one interrupt vector. Each peripheral can request this interrupt and be distinguished by its own flag bit. The interrupt request for this vector can be enabled or disabled via the interrupt enable register INTC_INTEN.

For the specific interrupt vector allocation, see "[Interrupt Event Table](#)".

11.3.6 WFE Event Selection

All interrupt event requests can be configured as event inputs to wake up the core (WFE) by setting the event enable register INTC_EVTER. In this case, the corresponding interrupt inputs for interrupt vectors Vec0-31 become invalid. The specific block diagram is as follows:

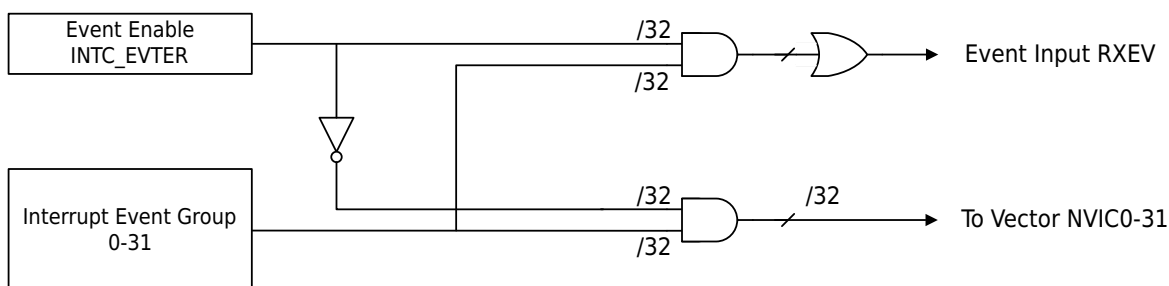


Figure 11-2 Interrupt Event Selection

11.3.7 WFE Wake-up Event Management

This series of products can wake up the core (WFE) through event inputs. Event input wake-up of the core can be generated in the following two ways:

- Cortex™-M0+ System Control Register SCR.SEVONPEND=1

Set SCR.SEVONPEND=1, enable an interrupt in the peripheral control register, set INTC_INTEN according to the selected interrupt vector, but do not enable the corresponding vector in the NVIC. When the MCU recovers from WFE, it is necessary to clear the interrupt flag bit of the corresponding peripheral and the NVIC interrupt flag register.

The WFE Stop mode entry and wake-up flow is as follows (using INTP0 as an example):

1. Set the Stop mode register (see the register description in the PWC chapter);
2. Set the wake-up event enable register INTC_WKEN;
3. Set the pin INTP0 input and the EIRQ pin interrupt control register INTC_EIRQCR0;
4. Set the SEVONPEND bit in SCR to "1";
5. Perform the following operations to ensure the system enters Stop mode:
 - ▶ _SEV(); Set the internal event register
 - ▶ _WFE(); Clear the event register
 - ▶ _WFE(); The system enters Stop mode
6. Wait for the selected interrupt event request to occur. The system will wake up from Stop mode but will not enter the interrupt service routine.

- Cortex™-M0+ System Control Register SCR.SEVONPEND=0

Set SCR.SEVONPEND=0, enable an interrupt in the peripheral control register, set INTC_INTEN according to the selected interrupt vector, and simultaneously set the corresponding bit in the event enable register INTC_EVTER. When the CPU recovers from WFE, it is necessary to clear the interrupt flag bit of the corresponding peripheral, but it is not necessary to clear the NVIC interrupt flag register.

The WFE Stop mode entry and wake-up flow is as follows (using INTP0 as an example):

1. Set the Stop mode register (see the register description in the PWC chapter);
2. Set the wake-up event enable register INTC_WKEN;
3. Set the pin INTP0 input and the EIRQ pin interrupt control register INTC_EIRQCR0;
4. Set the INTC_INTEN register to enable the corresponding interrupt event request (the reset value is valid);
5. Enable the corresponding event request in INTC_EVTER;
6. Perform the following operations to ensure the system enters Stop mode:
 - ▶ _SEV(); Set the internal event register
 - ▶ _WFE(); Clear the event register
 - ▶ _WFE(); The system enters Stop mode
7. Wait for the selected interrupt event request to occur. The system will wake up from Stop mode but will not enter the interrupt service routine.

11.3.8 Digital Filter

For the INTP_x (x=0-7) pin input, a digital filter can be selected for noise filtering. The filter's sampling clock is HCLK, and input signals shorter than 3 filter periods will be filtered out. The specific operation timing diagram is as follows:

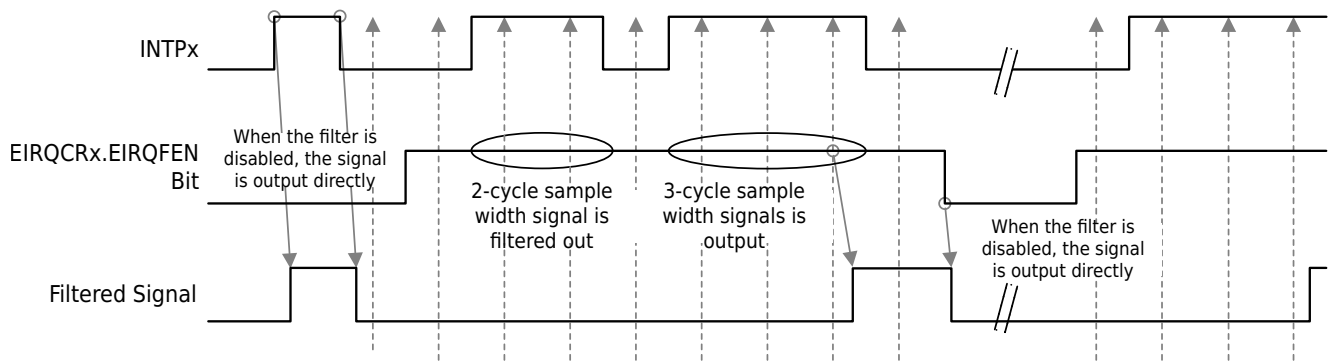


Figure 11-3 Digital Filter Operation Schematic Diagram

Before entering Stop mode, set `INTC_EIRQCRx.EIRQFEN=0` ($x=0-7$) to disable the digital filter. Re-enable the digital filter after returning from Stop mode. The configuration procedure is as follows:

1. Set the system stop mode register.
2. Configure the stop mode wake-up interrupt.
3. Stop the digital filter.
4. Execute WFI to enter stop mode.

11.3.9 Low Power Mode Return

Sleep Mode Return

When selecting an event interrupt source as the return condition from sleep mode, the following settings are required:

- Select the event as a CPU interrupt source.
- Enable the control register in the NVIC.
- If a non-maskable interrupt (NMI) is required, configure the `INTC_NMIER` enable register.

Stop Mode Return

You can select either a non-maskable event interrupt source or a maskable event source configured in the `INTC_WKEN` register as the return condition from stop mode.

The following settings are required to return from stop mode:

- Select an event interrupt source as the return condition from stop mode.
 1. For non-maskable interrupts (NMI), configure via the `INTC_NMIER` enable register.
 2. For maskable interrupts, configure via the `INTC_WKEN` enable register.
- Select the event as a CPU interrupt source.
- Enable the control register in the NVIC.

For EIRQ pins that are not selected, they will not be detected because their clock is turned off. Before executing the WFI instruction, ensure all status bits in the Non-Maskable Interrupt Flag Register (`INTC_NMIFR`) are "0".

11.4 Register Description

11.4.1 Register List

Base Address: 0x40011000

Table 11-4 Interrupt Controller (INTC) Register Offset

Offset	Register	Description	Width
0x003C	INTC_FPRCR	Function Protection Control Register	32
0x0000	INTC_NMICR	NMI Pin Interruption 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
0x0040+0x4*x	INTC_EIRQCRx	EIRQ Pin Interrupt Control Register x (x=0-7)	32
0x0074	INTC_EIFR	EIRQ Pin Interrupt Flag Register	32
0x0078	INTC_EIFCR	EIRQ Pin Interrupt Flag Clear Register	32
0x0080+0x4*x	INTC_INTENx	Interrupt Event Enable Register x (x=0-3)	32
0x0070	INTC_WKEN	Wake-up Event Enable Register	32
0x0020	INTC_EVTER	WFE Event Enable Register	32
0x0010	INTC_WKFR	USART/SPI Wake-up Event Flag Register	32
0x0014	INTC_WKFCR	USART/SPI Wake-up Event Flag Clear Register	32

11.4.2 Function Protection Control Register (INTC_FPRCR)

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																								FPRC							
Access																									RW							

Bit	Name	Function	Description	Access
31:8	Reserved	Reserved	-	-
7:0	FPRC	Function Protection Bit	When the bit is written with A5, write operations can be performed on other registers of the INTC; otherwise, the write operation is invalid.	RW

11.4.3 NMI Pin Interrupt Control Register (INTC_NMICR)

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	0x000000XX																															
Name	Reserved																								NMIFEN	Reserved	NMIFCLK	Reserved		NMITRG		
Access																									RW		RW			RW		

Bit	Name	Function	Description	Access
31:8	Reserved	Reserved	-	-
7	NMIFEN	NMI Pin Digital Filter Enable	0b0: Disable digital filter function 0b1: Enable digital filter function	RW
6	Reserved	Reserved	-	-
5:4	NMIFCLK	NMI Pin Filter Sampling Cycle Selection	0b00: 1 ×HCLK period 0b01: 8 ×HCLK periods 0b10: 32 ×HCLK periods 0b11: 64 ×HCLK periods	RW
3:1	Reserved	Reserved	-	-
0	NMITRG	NMI Pin Trigger Edge Selection	0b0: Falling edge 0b1: Rising edge	RW

**Notice**

When ICGA1x.NMIICGEN=1, a write to the INTC_NMICR register is valid, and the reset value of INTC_NMICR is 0x00000000.

When ICGA1x.NMIICGEN=0, a write to the INTC_NMICR register is invalid. The reset value of INTC_NMICR is determined by ICGA1x:

- NMIFEN is the setting value of ICGA1x.NMIFEN;
- NMIFCLK is the setting value of ICGA1x.NMIFCLK;
- NMITRG is the setting value of ICGA1x.NMITRG;
- When ICGA1x[31:26]=0b011100, please keep the NMI pin at a low level during the reset period; otherwise, an unnecessary NMI interrupt will be generated after the reset is released.

11.4.4 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	0x0000000X																																					
Name	Reserved																								FECCERREN		RECCERREN		LVD2EN		LVD1EN		SWDTEN		XATLSTPEN		NMIEN	
Access																									RW		RW		RW		RW		RW		RW		RW	

Bit	Name	Function	Description	Access
31:7	Reserved	Reserved	-	-
6	FECCERREN	Flash ECC Check Error Interrupt Selection	0b0: Disable interrupt as non-maskable interrupt source 0b1: Select the interrupt as a non-maskable interrupt source	RW
5	RECCERREN	SRAM ECC Check Error Interrupt Selection	0b0: Disable interrupt as non-maskable interrupt source 0b1: Select the interrupt as a non-maskable interrupt source	RW
4	LVD2EN	LVD2 Interrupt Selection	0b0: Disable interrupt as non-maskable interrupt source 0b1: Select the interrupt as a non-maskable interrupt source	RW
3	LVD1EN	LVD1 Interrupt Selection	0b0: Disable interrupt as non-maskable interrupt source 0b1: Select the interrupt as a non-maskable interrupt source	RW
2	SWDTEN	SWDT Underflow/Refresh Error Interrupt Selection	0b0: Disable interrupt as non-maskable interrupt source 0b1: Select the interrupt as a non-maskable interrupt source	RW
1	XATLSTPEN	XTAL Oscillation Failure Interrupt Selection	0b0: Disable interrupt as non-maskable interrupt source 0b1: Select the interrupt as a non-maskable interrupt source	RW
0	NMIEN	NMI Pin Interrupt Selection	0b0: Disable interrupt as non-maskable interrupt source 0b1: Select the interrupt as a non-maskable interrupt source	RW

**Notice**

When ICGA1x.NMIICGEN=1, writing to the NMIEN bit of this register is valid, and the reset value of NMIEN is 0.

When ICGA1x.NMIICGEN=0, writing to the NMIEN bit of this register is invalid, and the reset value of NMIEN is determined by ICGA1x.NMIEN.

11.4.5 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																										FECERRF	RECCERRF	LVD2F	LVD1F	SWDTF	XATLSTPF	NMIF
Access																											RO	RO	RO	RO	RO	RO	RO

Bit	Name	Function	Description	Access
31:7	Reserved	Reserved	-	-
6	FECCERRF	Flash ECC Check Error Interrupt Flag	0b0: No Flash ECC check error interrupt request occurred 0b1: Flash ECC check error interrupt request occurred	RO
5	RECCERRF	SRAM ECC Check Error Interrupt Flag	0b0: No SRAM ECC check error interrupt request occurred 0b1: SRAM ECC check error interrupt request occurred	RO
4	LVD2F	LVD2 Interrupt Flag	0b0: No LVD2 interrupt request occurred 0b1: LVD2 interrupt request occurred	RO
3	LVD1F	LVD1 Interrupt Flag	0b0: No LVD1 interrupt request occurred 0b1: LVD1 interrupt request occurred	RO
2	SWDTF	SWDT Underflow/Refresh Error Interrupt Flag	0b0: No SWDT underflow/refresh request occurred 0b1: SWDT underflow/refresh request occurred	RO
1	XTALSTPF	XTAL Oscillation Fault Interrupt Flag	0b0: No XTAL oscillation failure interrupt request occurred 0b1: XTAL oscillation failure interrupt request occurred	RO
0	NMIF	NMI Pin Interrupt Flag	0b0: No NMI pin interrupt request occurred 0b1: NMI pin interrupt request occurred	RO

11.4.6 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																										FECERRCLR						RECCERRCLR						LVD2CLR						LVD1CLR						SWDTCLR						XATLSTPCLR						NMICLR					
Access																											RO						RO						RO						RO						RO						RO						RO					

Bit	Name	Function	Description	Access
31:7	Reserved	Reserved	-	-
6	FECCERRFCLR	Flash ECC Check Error Interrupt Flag Clear	Write 1 to clear the flag, write 0 has no effect; read always returns 0.	WO
5	RECCERRFCLR	SRAM ECC Parity Error Interrupt Flag Clear	Write 1 to clear the flag, write 0 has no effect; read always returns 0.	WO
4	LVD2FCLR	LVD2 Interrupt Flag Clear	Write 1 to clear the flag, write 0 has no effect; read always returns 0.	WO
3	LVD1FCLR	LVD1 Interrupt Flag Clear	Write 1 to clear the flag, write 0 has no effect; read always returns 0.	WO
2	SWDTFCLR	SWDT Under-flow/Refresh Error Interrupt Flag Clear	Write 1 to clear the flag, write 0 has no effect; read always returns 0.	WO
1	XTALSTPFCLR	XTAL Oscillation Failure Interrupt Flag Clear	Write 1 to clear the flag, write 0 has no effect; read always returns 0.	WO
0	NMIFCLR	NMI Pin Interrupt Flag Clear	Write 1 to clear the flag, write 0 has no effect; read always returns 0.	WO

11.4.7 EIRQ Pin Interrupt Control Register (INTC_EIRQCRx) (x=0-7)

Offset: 0x0040+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	0x00000002																															
Name	Reserved																								EIRQFEN	Reserved	EIRQFCLK	Reserved	EIRQTRG			
Access																									RW		RW		RW			

Bit	Name	Function	Description	Access
31:8	Reserved	Reserved	-	-
7	EIRQFEN	EIRQ Digital Filter Enabled	0b0: Disable digital filter function 0b1: Enable digital filter function	RW
6	Reserved	Reserved	-	-
5:4	EIRQFCLK	EIRQ Digital Filter Sampling Period Selection	0b00: 1 × HCLK period 0b01: 8 × HCLK period 0b10: 32 × HCLK period 0b11: 64 × HCLK period	RW
3:2	Reserved	Reserved	-	-
1:0	EIRQTRG	Trigger Edge Selection	0b00: Low level 0b01: Rising edge 0b10: Falling edge 0b11: Double edge	RW

11.4.8 EIRQ Pin Interrupt Flag Register (INTC_EIFR)

Offset	Bit Position																															
0x0074	31	30	29	28	27	26	25	24	23	22	21	20	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							
Access																									RO							

Bit	Name	Function	Description	Access
31:8	Reserved	Reserved	-	-
7:0	EIF	EIRQ Flag Bit	Corresponding to EIF7-EIF0: 0b0: No EIRQ pin input event occurred 0b1: An EIRQ pin input event occurred  Note Each bit corresponds to one EIRQ number.	RO

11.4.9 EIRQ Pin Interrupt Flag Clear Register (INTC_EIFCR)

Offset	Bit Position																															
0x0078	31	30	29	28	27	26	25	24	23	22	21	20	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
31:8	Reserved	Reserved	-	-
7:0	EIFCLR	EIRQ Flag Clear Bit	Corresponding to EIFCLR7-EIFCLR0, write 1 to clear the corresponding bit of the EIFR register, write 0 has no effect; read always returns 0.  Note Each bit corresponds to one EIRQ number.	WO

11.4.10 Interrupt Event Enable Register (INTC_INTENx) (x=0-3)

Offset: 0x0080+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	0xFFFFFFFF																															
Name	INTENy																															
Access	RW																															

Bit	Name	Function	Description	Access
31:0	INTENy (y=31-0)	Interrupt Event Enabled	Corresponding interrupt event number z (z=x*32+y) ● 0b0: Interrupt event disabled ● 0b1: Interrupt event enabled  Note Interrupt events set to "0" will not be received by the NVIC interrupt request.	RW

11.4.11 Wake-up Event Enable Register (INTC_WKEN)

Offset	Bit Position																															
0x0070	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x00000000																															
Name	Reserved	USART4WKEN	USART3WKEN	SPI1WKEN	Reserved	TMR2OVFWKEN	TMR2GCMPWKEN	TMR0CMPWKEN	RTCPRDWKEN	RTCALMWKEN	Reserved	CMP1WKEN	Reserved	SWDTWKEN	Reserved								EIRQWKEN									
Access		RW	RW	RW		RW	RW	RW	RW	RW		RW			RW									RW								

Bit	Name	Function	Description	Access
31:30	Reserved	Reserved	-	-
29	USART4WKEN	USART_4_RX Line Stop Mode Wake-up Enabled	0b0: Wake-up disabled 0b1: Wake-up enabled	RW
28	USART3WKEN	USART_3_RX Line Stop Mode Wake-up Enabled	0b0: Wake-up disabled 0b1: Wake-up enabled	RW
27	SPI1WKEN	SPI_1_SSL0 Line Stop Mode Wake-up Enabled	0b0: Wake-up disabled 0b1: Wake-up enabled	RW
26	Reserved	Reserved	-	-
25	TMR2OVFWKEN	TMR2_OVFA Stop Mode Wake-up Enabled	0b0: Wake-up disabled 0b1: Wake-up enabled	RW
24	TMR2GCMPPWKEN	TMR2_GCMA Stop Mode Wake-up Enabled	0b0: Wake-up disabled 0b1: Wake-up enabled	RW
23	TMR0CMPWKEN	TMR0_1_CMPA Stop Mode Wake-up Enabled	0b0: Wake-up disabled 0b1: Wake-up enabled	RW
22	RTCPRDWKEN	RTC_PRD Stop Mode Wake-up Enabled	0b0: Wake-up disabled 0b1: Wake-up enabled	RW
21	RTCALMWKEN	RTC_ALM Stop Mode Wake-up Enabled	0b0: Wake-up disabled 0b1: Wake-up enabled	RW

Bit	Name	Function	Description	Access
20	<i>Reserved</i>	<i>Reserved</i>	-	-
19	CMP1WKEN	CMP1 Stop Mode Wake-up Enabled	0b0: Wake-up disabled 0b1: Wake-up enabled	RW
18:17	<i>Reserved</i>	<i>Reserved</i>	-	-
16	SWDTWKEN	SWDT_REFUDF Stop Mode Wake-up Enabled	0b0: Wake-up disabled 0b1: Wake-up enabled	RW
15:8	<i>Reserved</i>	<i>Reserved</i>	-	-
7:0	EIRQWKEN	EIRQ Stop Mode Wake-up Enabled	Corresponding to EIRQWKEN7-EIRQWKEN0: 0b0: Wake-up disabled 0b1: Wake-up enabled	RW

11.4.12 WFE Event Enable Register (INTC_EVTER)

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	EVTEN																															
Access	RW																															

Bit	Name	Function	Description	Access
31:0	EVTEN	Event Enabled	Corresponding to EVTEN31-EVTEN0: 0b0: IRQ is sent to NVIC as an interrupt input 0b1: IRQ is sent to NVIC as an event input  Note Each bit corresponds to an IRQ number.	RW

11.4.13 USART/SPI Wake-up Event Flag Register (INTC_WKFR)

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	USART4WKF	USART3WKF	SPI1WKF	Reserved																											
Access		RO	RO	RO																												

Bit	Name	Function	Description	Access
31:30	Reserved	Reserved	-	-
29	USART4WKF	USART_4_RX Line Stop Mode Wake-up Flag Bit	0b0: No USART_4_RX line stop mode wake-up request occurred 0b1: USART_4_RX line stop mode wake-up request occurred	RO
28	USART3WKF	USART_3_RX Line Stop Mode Wake-up Flag Bit	0b0: No USART_3_RX line stop mode wake-up request occurred 0b1: USART_3_RX line stop mode wake-up request occurred	RO
27	SPI1WKF	SPI_1_SSL0 Line Stop Mode Wake-up Flag Bit	0b0: No SPI_1_SSL0 line stop mode wake-up request occurred 0b1: SPI_1_SSL0 line stop mode wake-up request occurred	RO
26:0	Reserved	Reserved	-	-

11.4.14 USART/SPI Wake-up Event Flag Clear Register (INTC_WKFCR)

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		USART4WKFCR		USART3WKFCR		SPI1WKFCR		Reserved																							
Access			WO		WO		WO																									

Bit	Name	Function	Description	Access
31:30	Reserved	Reserved	-	-
29	USART4WKFCR	USART_4_RX Line Stop Mode Wake-up Flag Bit Clear	Write 1 to clear the flag, write 0 has no effect; read always returns 0	WO
28	USART3WKFCR	USART_3_RX Line Stop Mode Wake-up Flag Bit Clear	Write 1 to clear the flag, write 0 has no effect; read always returns 0	WO
27	SPI1WKFCR	SPI_1_SSL0 Line Stop Mode Wake-up Flag Bit Clear	Write 1 to clear the flag, write 0 has no effect; read always returns 0	WO
26:0	Reserved	Reserved	-	-

11.5 Precautions

For details regarding ARM core interrupts, please refer to the ARM manual, Cortex-M0+ Technical Reference Manual.

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.

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. 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 3 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 DMA trigger targets
- 1 Timer0 trigger target
- 2 Timer2 trigger targets
- 1 Timer4 trigger target
- 2 TimerA trigger targets
- 2 Event Port trigger targets
- 2 ADC trigger targets

12.3 Functional Description

12.3.1 Function Block Diagram

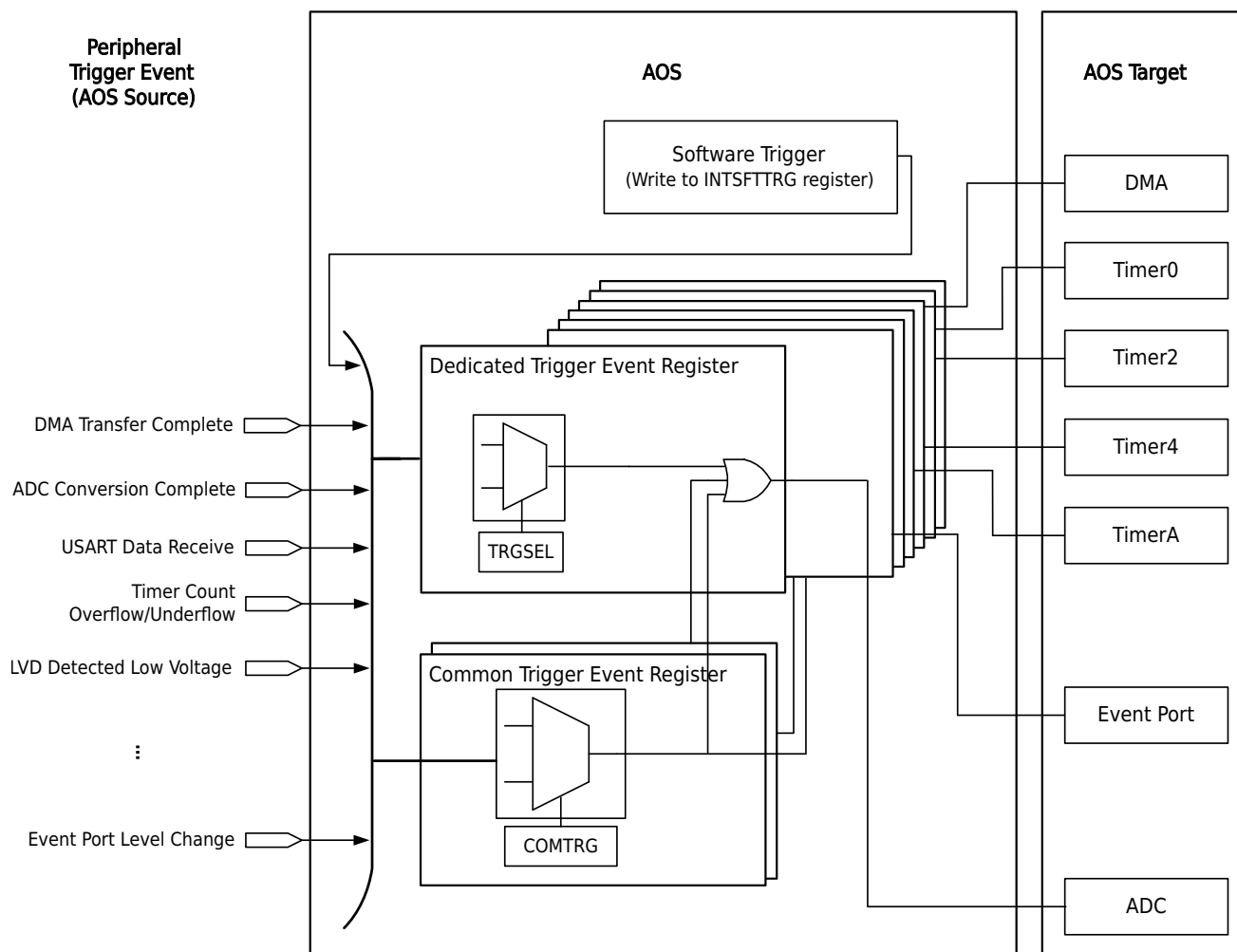


Figure 12-1 AOS Module Block Diagram

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
Timer0	0	Trigger counter start/stop/clear/capture
Timer2	1	Trigger channel A counter start/stop/clear/capture
	2	Trigger channel B counter start/stop/clear/capture
Timer4	3	Trigger counter start/stop/clear/capture
TimerA	4	Trigger counter start/stop/clear/increment/decrement
	5	Trigger counter input capture
ADC	6, 7	Trigger sequence A/B analog-to-digital conversion start

Modules	SN	Action
DMA	8	Trigger channel 0 starting transmission
	9	Trigger channel 1 starting transmission
	10	Trigger channel 2 starting transmission
	11	Trigger channel 3 starting transmission
Event Port	12	Trigger Event Port 1/2 action
	13	Trigger Event Port 3/4 action

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.

12.4 Register Description

12.4.1 Register List

Base address: 0x40000C00

Table 12-2 Automatic Operation System (AOS) Register Offset

Offset	Register	Description	Width
0x00	INTSFTTRG	Software Trigger Event Register	32
0x24+0x04*x	DMA_TRGSELx	DMA Transfer Start Trigger Source Selection Register (x=0-3)	32
0x34	PEVNT_TRGSEL12	Event Port1/2 Trigger Source Selection Register	32
0x38	PEVNT_TRGSEL34	Event Port3/4 Trigger Source Selection Register	32
0x04	TMR0_TRGSEL	Timer0 Hardware Trigger Event Selection Register	32
0x08+0x04*x	TMR2_TRGSELx	Timer2 Hardware Trigger Event Selection Register (x=0-1)	32
0x10	TMR4_TRGSEL	Timer4 Hardware Trigger Event Selection Register	32
0x14+0x04*x	TMRA_TRGSELx	TimerA Hardware Trigger Event Selection Register (x=0-1)	32
0x1C+0x04*x	ADC_TRGSELx	ADC Conversion Start On-Chip Trigger Source Selection Register (x=0-1)	32
0x3C	AOS_COMTRGSEL1	Common Trigger Source Selection Register 1	32
0x40	AOS_COMTRGSEL2	Common Trigger Source Selection Register 2	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 DMA Transfer Start Trigger Source Selection Register (DMA_TRGSELx) (x=0-3)

After enabling DMA, please use this register to select the trigger event source. When the peripheral event corresponding to the assigned number occurs, the DMA will be triggered by this event to initiate and carry out the data transfer.

Each data transfer channel corresponds to one register.

Offset: 0x24+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	0x0000007F																																
Name	COMEN	Reserved																								TRGSEL							
Access	RW																									RW							

Bit	Name	Function	Description	Access
31	COMEN[1]	Common Trigger Enabled	0b0: Disable triggering of this AOS target by the common trigger event of AOS_COMTRGSEL2 0b1: Enable triggering of this AOS target by the common trigger event of AOS_COMTRGSEL2	RW
30	COMEN[0]	Common Trigger Enabled	0b0: Disable triggering of this AOS target by the common trigger event of AOS_COMTRGSEL1 0b1: Enable triggering of this AOS target by the common trigger event of AOS_COMTRGSEL1	RW
29:7	Reserved	Reserved	-	-
6:0	TRGSEL	Trigger Source Selection	Write the number of the peripheral circuit event to be selected	RW

12.4.4 Event Port Trigger Source Selection Register (PEVNT_TRGSEL12/PEVNT_TRGSEL34)

Write the trigger event number to the TRGSEL[6:0] bits in this register to select the trigger event source. Event Port1/2 share the PEVNT_TRGSEL12 register, and Event Port3/4 share the PEVNT_TRGSEL34 register.

Offset: 0x34 (PEVNT_TRGSEL12), 0x38 (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	0x0000007F																																
Name	COMEN	Reserved																								TRGSEL							
Access	RW																									RW							

Bit	Name	Function	Description	Access
31	COMEN[1]	Common Trigger Enabled	0b0: Disable triggering of this AOS target by the common trigger event of AOS_COMTRGSEL2 0b1: Enable triggering of this AOS target by the common trigger event of AOS_COMTRGSEL2	RW
30	COMEN[0]	Common Trigger Enabled	0b0: Disable triggering of this AOS target by the common trigger event of AOS_COMTRGSEL1 0b1: Enable triggering of this AOS target by the common trigger event of AOS_COMTRGSEL1	RW
29:7	Reserved	Reserved	-	-
6:0	TRGSEL	Trigger Source Selection	Write the number of the peripheral circuit event to be selected	RW

12.4.5 Timer0 Hardware Trigger Event Selection Register (TMR0_TRGSEL)

After selecting hardware trigger in Timer0, use this register to select the trigger event source. Write the trigger event number to the TRGSEL[6:0] bits in this register. When the peripheral event corresponding to the number occurs, Timer0 will be triggered to act (including start/stop/clear/input capture) by this event.

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	0x0000007F																																
Name	COMEN	Reserved																								TRGSEL							
Access	RW																									RW							

Bit	Name	Function	Description	Access
31	COMEN[1]	Common Trigger Enabled	0b0: Disable triggering of this AOS target by the common trigger event of AOS_COMTRGSEL2 0b1: Enable triggering of this AOS target by the common trigger event of AOS_COMTRGSEL2	RW
30	COMEN[0]	Common Trigger Enabled	0b0: Disable triggering of this AOS target by the common trigger event of AOS_COMTRGSEL1 0b1: Enable triggering of this AOS target by the common trigger event of AOS_COMTRGSEL1	RW
29:7	Reserved	Reserved	-	-
6:0	TRGSEL	Trigger Source Selection	Write the number of the peripheral circuit event to be selected	RW

12.4.6 Timer2 Hardware Trigger Event Selection Register (TMR2_TRGSELx) (x=0-1)

After selecting hardware trigger in Timer2, use this register to select the trigger event source. Write the trigger event number to the TRGSEL[6:0] bits in this register. 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 channel A of Timer2 units share the TMR2_TRGSEL0 register, and all channel B of Timer2 units share the TMR2_TRGSEL1 register.

Offset: 0x08+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	0x0000007F																																					
Name	COMEN	Reserved																																				TRGSEL
Access	RW																																					RW

Bit	Name	Function	Description	Access
31	COMEN[1]	Common Trigger Enabled	0b0: Disable triggering of this AOS target by the common trigger event of AOS_COMTRGSEL2 0b1: Enable triggering of this AOS target by the common trigger event of AOS_COMTRGSEL2	RW
30	COMEN[0]	Common Trigger Enabled	0b0: Disable triggering of this AOS target by the common trigger event of AOS_COMTRGSEL1 0b1: Enable triggering of this AOS target by the common trigger event of AOS_COMTRGSEL1	RW
29:7	Reserved	Reserved	-	-
6:0	TRGSEL	Trigger Source Selection	Write the number of the peripheral circuit event to be selected	RW

12.4.7 Timer4 Hardware Trigger Event Selection Register (TMR4_TRGSEL)

After selecting internal trigger hardware synchronization in Timer4, use this register to select the trigger event source. Write the trigger event number to the TRGSEL[6:0] bits in this register. When the peripheral circuit event corresponding to the number occurs, Timer4 will be triggered for hardware synchronization by this event.

All Timer4 units and their channels share 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	0x0000007F																																
Name	COMEN	Reserved																								TRGSEL							
Access	RW																									RW							

Bit	Name	Function	Description	Access
31	COMEN[1]	Common Trigger Enabled	0b0: Disable triggering of this AOS target by the common trigger event of AOS_COMTRGSEL2 0b1: Enable triggering of this AOS target by the common trigger event of AOS_COMTRGSEL2	RW
30	COMEN[0]	Common Trigger Enabled	0b0: Disable triggering of this AOS target by the common trigger event of AOS_COMTRGSEL1 0b1: Enable triggering of this AOS target by the common trigger event of AOS_COMTRGSEL1	RW
29:7	Reserved	Reserved	-	-
6:0	TRGSEL	Trigger Source Selection	Write the number of the peripheral circuit event to be selected	RW

12.4.8 TimerA Hardware Trigger Event Selection Register (TMRA_TRGSELx) (x=0-1)

After selecting internal hardware trigger in TimerA, use this register to select the trigger event source. Write the trigger event number to the TRGSEL[6:0] bits in this register. When the peripheral event corresponding to the number occurs, TimerA will be triggered to act (including start/stop/clear/increment/decrement/input capture) by this event.

All TimerA units share these 2 registers. TMRA_TRGSEL0 is used to trigger operations such as start/stop/clear/increment/decrement. TMRA_TRGSEL1 is used to trigger input capture.

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	0x0000007F																																
Name	COMEN	Reserved																								TRGSEL							
Access	RW																									RW							

Bit	Name	Function	Description	Access
31	COMEN[1]	Common Trigger Enabled	0b0: Disable triggering of this AOS target by the common trigger event of AOS_COMTRGSEL2 0b1: Enable triggering of this AOS target by the common trigger event of AOS_COMTRGSEL2	RW
30	COMEN[0]	Common Trigger Enabled	0b0: Disable triggering of this AOS target by the common trigger event of AOS_COMTRGSEL1 0b1: Enable triggering of this AOS target by the common trigger event of AOS_COMTRGSEL1	RW
29:7	Reserved	Reserved	-	-
6:0	TRGSEL	Trigger Source Selection	Write the number of the peripheral circuit event to be selected	RW

12.4.9 ADC Conversion Start On-Chip Trigger Source Selection Register (ADC_TRGSELx) (x=0-1)

After selecting the on-chip trigger mode in ADC, use this register to select the trigger event source. Write the trigger event number to the TRGSEL[6:0] bits in this register. When the peripheral event corresponding to the number occurs, ADC will be triggered to start ADC conversion by this event.

Sequence A and sequence B of ADC share these 2 registers.

Offset: 0x1C+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	0x0000007F																																
Name	COMEN	Reserved																								TRGSEL							
Access	RW																									RW							

Bit	Name	Function	Description	Access
31	COMEN[1]	Common Trigger Enabled	0b0: Disable triggering of this AOS target by the common trigger event of AOS_COMTRGSEL2 0b1: Enable triggering of this AOS target by the common trigger event of AOS_COMTRGSEL2	RW
30	COMEN[0]	Common Trigger Enabled	0b0: Disable triggering of this AOS target by the common trigger event of AOS_COMTRGSEL1 0b1: Enable triggering of this AOS target by the common trigger event of AOS_COMTRGSEL1	RW
29:7	Reserved	Reserved	-	-
6:0	TRGSEL	Trigger Source Selection	Write the number of the peripheral circuit event to be selected	RW

12.4.10 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: 0x3C (AOS_COMTRGSEL1), 0x40 (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	0x0000007F																															
Name	Reserved																								COMTRG							
Access																									RW							

Bit	Name	Function	Description	Access
31:7	Reserved	Reserved	-	-
6:0	COMTRG	Trigger Source Selection	Write the number of the peripheral circuit event to be selected	RW

13 Clock Trimming Controller (CTC)

13.1 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.

13.2 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 4-bit calibration value for frequency calibration
- Error interrupt for indicating calibration failure

13.3 Functional Description

13.3.1 Function Block Diagram

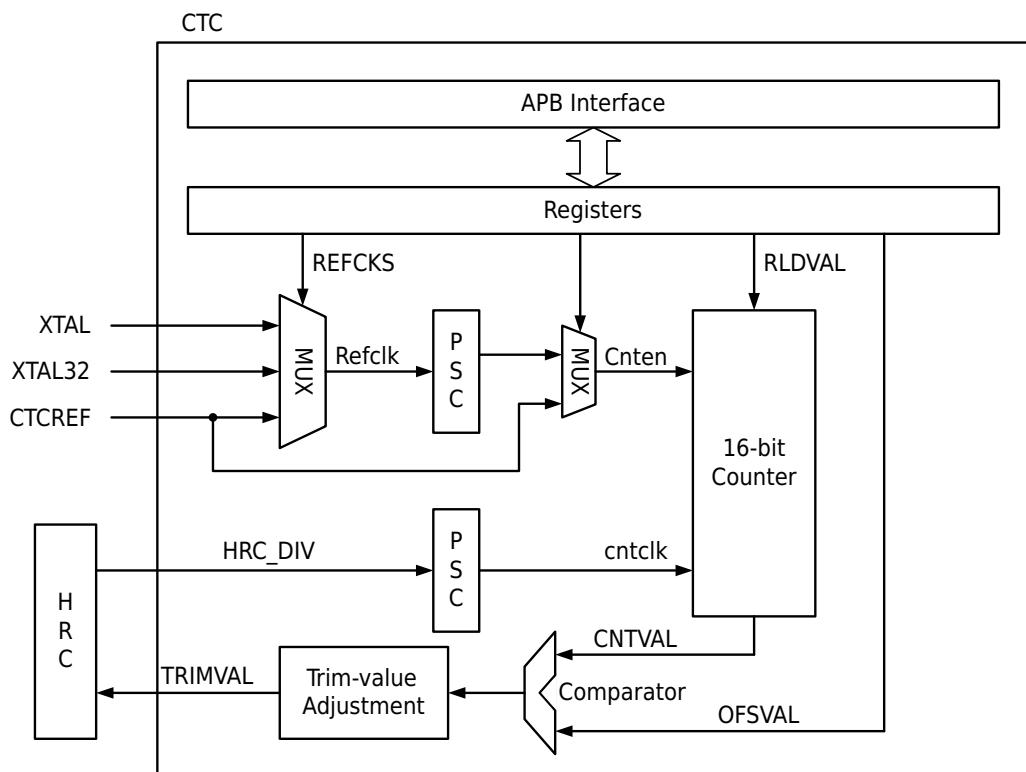


Figure 13-1 CTC Function Block Diagram

13.3.2 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, =32.768kHz)
- External reference-speed clock (CTCREF, $\geq 1\text{Hz}$)

Both the jitter of HRC and the counting error during calibration can affect the calibration accuracy. Additionally, both the CTC_CR1.REFPSC and HRC_PSC bits can be set for the reference clock and HRC to select appropriate prescalers to reduce error. The error calculation formula is as follows:

$$TE = \frac{0.099\%}{\sqrt{(F_{hrc} \div F_{ref}) \times PSC_{ref} \div 10^5}} + \frac{4}{(F_{hrc} \div F_{ref}) \times (PSC_{ref} \div PSC_{hrc})}$$

Where:

- F_{hrc} represents the target frequency of HRC, in Hz;
- F_{ref} represents the frequency of the reference clock, in Hz;
- PSC_{hrc} represents the HRC frequency division number;
- PSC_{ref} represents the reference clock frequency division number.

During single calibration with CTC, only the CTCREF input can be used as the reference signal, and the active edge of the reference signal is selected through the CTC_CR1.REFEDG bit. At this time, the setting of CTC_CR1.REFPSC has no effect, and each calibration is performed only between two active edges of the reference signal input (e.g., from rising edge to rising edge, called a calibration period). The error for single calibration is calculated using the following formula:

$$TE = \frac{0.099\%}{\sqrt{F_{hrc} \times T_{trim} \div 10^5}} + \frac{4}{F_{hrc} \times T_{trim} \div PSC_{hrc}}$$

Where:

- F_{hrc} represents the target frequency of HRC, in Hz;
- T_{trim} represents the calibration period, in s;
- PSC_{hrc} represents the HRC 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 48MHz and 64MHz, 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 13-1 Measurement Error for HRC Target Frequency of 64MHz

F _{ref}		1kHz							
PSC _{ref}		2	4	8	16	64	256	1024	4096
PSC _{hrc}	1	Cannot be set	Cannot be set	Cannot be set	Cannot be set	Cannot be set	Cannot be set	Cannot be set	Cannot be set
	2	0.094%	Cannot be set	Cannot be set	Cannot be set	Cannot be set	Cannot be set	Cannot be set	Cannot be set
	4	0.100%	0.068%	Cannot be set	Cannot be set	Cannot be set	Cannot be set	Cannot be set	Cannot be set
	8	0.113%	0.074%	0.050%	Cannot be set	Cannot be set	Cannot be set	Cannot be set	Cannot be set
	32	0.188%	0.112%	0.069%	0.043%	Cannot be set	Cannot be set	Cannot be set	Cannot be set
	128	0.488%	0.262%	0.144%	0.081%	0.028%	Cannot be set	Cannot be set	Cannot be set
	512	Not Recommended	0.862%	0.444%	0.231%	0.065%	0.020%	Cannot be set	Cannot be set
	2048	Not Recommended	Not Recommended	Not Recommended	0.831%	0.215%	0.058%	0.016%	Cannot be set
F _{ref}		32.768kHz							
PSC _{ref}		2	4	8	16	64	256	1024	4096
PSC _{hrc}	1	0.603%	0.405%	0.276%	0.190%	Cannot be set	Cannot be set	Cannot be set	Cannot be set
	2	0.706%	0.457%	0.302%	0.203%	0.095%	Cannot be set	Cannot be set	Cannot be set
	4	0.911%	0.559%	0.353%	0.228%	0.101%	Cannot be set	Cannot be set	Cannot be set
	8	Not Recommended	0.764%	0.455%	0.279%	0.114%	0.051%	Cannot be set	Cannot be set
	32	Not Recommended	Not Recommended	1.070%	0.587%	0.191%	0.070%	0.029%	Cannot be set
	128	Not Recommended	Not Recommended	Not Recommended	Not Recommended	0.498%	0.147%	0.048%	0.017%
	512	Not Recommended	Not Recommended	Not Recommended	Not Recommended	Not Recommended	0.454%	0.125%	0.037%

PSC_{hrc}	2048	Not Recommended	Not Recommended	Not Recommended	Not Recommended	Not Recommended	Not Recommended	0.432%	0.113%
F_{ref}	100kHz								
PSC_{ref}		2	4	8	16	64	256	1024	4096
PSC_{hrc}	1	1.188%	0.775%	0.516%	0.348%	0.164%	Cannot be set	Cannot be set	Cannot be set
	2	Not Recommended	0.931%	0.594%	0.388%	0.174%	Cannot be set	Cannot be set	Cannot be set
	4	Not Recommended	Not Recommended	0.750%	0.466%	0.194%	0.087%	Cannot be set	Cannot be set
	8	Not Recommended	Not Recommended	1.063%	0.622%	0.233%	0.097%	Cannot be set	Cannot be set
	32	Not Recommended	Not Recommended	Not Recommended	Not Recommended	0.467%	0.155%	0.058%	Cannot be set
	128	Not Recommended	Not Recommended	Not Recommended	Not Recommended	Not Recommended	0.390%	0.117%	0.039%
	512	Not Recommended	Not Recommended	Not Recommended	Not Recommended	Not Recommended	Not Recommended	0.351%	0.097%
	2048	Not Recommended	Not Recommended	Not Recommended	Not Recommended	Not Recommended	Not Recommended	Not Recommended	0.332%
F_{ref}	1MHz								
PSC_{ref}		2	4	8	16	64	256	1024	4096
PSC_{hrc}	1	Not Recommended	Not Recommended	Not Recommended	Not Recommended	0.587%	0.269%	Cannot be set	Cannot be set
	2	Not Recommended	Not Recommended	Not Recommended	Not Recommended	0.684%	0.293%	0.134%	Cannot be set
	4	Not Recommended	Not Recommended	Not Recommended	Not Recommended	0.880%	0.342%	0.147%	Cannot be set

PSC _{hrc}	8	Not Recommended	Not Recommended	Not Recommended	Not Recommended	Not Recommended	0.440%	0.171%	0.073%
	32	Not Recommended	Not Recommended	Not Recommended	Not Recommended	Not Recommended	1.026%	0.318%	0.110%
	128	Not Recommended	Not Recommended	Not Recommended	Not Recommended	Not Recommended	Not Recommended	0.904%	0.256%
	512	Not Recommended	Not Recommended	Not Recommended	Not Recommended	Not Recommended	Not Recommended	Not Recommended	0.842%
	2048	Not Recommended	Not Recommended	Not Recommended	Not Recommended	Not Recommended	Not Recommended	Not Recommended	Not Recommended
F _{ref}		8MHz							
PSC _{ref}		2	4	8	16	64	256	1024	4096
PSC _{hrc}	1	Not Recommended	Not Recommended	Not Recommended	Not Recommended	Not Recommended	0.887%	0.395%	0.185%
	2	Not Recommended	Not Recommended	Not Recommended	Not Recommended	Not Recommended	1.082%	0.444%	0.197%
	4	Not Recommended	Not Recommended	Not Recommended	Not Recommended	Not Recommended	Not Recommended	0.541%	0.222%
	8	Not Recommended	Not Recommended	Not Recommended	Not Recommended	Not Recommended	Not Recommended	0.737%	0.271%
	32	Not Recommended	Not Recommended	Not Recommended	Not Recommended	Not Recommended	Not Recommended	Not Recommended	0.564%
	128	Not Recommended	Not Recommended	Not Recommended	Not Recommended	Not Recommended	Not Recommended	Not Recommended	Not Recommended
	512	Not Recommended	Not Recommended	Not Recommended	Not Recommended	Not Recommended	Not Recommended	Not Recommended	Not Recommended
	2048	Not Recommended	Not Recommended	Not Recommended	Not Recommended	Not Recommended	Not Recommended	Not Recommended	Not Recommended

Table 13-2 Measurement Error for HRC Target Frequency of 48MHz

F_{ref}		1kHz							
PSC_{ref}		2	4	8	16	64	256	1024	4096
PSC_{hrc}	1	Cannot be set	Cannot be set	Cannot be set	Cannot be set	Cannot be set	Cannot be set	Cannot be set	Cannot be set
	2	0.109%	Cannot be set	Cannot be set	Cannot be set	Cannot be set	Cannot be set	Cannot be set	Cannot be set
	4	0.118%	0.080%	Cannot be set	Cannot be set	Cannot be set	Cannot be set	Cannot be set	Cannot be set
	8	0.134%	0.088%	0.059%	Cannot be set	Cannot be set	Cannot be set	Cannot be set	Cannot be set
	32	0.234%	0.138%	0.084%	0.052%	Cannot be set	Cannot be set	Cannot be set	Cannot be set
	128	0.634%	0.338%	0.184%	0.102%	0.035%	Cannot be set	Cannot be set	Cannot be set
	512	Not Recommended	Not Recommended	0.584%	0.302%	0.085%	0.026%	Cannot be set	Cannot be set
	2048	Not Recommended	Not Recommended	Not Recommended	Not Recommended	0.285%	0.076%	0.021%	Cannot be set
F_{ref}		32.768kHz							
PSC_{ref}		2	4	8	16	64	256	1024	4096
PSC_{hrc}	1	0.715%	0.477%	0.323%	0.222%	0.107%	Cannot be set	Cannot be set	Cannot be set
	2	0.851%	0.546%	0.357%	0.239%	0.111%	Cannot be set	Cannot be set	Cannot be set
	4	1.125%	0.682%	0.426%	0.273%	0.119%	0.055%	Cannot be set	Cannot be set
	8	Not Recommended	0.955%	0.562%	0.341%	0.136%	0.060%	Cannot be set	Cannot be set
	32	Not Recommended	Not Recommended	Not Recommended	0.751%	0.239%	0.085%	0.034%	Cannot be set
	128	Not Recommended	Not Recommended	Not Recommended	Not Recommended	0.648%	0.188%	0.060%	0.021%
	512	Not Recommended	Not Recommended	Not Recommended	Not Recommended	Not Recommended	0.597%	0.162%	0.047%

PSC _{hrc}	2048	Not Recommended	Not Recommended	Not Recommended	Not Recommended	Not Recommended	Not Recommended	0.572%	0.149%
F _{ref}		100kHz							
PSC _{ref}		2	4	8	16	64	256	1024	4096
PSC _{hrc}	1	Not Recommended	0.923%	0.609%	0.409%	0.192%	Cannot be set	Cannot be set	Cannot be set
	2	Not Recommended	1.131%	0.714%	0.461%	0.205%	0.096%	Cannot be set	Cannot be set
	4	Not Recommended	Not Recommended	0.922%	0.566%	0.231%	0.102%	Cannot be set	Cannot be set
	8	Not Recommended	Not Recommended	Not Recommended	0.774%	0.283%	0.115%	0.051%	Cannot be set
	32	Not Recommended	Not Recommended	Not Recommended	Not Recommended	0.595%	0.193%	0.071%	0.029%
	128	Not Recommended	Not Recommended	Not Recommended	Not Recommended	Not Recommended	0.506%	0.149%	0.048%
	512	Not Recommended	Not Recommended	Not Recommended	Not Recommended	Not Recommended	Not Recommended	0.461%	0.126%
	2048	Not Recommended	Not Recommended	Not Recommended	Not Recommended	Not Recommended	Not Recommended	Not Recommended	0.439%
F _{ref}		1MHz							
PSC _{ref}		2	4	8	16	64	256	1024	4096
PSC _{hrc}	1	Not Recommended	Not Recommended	Not Recommended	Not Recommended	0.695%	0.215%	0.149%	Cannot be set
	2	Not Recommended	Not Recommended	Not Recommended	Not Recommended	0.825%	0.348%	0.157%	Cannot be set
	4	Not Recommended	Not Recommended	Not Recommended	Not Recommended	1.086%	0.413%	0.174%	0.079%

PSC _{hrc}	8	Not Recommended	Not Recommended	Not Recommended	Not Recommended	Not Recommended	0.543%	0.206%	0.087%
	32	Not Recommended	Not Recommended	Not Recommended	Not Recommended	Not Recommended	Not Recommended	0.402%	0.136%
	128	Not Recommended	Not Recommended	Not Recommended	Not Recommended	Not Recommended	Not Recommended	Not Recommended	0.331%
	512	Not Recommended	Not Recommended	Not Recommended	Not Recommended	Not Recommended	Not Recommended	Not Recommended	Not Recommended
	2048	Not Recommended	Not Recommended	Not Recommended	Not Recommended	Not Recommended	Not Recommended	Not Recommended	Not Recommended
F _{ref}		8MHz							
PSC _{ref}		2	4	8	16	64	256	1024	4096
PSC _{hrc}	1	Not Recommended	Not Recommended	Not Recommended	Not Recommended	Not Recommended	1.059%	0.465%	0.216%
	2	Not Recommended	Not Recommended	Not Recommended	Not Recommended	Not Recommended	Not Recommended	0.530%	0.232%
	4	Not Recommended	Not Recommended	Not Recommended	Not Recommended	Not Recommended	Not Recommended	0.660%	0.265%
	8	Not Recommended	Not Recommended	Not Recommended	Not Recommended	Not Recommended	Not Recommended	0.920%	0.330%
	32	Not Recommended	Not Recommended	Not Recommended	Not Recommended	Not Recommended	Not Recommended	Not Recommended	0.721%
	128	Not Recommended	Not Recommended	Not Recommended	Not Recommended	Not Recommended	Not Recommended	Not Recommended	Not Recommended
	512	Not Recommended	Not Recommended	Not Recommended	Not Recommended	Not Recommended	Not Recommended	Not Recommended	Not Recommended
	2048	Not Recommended	Not Recommended	Not Recommended	Not Recommended	Not Recommended	Not Recommended	Not Recommended	Not Recommended

13.3.3 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.

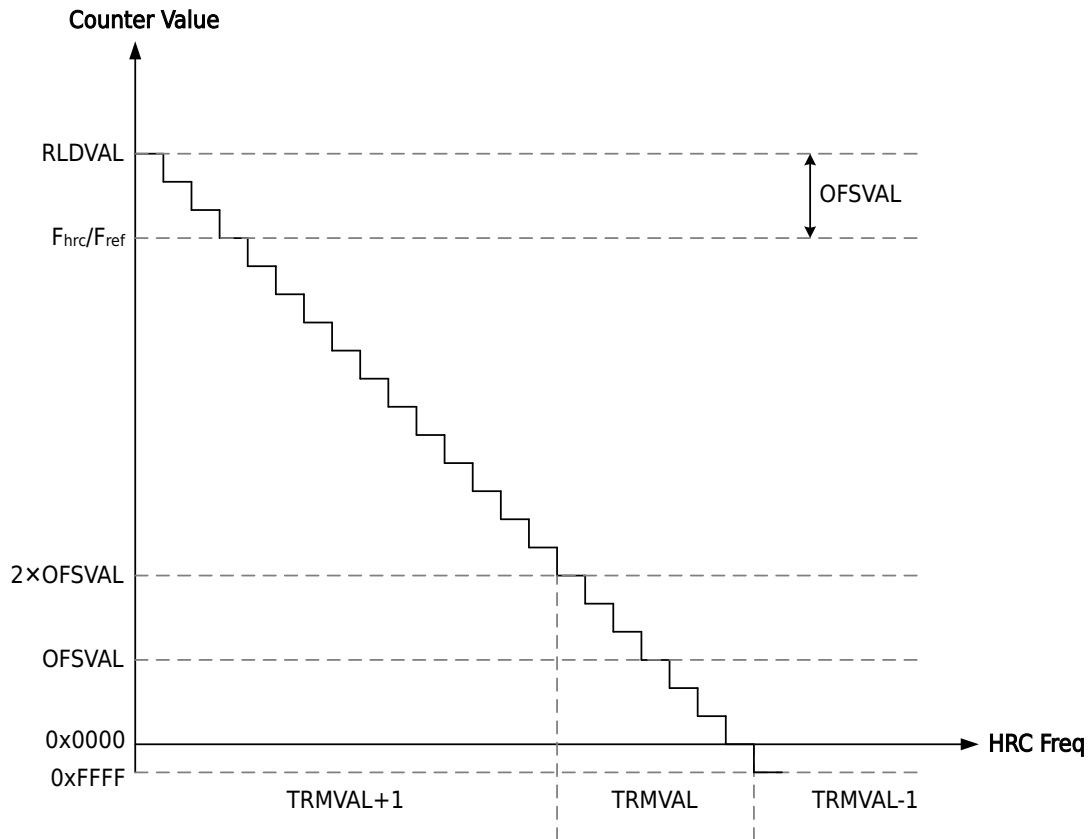


Figure 13-2 CTC Calibration Diagram

During continuous calibration, CTC decides to increment or decrement the value of TRMVAL in the CTC_CR1 register by 1 based on the relationship between the value of the 16-bit calibration counter after it stops counting and the value of $2 \times \text{OFSVAL}$. 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 $2 \times \text{OFSVAL}$ 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 > ($\text{OFSVAL} \times 2$), counting stops, and the value of TRMVAL is automatically incremented by 1, and the CTC_STR.TRIMOK bit automatically becomes 0.
- 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, and the CTC_STR.TRIMOK bit automatically becomes 0.

During continuous calibration, the next calibration starts automatically after each adjustment of TRMVAL, and stops when CTCEN is cleared by software or a calibration error occurs. During single calibration, it stops automatically after each adjustment of TRMVAL, and software is required to write 1 to CTCEN to start the next calibration.

13.3.4 Interrupt

The mid-value of TRMVAL is 0x00. If the result of automatically adding 1 exceeds 0x07, the value of TRMVAL remains 0x07, 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 0x08, the value of TRMVAL remains 0x08, 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.

13.4 Operation Example

13.4.1 Use XTAL32 as the Reference Clock to Calibrate HRC to 48MHz with a Calibration Target Deviation of Less Than 0.5%

Set XTAL32 to a 8-division prescaler and HRC to no division prescaler. According to [Table 13-1 : Measurement Error for HRC Target Frequency of 64MHz](#), the calibration error is 0.323%, which is less than the 0.5% calibration target deviation, so the scheme is feasible.

- Set register CTC_CR2=0x2E01003A, where:
 - $\text{OFSVAL} = (48 \div 0.032768) \times 8 \times 0.5\% \approx 58$ (0x3A)
 - $\text{RLDVAL} = (48 \div 0.032768) \times 8 + 58 \approx 11777$ (0x2E01)
- Set register CTC_CR1=0x000000E2, where:
 - TRMVAL=0x00 (Start calibration from the mid-value)
 - HRCPSC=0b000 (Select no frequency division of HRC)
 - CTCEN=1 (Start CTC)
 - ERRIE=1 (Enable error interrupt)
 - REFCKS=0b10 (Select XTAL32 as the reference clock for continuous calibration)
 - REFPSC=0b010 (Select 8 frequency division of reference clock)
- 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.
- Confirm CTC_STR.CTCBSY=0 before proceeding with subsequent operations.

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.

13.5 Register Description

13.5.1 Register List

Base Address: 0x40000000

Table 13-3 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
0x0C	CTC_CNT	CTC Counter Register	16

13.5.2 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	REFEDG		Reserved	HRCPSC		CTCEN	ERRIE	REFCKS		Reserved	REFPSC						
Access													RW			RW			RW		RW	RW	RW			RW						

Bit	Name	Function	Description	Access
31:20	Reserved	Reserved	-	-
19:16	TRMVAL	HRC Calibration Value	TRMVAL is a 4-bit signed calibration value with a median value of 0. ● 0x8: -8 ● 0x9: -7 ● ● 0xF: -1 ● 0x0: 0 (Median value) ● 0x1: +1 ● ● 0x6: +6 ● 0x7: +7 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. 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.	RW
15:14	Reserved	Reserved	-	-
13:12	REFEDG	Reference Signal Edge Selection	Used to select the active edge of CTCREF when REFCKS=0b01. ● 0b00: Selects rising edge to rising edge calibration for CTCREF. ● 0b01: Selects rising edge to falling edge calibration for CTCREF. ● 0b10: Selects falling edge to rising edge calibration for CTCREF. ● 0b11: Selects falling edge to falling edge calibration for CTCREF. When CTCEN is 1, modifying the value of REFEDG is prohibited. To modify, write CTCEN to 0 and confirm CTC_STR.CTCBSY is 0 before making changes.	RW
11	Reserved	Reserved	-	-

Bit	Name	Function	Description	Access
10:8	HRCPSC	HRC Prescaler Selection	● 0b000: No division ● 0b001: Divide by 2 ● 0b010: Divide by 4 ● 0b011: Divide by 8 ● 0b100: Divide by 32 ● 0b101: Divide by 128 ● 0b110: Divide by 512 ● 0b111: Divide by 2048 When CTCEN is 1, modifying the value of HRCPSC is prohibited. To modify, write CTCEN to 0 and confirm CTC_STR.CTCBSY is 0 before making changes.	RW
7	CTCEN	Calibration Enable	Used to start frequency calibration. ● 0b0: Stops frequency calibration ● 0b1: Starts frequency calibration  Note ● Setting Conditions: Set by software. ● Clearing Conditions: ▶ Software Clear ▶ Automatically clears on calibration overflow or underflow ▶ Automatically clears at the end of single calibration	RW
6	ERRIE	Error Interrupt Enable	Used to select whether an error interrupt is generated on calibration overflow or underflow. ● 0b0: Disables error interrupt ● 0b1: Enables error interrupt	RW
5:4	REFCKS	Reference Clock Selection	Used to select the reference clock source for frequency calibration. ● 0b00: Selects CTCREF as the reference clock (continuous calibration) ● 0b01: Selects CTCREF as the reference signal (single calibration) ● 0b10: Selects XTAL32 as the reference clock (continuous calibration) ● 0b11: Selects XTAL as the reference clock (continuous calibration) 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>	-	-

Bit	Name	Function	Description	Access
2:0	REFPSC	Reference Clock Prescaler Selection	Selects the prescaler for the reference clock when REFCKS=0b00/0b10/0b11. ● 0b000: Divide by 2 ● 0b001: Divide by 4 ● 0b010: Divide by 8 ● 0b011: Divide by 16 ● 0b100: Divide by 64 ● 0b101: Divide by 256 ● 0b110: Divide by 1024 ● 0b111: Divide by 4096 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

13.5.3 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	RLDVAL sets the reload value for the 16-bit calibration counter. Continuous Calibration: $RLDVAL = (F_{hrc} \div PSC_{hrc}) \div (F_{ref} \div PSC_{ref}) + OFSVAL$ Single Calibration: $RLDVAL = ((F_{hrc} \div PSC_{hrc}) \times T_{trim}) + OFSVAL$ ● F_{hrc}: HRC target frequency; ● F_{ref}: Reference clock frequency; ● T_{trim}: Single calibration period; ● OFSVAL: Calibration count deviation value; ● PSC_{hrc}: HRC prescaler; ● PSC_{ref}: Reference clock prescaler.  Note 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.	RW
15:8	Reserved	Reserved	-	-

Bit	Name	Function	Description	Access
7:0	OFSVAL	Calibration Count Deviation Value	OFSVAL sets the allowed deviation value for HRC during calibration. Continuous Calibration: $OFSVAL = (F_{hrc} \div PSC_{hrc}) \div (F_{ref} \div PSC_{ref}) \times TA$ Single Calibration: $OFSVAL = ((F_{hrc} \div PSC_{hrc}) \times T_{trim}) \times TA$ ● F_{hrc}: HRC target frequency; ● F_{ref}: Reference clock frequency; ● T_{trim}: Single calibration period; ● TA: Calibration accuracy, e.g., 0.5%; ● PSC_{hrc}: HRC prescaler; ● PSC_{ref}: Reference clock prescaler.  Note 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.	RW

13.5.4 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	0b0: CTC is stopped 0b1: CTC is operating	RO
2	TRMUDF	Calibration Underflow Flag	Calibration underflow indicates that the TRMVAL has been decremented to 0x08, 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. 0b0: No calibration underflow. 0b1: Calibration underflow occurred. 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 0x07, 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. 0b0: No calibration overflow. 0b1: Calibration overflow occurred. Cleared when CTC_CR1.CTCEN is written to 1.	RO
0	TRIMOK	Calibration Success Flag	0b0: HRC frequency has not been calibrated to the target range. 0b1: HRC frequency has been calibrated to the target range. Cleared when CTC_CR1.CTCEN is written to 1.	RO

13.5.5 CTC Counter Register (CTC_CNT)

Offset	Bit Position															
0x0C	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0000															
Name	CTCCNT															
Access	RO															

Bit	Name	Function	Description	Access
15:0	CTCCNT	Calibration Counter	The calibration counter reloads the value of RLDVAL at the start of each calibration and counts down from there. The counter value is retained until the next RLDVAL reload after calibration ends. Read after writing CTCEN to 0 and confirming CTC_STR.CTCBSY is 0.	RO

14 DMA Controller (DMA)

14.1 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.

14.2 Features

- The DMA bus is independent of the CPU bus and transmits data according to the AMBA AHB-Lite bus protocol
- With 4 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 256 data
- The width of each data item can be configured as 8-bit, 16-bit, or 32-bit
- It can be configured for 1-1023 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
- When not in use, it can be set to stop state to reduce power consumption

14.3 Functional Description

14.3.1 DMA Block Diagram

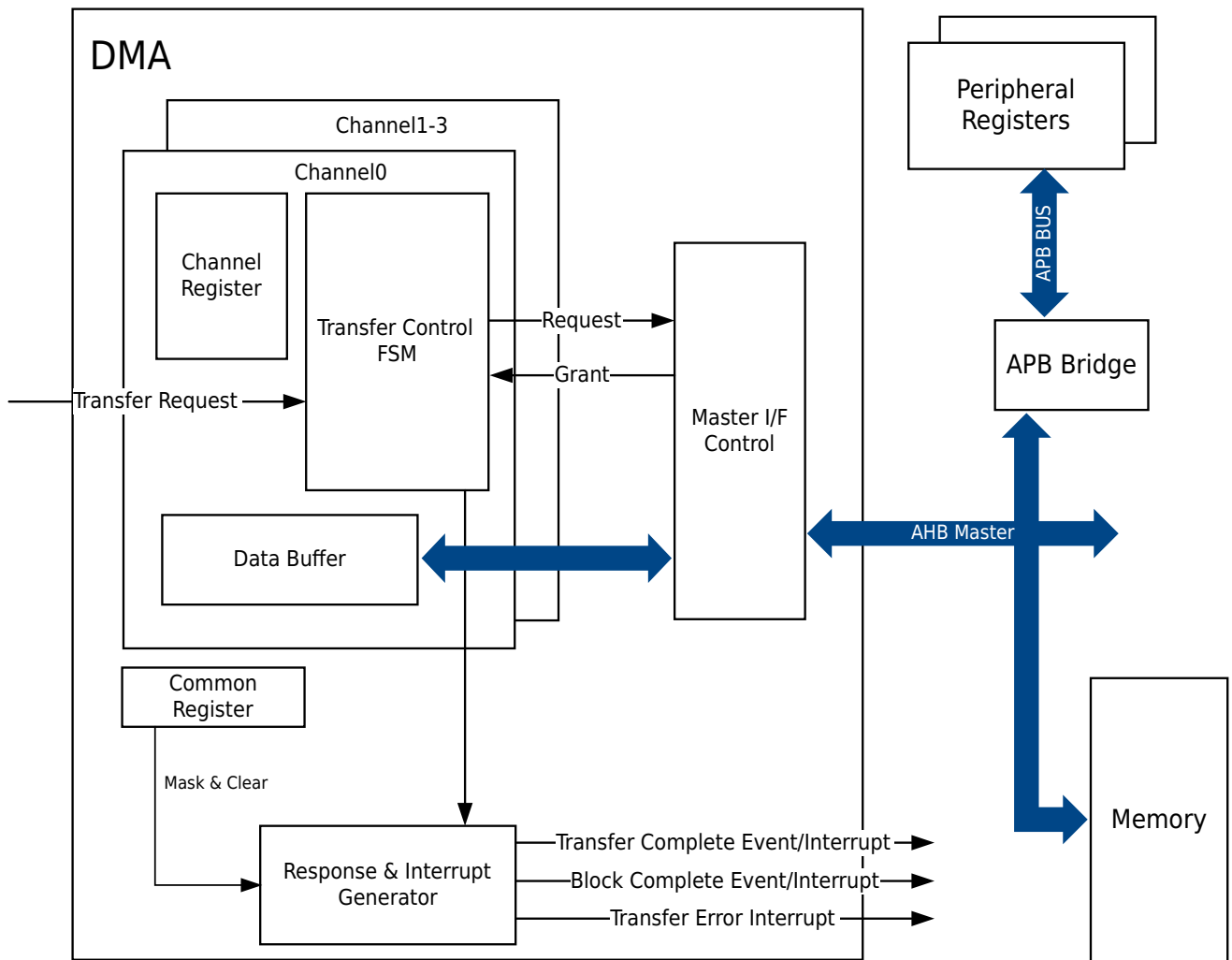


Figure 14-1 DMA Architecture Diagram

14.3.2 Enabling the DMA Controller

When using DMA, you need to write the DMA enable register controller first. The enabling method is to write the DMA enable register DMA_EN.EN bit.

When DMA is not in use or the chip needs to enter STOP mode, please set DMA_EN.EN to 0. Before setting EN to 0, please confirm that DMA_CHSTAT.DMAACT is 0 to ensure that DMA has completed all transfers.

14.3.3 Channel Selection and Channel Priority

The DMA control unit contains 4 channels, and each channel can independently configure transfer functions.

The channel priority order is: Channel 0 > Channel 1 > ... > Channel 3.

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.

14.3.4 Start DMA

DMA is started by requests generated by peripheral circuits, which are configured through the trigger source selection register (DMA_TRGSELx, where x=0-3) for each channel independently. 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 CMU_FCG register to active.

14.3.5 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_CHxCTL0.BLKSIZE, and the maximum size that can be set is 256 data. The data width of each is determined by DMA_CHxCTL0.HSIZE and can be 8-bit, 16-bit, or 32-bit.

14.3.6 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. Among them, repeated jump and NonSequence jump can only be selected one at a time.

- **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 DMA_CHxCTL0.HSIZE and the setting value of the DMA_CHxCTL1.RPTNSCNT register.
- **NonSequence Jump:** After transferring a specified number of data items, the source and destination addresses skip a specified offset.

14.3.7 Transfer Count

The total number of data blocks transferred by DMA is set by the DMA_CHxCTL0.CNT bit. The maximum number of transmissions can be set to 1023. 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_CHxCTL0.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.

14.3.8 Chain Transfer

DMA controller has the function of chain transfer. The chain transfer needs to configure the following 4 registers, a total of 4 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 transfer. The register DMA_CHxCTL1.LLPSEL can select the descriptor to be stored at chain descriptor base address 1 or chain descriptor base address 2.

- DMA_SARx
- DMA_DARx
- DMA_CHxCTL0
- DMA_CHxCTL1

Where DMA_CHxCTL0.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_CHxCTL0.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_CHxCTL0.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_CHxCTL0.LLPRUN settings, starting the first transfer directly after loading the new descriptor.

- When DMA_CHxCTL0.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_CHxCTL0.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.

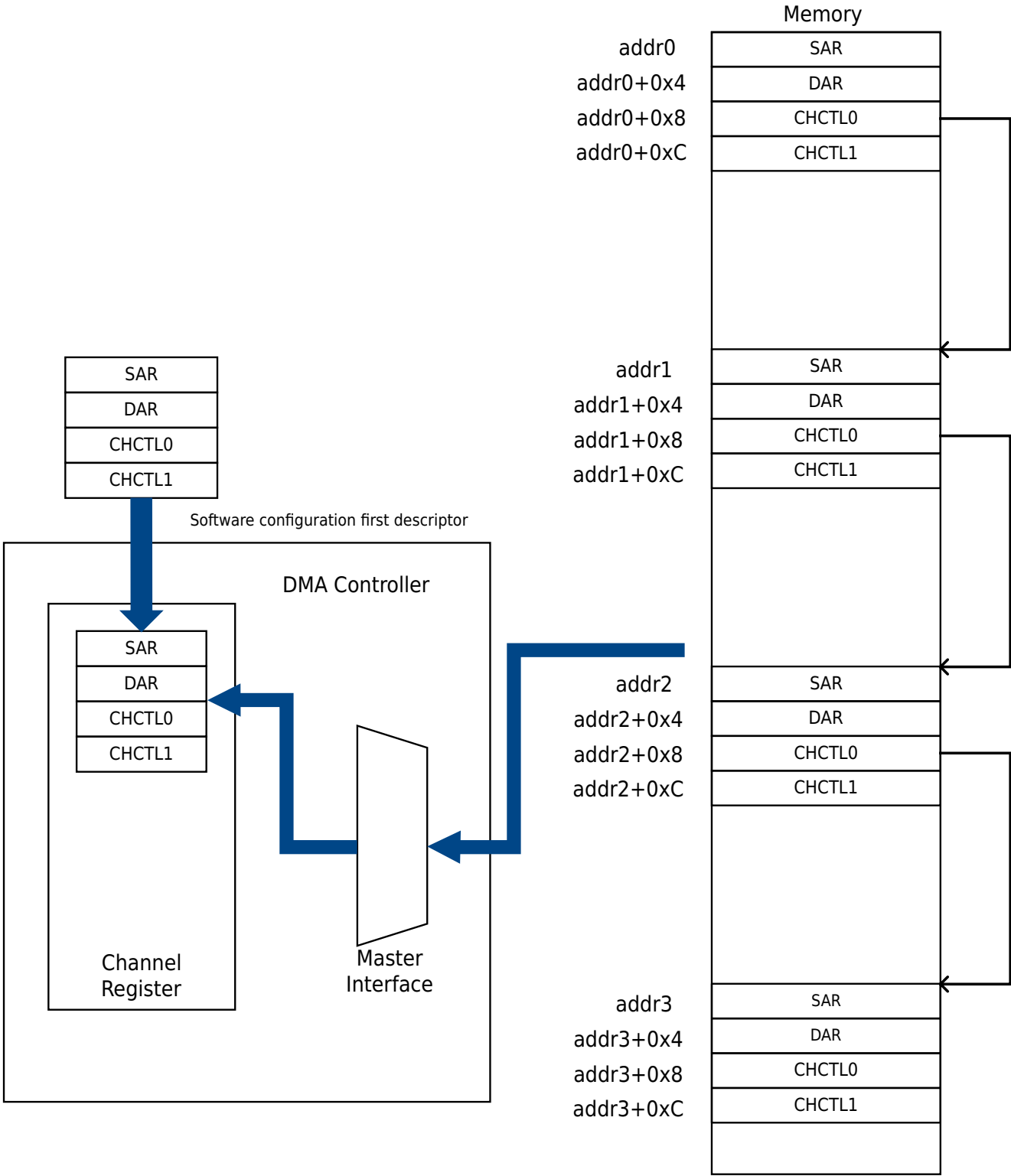


Figure 14-2 Chain Transmission Diagram

14.3.9 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. The direction of the jump follows DMA_CHxCTL1.SINC and DMA_CHxCTL1.DINC settings jump forward or backward. To use it, first set DMA_CHxCTL1.RPTNSEN to 1 and configure DMA_CHxCTL1.RPTNSSEL to enable Non-Sequence address transfer. Then configure the channel control registers DMA_CHxCTL1.OFFSET and DMA_CHxCTL1.RPTNSCNT. The transfer process proceeds as shown in the following diagram.

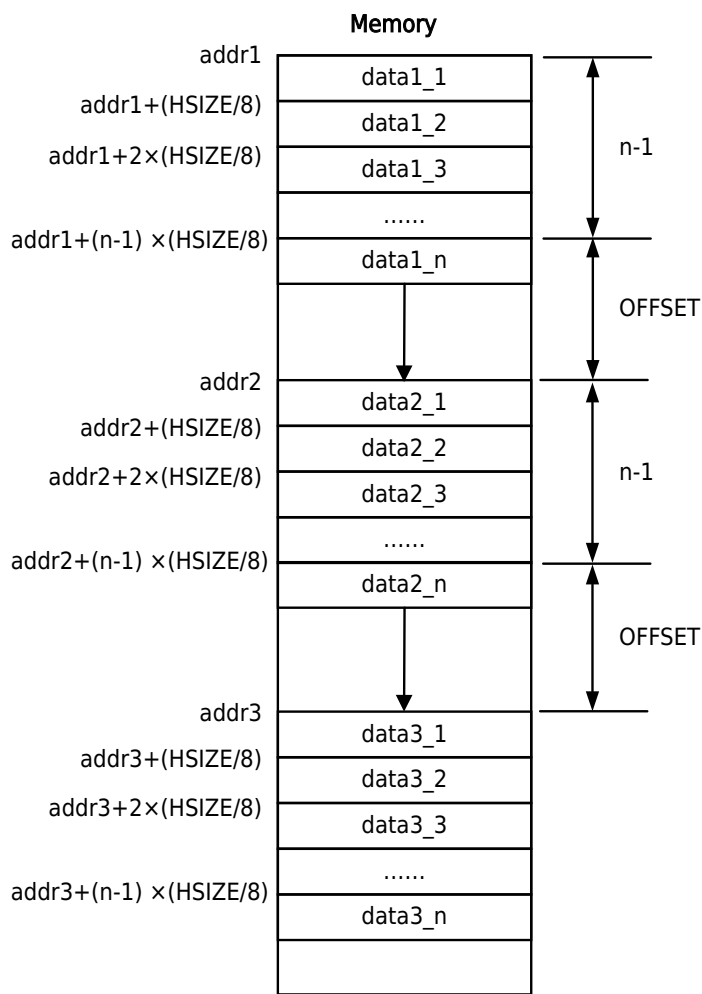


Figure 14-3 Schematic Diagram of NonSequence Address Transfer (Taking source address as an example)

$n = \text{RPTNSCNT}$

$\text{addr2} = \text{addr1} + (n-1) \times (\text{HSIZE}/8) + \text{OFFSET} \times (\text{HSIZE}/8)$

$\text{addr3} = \text{addr2} + (n-1) \times (\text{HSIZE}/8) + \text{OFFSET} \times (\text{HSIZE}/8)$

Where HSIZE is the data width (bit), which can be 8 bits (byte), 16 bits (half-word), or 32 bits (word).

14.3.10 Early Termination of Transfer

During transfer, the DMA_CHEN.CHENx bit remains active. For non-chain transfers, the transfer count set in register DMA_CHxCTL0 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_x, 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_x 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.

14.3.11 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 Complete Interrupt DMA_TCx**: generated after completing the transfer count set by DMA_CHxCTL0.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.

Each interrupt can be configured to be masked or unmasked by configuring the corresponding MASK register.

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.



Note

Events are not controlled by the MASK register.

14.4 Operation Example

14.4.1 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_CH0CTL0 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_CH0CTL1.RPTNSCNT 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_CH0CTL0 and DMA_CH0CTL1 to achieve:
 - 1) The source address and destination address chained transfer is disabled.
 - 2) The source address repeat enabled, destination address increment.
 - 3) The data width is 32-bit.
 - 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_CH0CTL0.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 initial address 0x20000000 based on the current transfer address and repeat transmission counter DMA_CH0CTL1.RPTNSCNT, and continue to transfer the remaining 2 addresses. After the second transfer is completed, the number of transfers DMA_CH0CTL0.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_CH0CTL0.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.

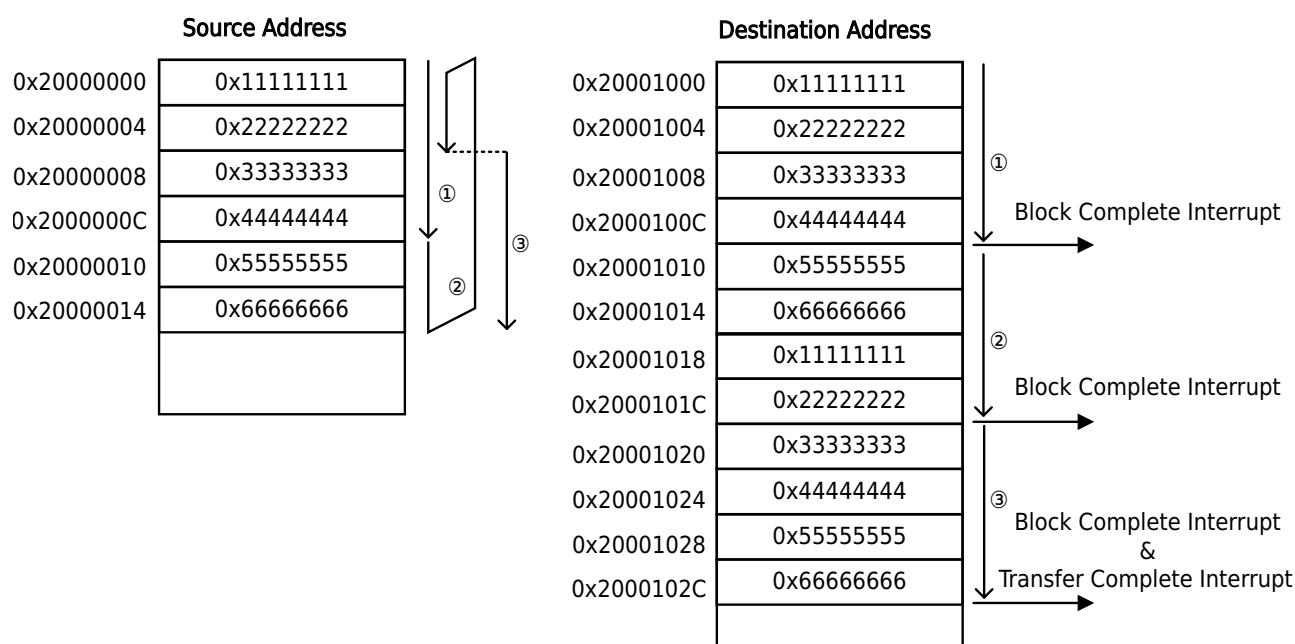


Figure 14-4 Application Example 1: Memory-to-Memory Transfer

14.4.2 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 0x4001D000.
 - c. Write DMA_CH0CTL0 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_CH0CTL0 and DMA_CH0CTL1 to achieve:
 - 1) The source address and destination address chained transfer is disabled.
 - 2) The source address increment, destination address fixed.
 - 3) The source address and destination address data width is 16-bit.

- 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.

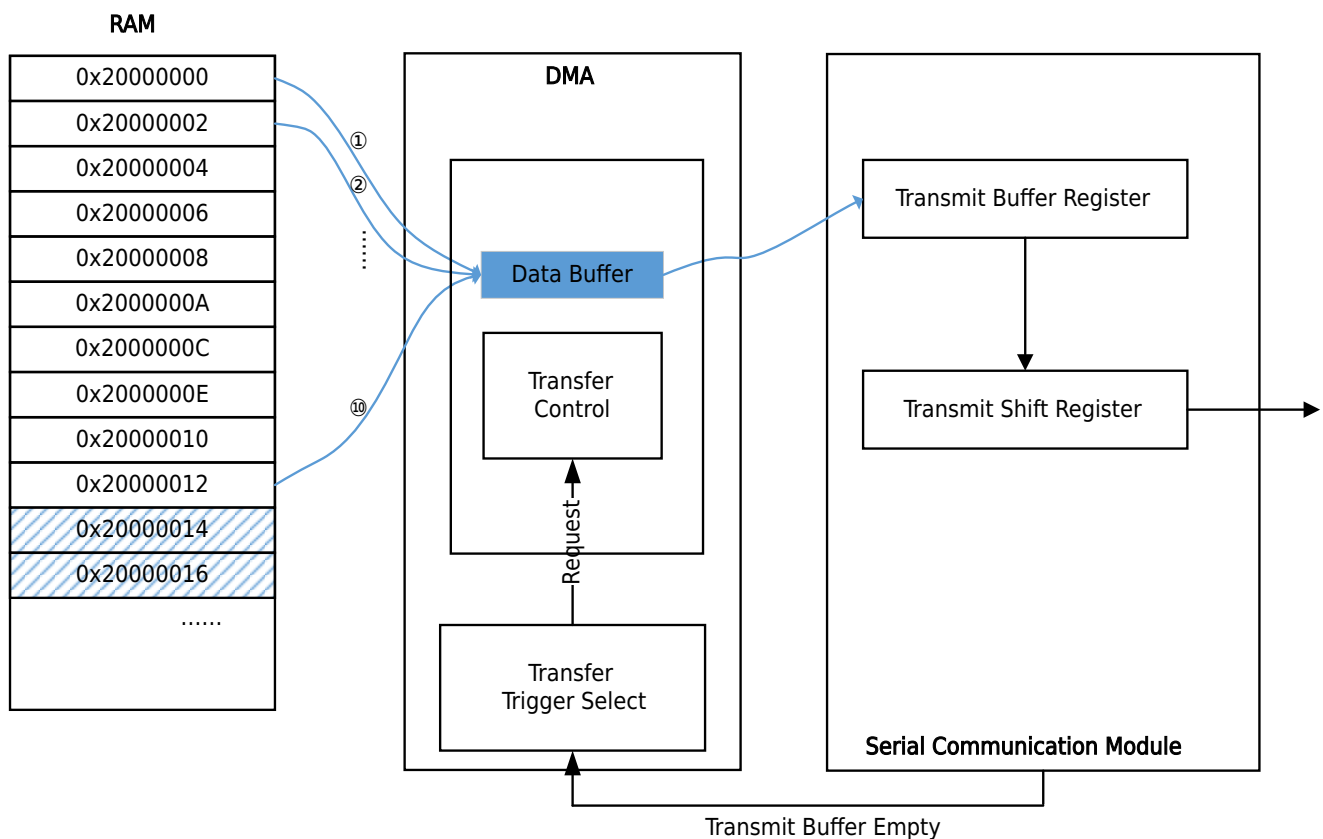


Figure 14-5 Application Example 2: Memory-to-Peripheral Transfer

14.4.3 Memory-to-Memory Chain Transfer

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 0x20000100.
 - c. Write DMA_CH0CTL0.BLKSIZE to configure the block size as 10.
 - d. Write the address of the second descriptor (descriptor1) 0x100 to DMA_CH0CTL0.LLP[13:4].
 - e. Configure the channel control registers DMA_CH0CTL0 and DMA_CH0CTL1 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.
3. Configure the second transfer descriptor (descriptor1) at address 0x20001000 in RAM as follows:
 - a. Write 32-bit data 0x20000200 to 0x20001000, which is the source address for the second transfer.
 - b. Write 32-bit data 0x20000300 to 0x20001004, which is the destination address for the second transfer.
 - c. Write to 0x20001008 to configure the block size as 20.
 - d. Set 0x110 to 0x20001018, which is the address of the third transfer descriptor (descriptor2).
 - e. Write the control data for the second transfer to 0x2000101C, 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.
4. Configure the third transfer descriptor (descriptor2) at address 0x20001100 in RAM as follows:
 - a. Write the 32-bit data 0x20000400 to 0x20001100; this sets the source address for the third transfer.
 - b. Write the 32-bit data 0x20000500 to 0x20001104; this sets the destination address for the third transfer.
 - c. Write to 0x20001108 to configure the block size as 40.
 - d. Set the LLPEN to 0 in 0x20001118, indicating the last transfer in the chain.
 - e. Write the control data for the third transfer to 0x2000111C, which configures:
 - 1) Chain transfer disabled.
 - 2) Source address and destination address increment.
 - 3) The data width is 8-bit.
5. Configure the DMA_TRGSEL0 register 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 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.

14.5 Register Description

14.5.1 Register List

Base address: 0x40013000

Table 14-1 DMA Controller (DMA) Register Offset

Offset	Register	Description	Width
0x00	DMA_EN	DMA Enable Register	32
0x04	DMA_INTSTAT0	Interrupt Status Register 0	32
0x08	DMA_INTSTAT1	Interrupt Status Register 1	32
0x0C	DMA_INTMASK0	Interrupt Mask Register 0	32
0x10	DMA_INTMASK1	Interrupt Mask Register 1	32
0x14	DMA_INTCLR0	Interrupt Status Clear Register 0	32
0x18	DMA_INTCLR1	Interrupt Status Clear Register 1	32
0x1C	DMA_CHEN	Channel Enable Register	32
0x24	DMA_CHSTAT	Channel Status Register	32
0x30	DMA_BASE1_LL	Chain Descriptor Base Address Register 1	32
0x34	DMA_BASE2_LL	Chain Descriptor Base Address Register 2	32
0x38	DMA_CHENCLR	Channel Enable Reset Register	32
0x40+0x40*x	DMA_SARx	Transmission Source Address Register	32
0x44+0x40*x	DMA_DARx	Transmission Destination Address Register	32
0x48+0x40*x	DMA_CHxCTL0	Transfer Data Control Register	32
0x4C+0x40*x	DMA_CHxCTL1	Channel Control Register	32



Note

The x in the table represents the existence of multiple registers, with values 0-3.

14.5.2 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	0b0: Disable 0b1: Enable	RW

14.5.3 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
31:20	Reserved	Reserved	-	-
19:16	REQERR	Transfer Request Overflow Error Interrupt Bit	Corresponding to REQERR3-REQERR0: 0b0: No transfer request overflow error for this channel 0b1: Transfer request overflow error for this channel, meaning a new transfer request arrived while the previous request is still pending  Note Each bit corresponds to a transmission channel.	RO
15:4	Reserved	Reserved	-	-
3:0	TRNERR	Transfer Error Interrupt Bit	Corresponding to TRNERR3-TRNERR0: 0b0: No transfer error for this channel 0b1: Transfer error for this channel  Note Each bit corresponds to a transmission channel.	RO

14.5.4 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
31:20	Reserved	Reserved	-	-
19:16	BTC	Block Transfer Complete Interrupt Bit	For BTC3-BTC0, the interrupt occurs after the completion of one block transfer: ● 0b0: No block transfer interrupt for this channel. ● 0b1: Block transfer interrupt for this channel.  Note Each bit corresponds to a transmission channel.	RO
15:4	Reserved	Reserved	-	-
3:0	TC	Transfer Complete Interrupt Bit	Corresponding to TC3-TC0, this interrupt occurs after completing the number of transmissions set by DMA_CHxCTL0.CNT: ● 0b0: No transfer completed interrupt for this channel. ● 0b1: Transfer completed interrupt for this channel.  Note Each bit corresponds to a transmission channel.	RO

14.5.5 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
31:20	Reserved	Reserved	-	-
19:16	MSKREQERR	Transmission Request Overflow Interrupt Mask	Corresponding to MSKREQERR3-MSKREQERR0: 0b0: Do not mask the transfer request overflow interrupt. 0b1: Mask the transfer request overflow interrupt.  Note Each bit corresponds to a transmission channel.	RW
15:4	Reserved	Reserved	-	-
3:0	MSKTRNERR	Transfer Error Interrupt Mask	Corresponding to MSKTRNERR3-MSKTRNERR0: 0b0: Do not mask the transfer error interrupt. 0b1: Mask the transfer error interrupt.  Note Each bit corresponds to a transmission channel.	RW

14.5.6 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
31:20	Reserved	Reserved	-	-
19:16	MSKBTC	Block Transfer Complete Interrupt Mask	Corresponding to MSKBTC3-MSKBTC0: 0b0: Do not mask the block transfer completed interrupt. 0b1: Mask the block transfer completed interrupt.  Note Each bit corresponds to a transmission channel.	RW
15:4	Reserved	Reserved	-	-
3:0	MSKTC	Transfer Completed Interrupt Mask	Corresponding to MSKTC3-MSKTC0: 0b0: Do not mask the transfer completed interrupt. 0b1: Mask the transfer completed interrupt.  Note Each bit corresponds to a transmission channel.	RW

14.5.7 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
31:20	Reserved	Reserved	-	-
19:16	CLRREQERR	Transfer Request Overflow Error Interrupt Reset	Corresponding to CLRREQERR3-CLRREQERR0: Write 1 to reset the transfer request overflow error interrupt status bit, write 0 has no effect; read always returns 0.  Note Each bit corresponds to a transmission channel.	WO
15:4	Reserved	Reserved	-	-
3:0	CLRTRNERR	Transfer Error Interrupt Reset Bit	Corresponding to CLRTRNERR3-CLRTRNERR0: Write 1 to reset the transfer error interrupt status bit, write 0 has no effect; read always returns 0.  Note Each bit corresponds to a transmission channel.	WO

14.5.8 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 RTC			
Access													WO																WO			

Bit	Name	Function	Description	Access
31:20	Reserved	Reserved	-	-
19:16	CLRBTC	Block Transfer Complete Interrupt Reset	Corresponding to CLRBTC3-CLRBTC0: Write 1 to reset the block transfer completed interrupt status bit, write 0 has no effect; read always returns 0.  Note Each bit corresponds to a transmission channel.	WO
15:4	Reserved	Reserved	-	-
3:0	CLRRTC	Transfer Completed Interrupt Reset	Corresponding to CLRRTC3-CLRRTC0: Write 1 to reset the transfer completed interrupt status bit, write 0 has no effect; read always returns 0.  Note Each bit corresponds to a transmission channel.	WO

14.5.9 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
31:4	Reserved	Reserved	-	-
3:0	CHEN	Channel Enable Bit	Corresponding to CHEN3-CHEN0: 0b0: The channel is disabled 0b1: The channel is enabled Write 1 to CHEN[x] sets CHEN[x] to 1, write 0 has no effect. 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_CHxCTL0.CNT is completed. If DMA_CHxCTL0.CNT is set to 0, it will not be automatically cleared after the transfer is completed, which means unlimited transfers.  Note - Each bit corresponds to a transmission channel. - 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.	RW

14.5.10 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																DMAACT
Access													RO																				RO

Bit	Name	Function	Description	Access
31:20	Reserved	Reserved	-	-
19:16	CHACT	Transfer Action Channel Monitor Bit	Corresponding to CHACT3-CHACT0: 0b0: The channel is idle 0b1: The channel is in action  Note Each bit corresponds to a transmission channel.	RO
15:1	Reserved	Reserved	-	-
0	DMAACT	DMA Action Monitor Bit	0b0: DMA is not in transfer action 0b1: DMA is in transfer action	RO

14.5.11 Chain Descriptor Base Address Register 1 (DMA_BASE1_LLDP)

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	0x20000000																															
Name	LLP[31:14]																		Reserved													
Access	RW																															

Bit	Name	Function	Description	Access
31:14	LLP[31:14]	Base Address 1 for Chain Descriptor	High 28 bits of descriptor address	RW
13:0	Reserved	Reserved	-	-

14.5.12 Chain Descriptor Base Address Register 2 (DMA_BASE2_LLDP)

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	LLP[31:14]																		Reserved													
Access	RW																															

Bit	Name	Function	Description	Access
31:14	LLP[31:14]	Base Address 2 for Chain Descriptor	High 28 bits of descriptor address	RW
13:0	Reserved	Reserved	-	-

14.5.13 Channel Enable Reset Register (DMA_CHENCLR)

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																												CHENCLR			
Access																													WO			

Bit	Name	Function	Description	Access
31:4	Reserved	Reserved	-	-
3:0	CHENCLR	Channel Enable Bit Reset	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.  Note Each bit corresponds to a transmission channel.	WO

14.5.14 Transmission Source Address Register (DMA_SARx) (x=0-3)

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	Set transfer source address  Note - When the transfer data width is 16-bit, i.e. DMA_CHxCTL0.HSIZE=0b01, SAR[0] is disabled. - When the transfer data width is 32-bit, i.e. DMA_CHxCTL0.HSIZE=0b10/0b11, SAR[1:0] is disabled.	RW

14.5.15 Transmission Destination Address Register (DMA_DARx) (x=0-3)

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	Set transmission destination address  Note - When the transfer data width is 16-bit, i.e. DMA_CHxCTL0.HSIZE=0b01, DAR[0] is disabled. - When the transfer data width is 32-bit, i.e. DMA_CHxCTL0.HSIZE=0b10/0b11, DAR[1:0] is disabled.	RW

14.5.16 Transfer Data Control Register (DMA_CHxCTL0) (x=0-3)

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	HSIZE				LLPRUN				LLPEN				LLP[13:4]								CNT								BLKSIZE									
Access	RW				RW				RW				RW								RW								RW									

Bit	Name	Function	Description	Access
31:30	HSIZE	Transfer Data Width	0b00: 8-bit 0b01: 16-bit 0b10/0b11: 32-bit	RW
29	LLPRUN	Chain Transfer Mode Selection	When chain transfer is enabled, set whether to immediately start a new descriptor's corresponding transfer after the current transfer completes and a new descriptor specified by the linked list pointer is loaded. 0b0: Do not initiate transfer immediately, wait for the next transfer request 0b1: Initiate transfer immediately after the new descriptor is loaded	RW
28	LLPEN	Chain Transfer Enable	0b0: Chain transfer is disabled 0b1: Chain transfer is enabled	RW
27:18	LLP[13:4]	Linked List Pointer	When chain transfer is enabled, set the address of the next transfer's descriptor. Descriptor address LLP[31:14] is specified by the Chain Descriptor Base Address Register; LLP[13:4] is specified by this register's bits; LLP[3:0] is fixed at 0.	RW
17:8	CNT	Transfer Count	Total Transfer Count, each request initiates a block transfer, and the CNT value decrements by 1 upon completion. A transfer completed interrupt occurs when the register value decrements to 0. If the register value is 0 at the start of transfer, one block is transferred per start request, CNT remains 0, and no transfer completed interrupt occurs upon completion.	RW
7:0	BLKSIZE	Data Block Size	Set the size of the data block, up to 256 pieces of data can be configured. The width of each data item is determined by the HSIZE bit. Configure the register value before transfer start. If the register value is 1, transfer 1 data item each time; if 0, transfer 256 data items each time.	RW

14.5.17 Channel Control Register (DMA_CHxCTL1) (x=0-3)

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	DINC		SINC		LLPSEL	RPTNSSEL		RPTNSEN	RPTNSCNT								OFFSET																
Access	RW		RW		RW	RW		RW	RW								RW																

Bit	Name	Function	Description	Access
31:30	DINC	Destination Address Update Method	0b00: Fixed 0b01: Incrementing 0b10/0b11: Decrementing	RW
29:28	SINC	Source Address Update Method	0b00: Fixed 0b01: Incrementing 0b10/0b11: Decrementing	RW
27	LLPSEL	Chain Descriptor Storage Address	0b0: Chain descriptor base address 1 0b1: Chain descriptor base address 2	RW
26:25	RPTNSSEL	Source and Destination Address Repeated or NonSequence Function Selection	0b00: Source address reload 0b01: Destination address reload 0b10: Source address NonSequence 0b11: Destination address NonSequence	RW
24	RPTNSEN	Repeat or NonSequence Address Transfer Function Enable Bit	Set whether to allow address reload to initial value or NonSequence. 0b0: Not allowed 0b1: Allow	RW
23:16	RPTNSCNT	Data Amount for Address Reload or Jump	For NonSequence address transfer, after transferring the data amount set by CNT, the source or destination address jumps according to the offset set by OFFSET. When address reload function is enabled, CNT represents the source or destination address repeat region size.	RW

Bit	Name	Function	Description	Access
15:0	OFFSET	Address Offset for Address Jump	For NonSequence address transfer, the source (or destination) address offset for jump. The jump direction is forward or backward based on the value of SINC (or DINC). NonSequence address transfer is disabled when SINC (or DINC) is set to fixed address. The address offset will be calculated based on the data width set by DMA_CHxCTL0.HSIZE and the value of OFFSET: $\text{Address Offset} = \text{OFFSET} \times (\text{HSIZE}(\text{bit}) / 8)$ For example, when OFFSET is set to 10 and HSIZE is 32-bit, the address offset is $10 \times 4 = 40$; for 16-bit, it's $10 \times 2 = 20$; for 8-bit, it's $10 \times 1 = 10$. The next transmission's source (or destination) address = current source (or destination) address $\pm$ address offset.	RW

15 Voltage Comparator (CMP)

15.1 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 1 comparison channel CMP.

The working clock of CMP, CMPCLK, is HCLK, i.e., synchronous action.

15.2 Features

- 1 comparison channels can independently perform normal comparison
- Each comparison channel's positive and negative voltage inputs have multiple input sources to choose from
- An 8-bit Digital-to-Analog Converter (8-bit DAC) provides one independent internal comparison voltage DA1
- A noise filter can filter the comparator output, with 7 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 VCOUNT
- Compare results can be used for EMB control events

15.3 Functional Description

15.3.1 Function Block Diagram

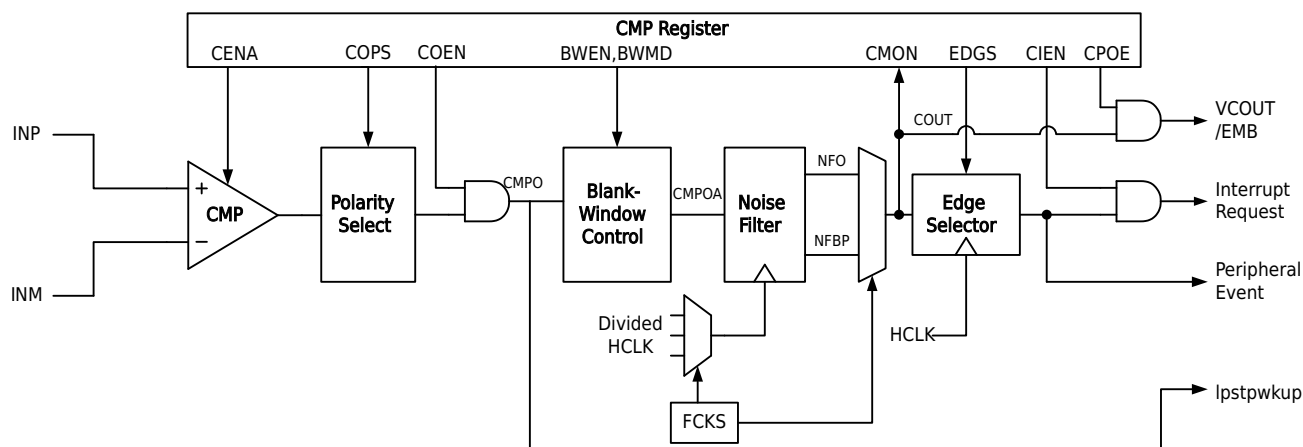


Figure 15-1 CMP Function Block Diagram

Table 15-1 CMP Pin List

Pin Name	Input/Output	Function
AVCC	Input	Analog Power
AVSS	Input	Analog Ground
CMP_IN0-7	Input	CMP Analog Input
VCOUT	Output	CMP External Port Output

15.3.2 Input Selection

The analog inputs of CMP are selected according to the following table.

Table 15-2 CMP Analog Input List

CMP				
SN	Register PMSR.CVSL	Positive input (INP)	Register PMSR.RVSL	Negative input (INM)
1	0b000000001	CMP_IN0	0b000000001	CMP_IN0
2	0b000000010	CMP_IN1	0b000000010	CMP_IN1
3	0b000000100	CMP_IN2	0b000000100	CMP_IN2
4	0b000001000	CMP_IN3	0b000001000	CMP_IN3
5	0b000010000	CMP_IN4	0b000010000	CMP_IN4
6	0b000100000	CMP_IN5	0b000100000	CMP_IN5
7	0b001000000	CMP_IN6	0b001000000	CMP_IN6
8	0b010000000	CMP_IN7	0b010000000	CMP_IN7
9	0b100000000	DA1	0b100000000	DA1

15.3.3 Normal Compare

CMP can independently perform normal comparison. the setting steps are as follows:

- Step1.** Enable the CMP module function (write 0 to PWC_FCG.Bit2);
- Step2.** Set CMP_MDR.CENA to 1 and start the comparator;
- Step3.** Set CMP_PMSR.CVSL to select the positive terminal voltage;
Set CMP_PMSR.RVSL to select the negative terminal voltage;
- Step4.** Wait for the comparator stabilization time t_{START} (for specific values, see the "Electrical Specifications" chapter in the "Datasheet");
- Step5.** Set CMP_FIR.FCKS and configure the sampling period for digital filtering;
Set CMP_FIR.EDGS and CMP_FIR.CIEN and configure the edge where the interrupt occurs;
- Step6.** Set CMP_OCR.CPOE and CMP_OCR.COPS and configure the VCOOUT output;
- Step7.** Set CMP_OCR.COEN to 1 to enable the comparator output.



Note

When setting CMP_MDR.CENA to 0 to stop the comparator, first set CMP_PMSR to 0x00000000 to ensure no voltage is input to the positive and negative terminals.

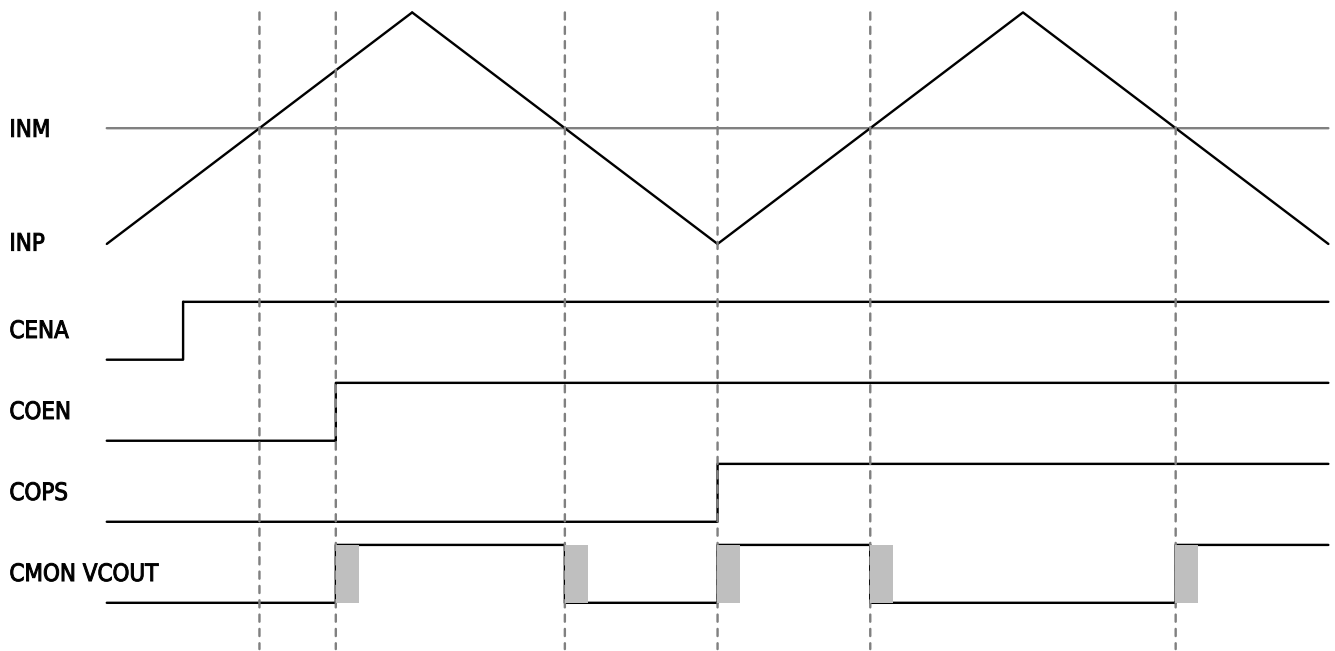


Figure 15-2 Voltage Comparison Diagram

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 CMP_OMR.CMON bit 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. If a digital filter is used, due to synchronization and filter sampling, the edge of CMON changes will be at least 6 CMPCLK periods late.

15.3.4 Window Output

The CMP can use the outputs of Timer4, 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. There are two modes for window output: level mode and edge mode.

1. Level Mode

When windowed output is enabled (set the CMP_OCR.BWEN bit to 1) and level mode is selected (set the CMP_OCR.BWMD bit to 0), select one or more timer PWMs as Blank-Windows through the CMP_BWSR1.CTWS bit, and select the active level for each Blank-Window through the CMP_BWSR1.CTWP bit. In this way, when a Blank-Window is active, the windowed output CMPOA is fixed to the state set by the CMP_OCR.BWOL bit.

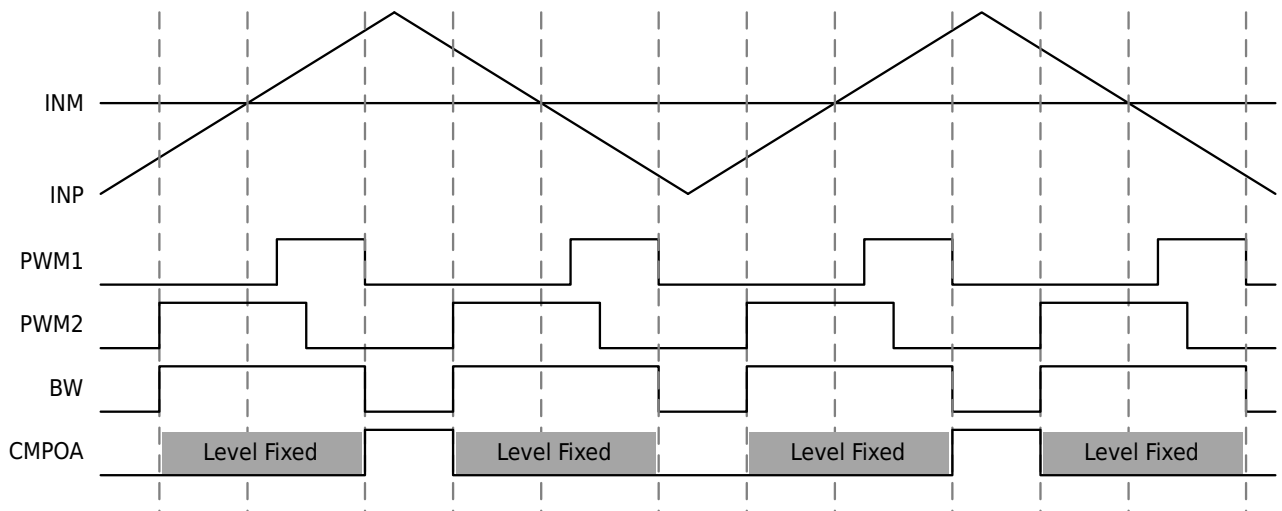


Figure 15-3 Window Output Diagram I

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.

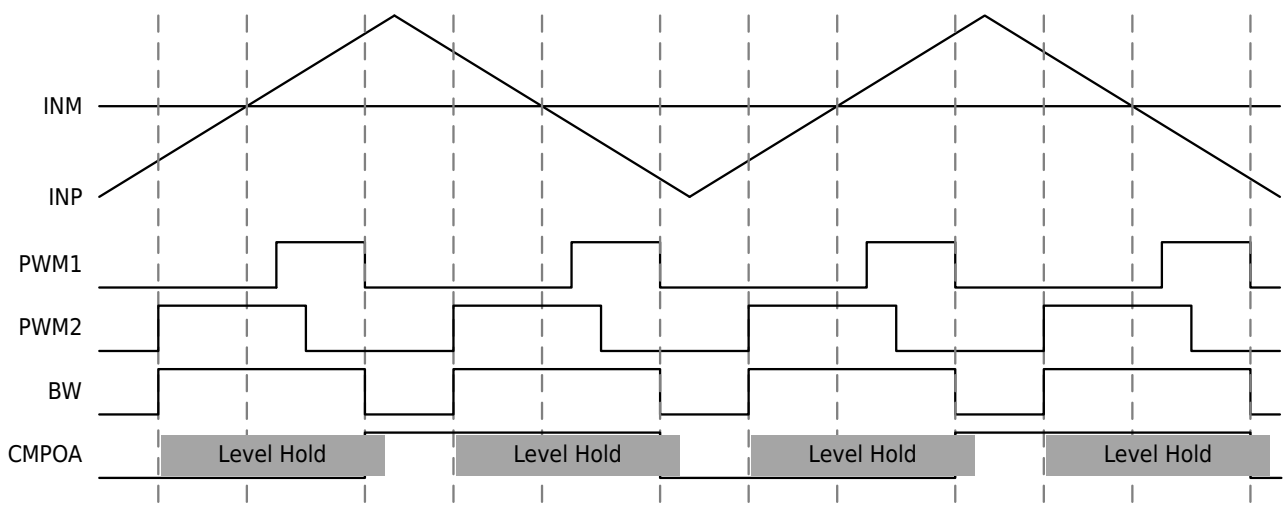


Figure 15-4 Window Output Diagram II

As shown in the diagram above, two PWMs are selected as Blank-Windows, both active at high level. When a Blank-Window is active, the output status of CMPOA is set to hold. Thus, when the Blank-Window signal is high, CMPOA maintains its previous state; when Blank-Window is at a low level, CMPOA outputs the normal comparator result, but the output is delayed by 1 to 3 CMPCLK periods due to synchronization and sampling.

2. Edge Mode

When windowed output is enabled (set the CMP_OCR.BWEN bit to 1) and edge mode is selected (set the CMP_OCR.BWMD bit to 1), select one or more timer PWMs as Blank-Windows through the CMP_BWSR1.CTWS bit, and select the active edge of the window signal through the CMP_BWSR2.TWEG. In this way, when a Blank-Window signal is active at the specified edge, the windowed output CMPOA is fixed to the state set by the CMP_OCR.BWOL bit and maintained for N CMPCLK periods set by the CMP_BWSR2.MSKW.

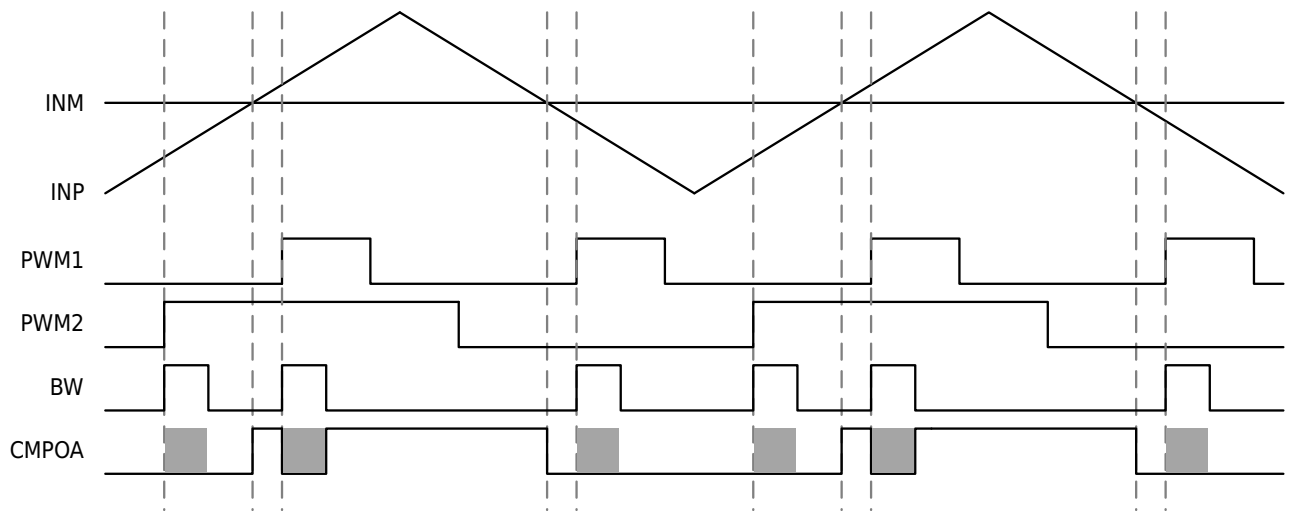


Figure 15-5 Window Output Diagram III

As shown in the diagram above, two PWMs are selected as Blank-Windows, with the rising edge selected as the active edge. When a Blank-Window is active, the output status of CMPOA is set to low level. As a result, a Blank-Window signal is generated at each rising edge of the PWM signal. When the Blank-Window signal is at a high level, CMPOA is fixed at a low level, and when the Blank-Window signal is at a low level, CMPOA outputs the comparison result normally.

15.3.5 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.

15.3.6 External Pin Output

The comparison result COUT output by the noise filter can be output to the external port VCOUT. Whether VCOUT 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 VCOUT is a general-purpose port in input status. Therefore, it must also be set to VCOUT output function through the port register.

15.3.7 Built-in DAC

A built-in 8-bit DAC provides an internal voltage DA1 for CMP.

The output voltage of the 8-bit DAC is:

$$V_{DA0} = V_{CCA} \times \text{Conversion Data} / 255$$

The DA conversion data is set by CMP_DADR.

Please follow the following process for setup when using:

- Step1.** Set CMP_DADR to set the conversion data.
- Step2.** Set CMP_DACR.DAE to start the 8-bit DAC.
- Step3.** Wait for the DA conversion stabilization time. (For specific values, please refer to the "Electrical Specifications" section of the Datasheet)
- Step4.** Set CMP_DACR.DAOE to allow analog output.
- Step5.** Set the comparator.

15.3.8 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.

15.4 Register Description

15.4.1 Register List

CMP Base Address: 0x4000C800

Table 15-3 Voltage Comparator (CMP) Register Offset

Offset	Register	Description	Width
0x00	CMP_MDR	Comparator Working Mode Register	8
0x01	CMP_FIR	Comparator Filter and Interrupt Register	8
0x02	CMP_OCR	Comparator Output Control Register	8
0x03	CMP_OMR	Comparator Output Monitoring Register	8
0x04	CMP_PMSR	Comparator Positive and Negative Input Selection Register	32
0x0C	CMP_DADR	Comparator DAC Data Register	16
0x0E	CMP_DACR	Comparator Digital to Analog Conversion Control Register	16
0x10	CMP_BWSR1	Comparator Blank Window Output Source Register 1	32
0x14	CMP_BWSR2	Comparator Blank Window Output Source Register 2	16

15.4.2 Comparator Working Mode Register (CMP_MDR)

Offset	Bit Position							
0x00	7	6	5	4	3	2	1	0
Reset	0x00							
Name	Reserved							CENA
Access								RW

Bit	Name	Function	Description	Access
7:1	Reserved	Reserved	-	-
0	CENA	Comparator Operation Permission	0b0: Comparator stopped 0b1: Comparator working  Note After setting CENA from 0 to 1 each time and setting CMP_PMSR, please wait for t_{START} (for specific values, please refer to the "Electrical Specifications" chapter in the "Datasheet") to ensure the CMP is stable before performing subsequent operations.	RW

15.4.3 Comparator Filter and Interrupt Register (CMP_FIR)

Offset	Bit Position							
0x01	7	6	5	4	3	2	1	0
Reset	0x00							
Name	CRF	CFF	EDGS		CIEN	FCKS		
Access	RW	RW	RW		RW	RW		

Bit	Name	Function	Description	Access
7	CRF	Comparator Output Rising Edge Flag	Write 1 to clear the flag, write 0 has no effect Read: 0b0: Rising edge of comparator output COUT not detected 0b1: Rising edge of comparator output COUT detected	RW
6	CFF	Comparator Output Falling Edge Flag	Write 1 to clear the flag, write 0 has no effect Read: 0b0: Falling edge of comparator output COUT not detected 0b1: Falling edge of comparator output COUT detected	RW
5:4	EDGS	Comparator Output Edge Detection Selection	0b00: Not detect edges of comparator output COUT 0b01: Detect rising edge of comparator output COUT 0b10: Detect falling edge of comparator output COUT 0b11: Detect rising and falling edges of comparator output COUT  Note 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.	RW
3	CIEN	Comparator Interrupt Permission	0b0: Comparator interrupt disabled 0b1: Comparator interrupt enabled	RW

Bit	Name	Function	Description	Access
2:0	FCKS	Comparator Result Filtering Sample Selection	● 0b000: Noise filter not used ● 0b001: Noise filter used, sampling through HCLK ● 0b010: Noise filter used, sampling through HCLK/2 ● 0b011: Noise filter used, sampling through HCLK/4 ● 0b100: Noise filter used, sampling through HCLK/8 ● 0b101: Noise filter used, sampling through HCLK/16 ● 0b110: Noise filter used, sampling through HCLK/32 ● 0b111: Noise filter used, sampling through HCLK/64  Note Please rewrite FCKS[2:0] when the comparator output is disabled (i.e., the CMP_OCR.COEN is "0"). When switching FCKS[2:0] from "0b000" to other values, please use the filtered output after 4 samples as the interrupt request or peripheral trigger event. Changing FCKS[2:0] 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.	RW

15.4.4 Comparator Output Control Register (CMP_OCR)

Offset	Bit Position							
0x02	7	6	5	4	3	2	1	0
Reset	0x00							
Name	BWOL		BWMD	BWEN	Reserved	CPOE	COPS	COEN
Access	RW		RW	RW		RW	RW	RW

Bit	Name	Function	Description	Access
7:6	BWOL	Blank-Window Output Level	0b00: During the active (output closed) period of the Blank-Window, the outputs a low level. 0b01: During the active (output closed) period of the Blank-Window, the outputs a high level. 0b1x: During the active (output closed) period of the Blank-Window, the output is held.	RW
5	BWMD	Blank-Window Mode	0b0: Level mode 0b1: Edge mode	RW
4	BWEN	Blank-Window Enable	0b0: Disable 0b1: Enable	RW
3	<i>Reserved</i>	<i>Reserved</i>	-	-
2	CPOE	VCOUT Output Enable	0b0: VCOUT output disabled 0b1: VCOUT output enabled	RW
1	COPS	Comparator Output Polarity Selection	0b0: Comparator non-inverting output 0b1: Comparator inverting output  Note 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.	RW
0	COEN	Comparator Output Permissible	0b0: Comparator output disabled (comparator output fixed at low level) 0b1: Comparator output enabled	RW

15.4.5 Comparator Output Monitoring Register (CMP_OMR)

Offset	Bit Position							
0x03	7	6	5	4	3	2	1	0
Reset	0x00							
Name	Reserved							CMON
Access								RO

Bit	Name	Function	Description	Access
7:1	Reserved	Reserved	-	-
0	CMON	Compare Result Monitoring	In normal comparative mode: 0b0: Positive voltage is lower than negative voltage 0b1: Positive voltage is higher than negative voltage	RO

15.4.6 Comparator Positive and Negative Input Selection Register (CMP_PMSR)

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								CVSL								Reserved								RVSL							
Access									RW																RW							

Bit	Name	Function	Description	Access
31:25	Reserved	Reserved	-	-
24:16	CVSL	Positive Input Selection	Select the positive input voltage of the voltage comparator. The positive inputs are detailed in the CMP Analog Input List: 0b000000000: No positive voltage input 0b000000001: Select INP1 as positive voltage 0b000000010: Select INP2 as positive voltage 0b000000100: Select INP3 as positive voltage 0b000001000: Select INP4 as positive voltage 0b000010000: Select INP5 as positive voltage 0b000100000: Select INP6 as positive voltage 0b001000000: Select INP7 as positive voltage 0b010000000: Select INP8 as positive voltage 0b100000000: Select INP9 as positive voltage - Others: Setting prohibited  Note Please rewrite CVSL when the comparator output is disabled (i.e., CMP_OCR.COEN = 0). After each re-write of CVSL, wait 300ns before enabling the comparator output (i.e., setting CMP_OCR.COEN to 1). Changing CVSL may cause interrupts or peripheral trigger events, so please set this register when interrupts are disabled or peripheral trigger functions are inactive. After setting the register, clear the corresponding interrupt flags.	RW
15:9	Reserved	Reserved	-	-
8:0	RVSL	Negative Input Selection	Select the negative input voltage of the voltage comparator. The selected inputs are detailed in the CMP Analog Input List: 0b000000000: No negative voltage input 0b000000001: Select INM1 as negative voltage 0b000000010: Select INM2 as negative voltage 0b000000100: Select INM3 as negative voltage 0b000001000: Select INM4 as negative voltage 0b000010000: Select INM5 as negative voltage	RW

Bit	Name	Function	Description	Access
8:0	RVSL	Negative Input Selection	● 0b000100000: Select INM6 as negative voltage ● 0b001000000: Select INM7 as negative voltage ● 0b010000000: Select INM8 as negative voltage ● 0b100000000: Select INM9 as negative voltage ● Others: Setting prohibited  Note Modify RVSL only when the comparator output is disabled (i.e., the CMP_OCR.COEN bit is set to 0). Please wait for 300ns after rewriting RVSL before enabling the comparator output (i.e., setting the CMP_OCR.COEN bit 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.	RW

15.4.7 Comparator DAC Data Register (CMP_DADR)

Offset	Bit Position															
0x0C	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0000															
Name	Reserved								DATA							
Access									RW							

Bit	Name	Function	Description	Access
15:8	Reserved	Reserved	-	-
7:0	DATA	DA Conversion Data	Data to be converted, with a value range of 0x00-0xFF	RW

15.4.8 Comparator Digital to Analog Conversion Control Register (CMP_DACR)

Offset	Bit Position															
0x0E	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0000															
Name	Reserved															DAE
Access																RW

Bit	Name	Function	Description	Access
15:1	Reserved	Reserved	-	-
0	DAE	DA Start Control	0b0: DA stop 0b1: DA enabled	RW

15.4.9 Comparator Blank Window Output Source Register 1 (CMP_BWSR1)

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								CTWP7	CTWP6	CTWP5	CTWP4	CTWP3	CTWP2	CTWP1	CTWP0	Reserved								CTWS7	CTWS6	CTWS5	CTWS4	CTWS3	CTWS2	CTWS1	CTWS0			
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	-	-
23:16	CTWPN (n=7-0)	Window Signal Polarity	When Blank-Window is in level mode (CMP_OCR.BWMD=0), set the active level of the window signal used to MASK the comparator output (this bit setting is only active when the corresponding CTWS is 1) : 0b0: Comparator output closed when window signal is high 0b1: Comparator output closed when window signal is low	RW
15:8	Reserved	Reserved	-	-
7:0	CTWSn (n=7-0)	Window Signal Enable	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: 0b0: Do not select timer PWM output as window signal 0b1: Select timer PWM output as window signal	RW

Table 15-20 Timer Window PWM List

CTWS Bit	CMP
7	TMR4_1_OUH
6	TMR4_1_OUL
5	TMR4_1_OVH
4	TMR4_1_OVL
3	TMR4_1_OWH
2	TMR4_1_OWL
1	TMRA_1_PWM2
0	TMRA_1_PWM1

15.4.10 Comparator Blank Window Output Source Register 2 (CMP_BWSR2)

Offset	Bit Position															
0x14	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0000															
Name	Reserved						TWEG		MSKW							
Access							RW		RW							

Bit	Name	Function	Description	Access
15:10	Reserved	Reserved	-	-
9:8	TWEG	Window Signal Edge	When Blank-Window is in edge mode (CMP_OCR.BWMD=1), set the active edge of the window signal used to MASK the comparator output. ● 0b00: No active edge ● 0b01: Comparator output closed at the rising edge of the window signal ● 0b10: Comparator output closed at the falling edge of the window signal ● 0b11: Comparator output closed at the changing edge of the window signal	RW
7:0	MSKW	Blank-Window Active Width	When Blank-Window is in edge mode (CMP_OCR.BWMD=1), set the width of the MASK comparator output. The settable range is 0-255. When the edge set by the TWEG bit is detected in the window signal, the comparator output will be closed and last for the number of CMPCLK periods set by the MSKW bit.	RW

15.5 Precautions

15.5.1 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.

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.

15.5.2 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_PMSR bit before clearing the CMP_MDR.CENA bit.

When DAC transitions from the working state to the module stop state, the DAC will continue to operate, and the power consumption is equivalent to the working state. To further reduce power consumption, please clear the CMP_DACR.DAE bit to 0.

15.5.3 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_PMSR bit before clearing the CMP_MDR.CENA bit.

When the chip enters stop low power mode, if DAC 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_DACR.DAE bit to 0.

16 Analog-to-Digital Converter (ADC)

16.1 Introduction

The 12-bit ADC is an analog-to-digital converter that adopts the successive approximation method. This MCU is equipped with 1 ADC units, a maximum of 16 channels. Analog input channels can be arbitrarily combined into a sequence for single scan or continuous scan conversion. 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.

16.2 Features

- High Performance
 - ▶ Configurable 12-bit, 10-bit and 8-bit resolution
 - ▶ The A/D conversion clock (ADCLK) frequency can be selected as 1, 2, 4, 8, 16, or 32 divisions of the peripheral clock (HCLK)
 - ▶ Supports 1Msps Sampling Rate
 - ▶ Sampling time is programmable
 - ▶ Each channel has an independent data register
 - ▶ Data registers can be configured for left/right alignment
 - ▶ Analog watchdog to monitor conversion results
 - ▶ The ADC module can be set to stop when not in use
- Analog Input Channel
 - ▶ Up to 16 external analog inputs
- 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
 - ▶ Dual sequence scan, with sequences A and B independently selecting trigger sources, and sequence B having higher priority than A
- 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

16.3 Functional Description

16.3.1 Function Block Diagram

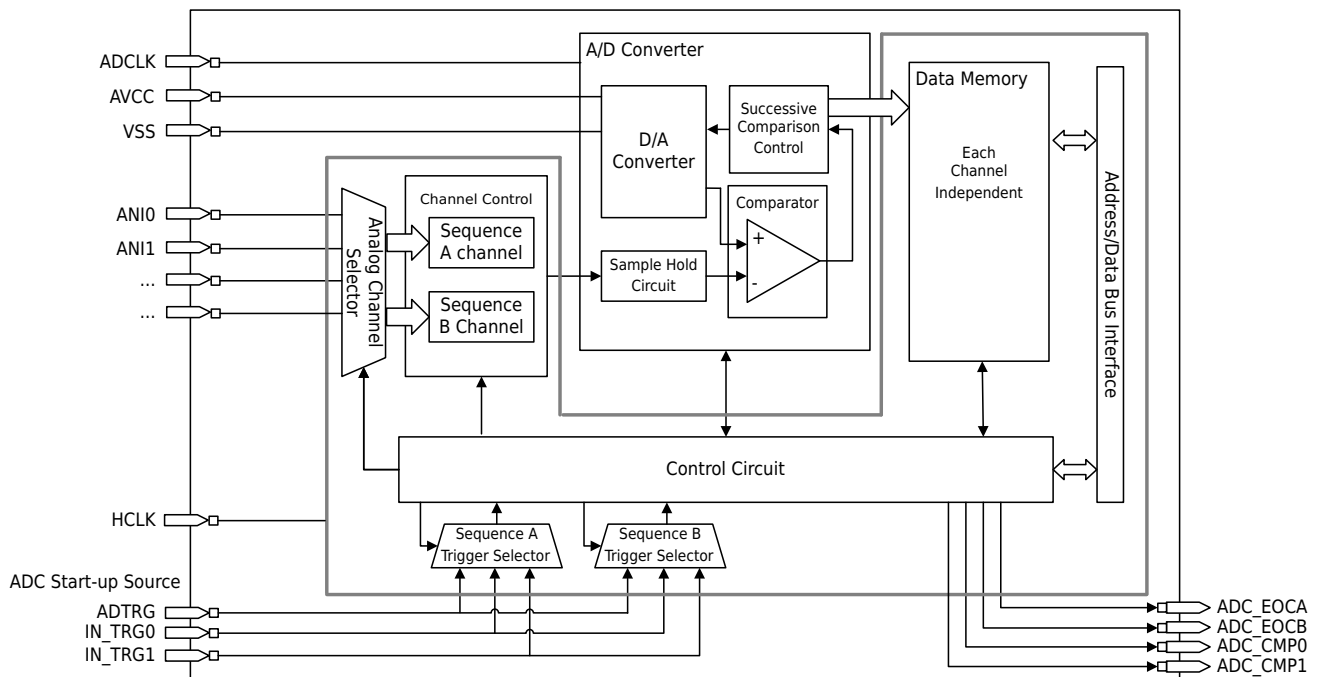


Figure 16-1 ADC Block Diagram

Table 16-1 ADC Input Signal Description

Pin Name	Function
AVCC/VSS	Power supply, ADC reference voltage VREFH is multiplexed with AVCC
ANI0-ANI15	Analog input
ADTRG	External pin start trigger input
IN_TRG0	Internal start trigger input
IN_TRG1	

16.3.2 ADC Clock

The ADC module requires two clocks: the analog circuit clock ADCLK and the digital interface clock HCLK. ADCLK can be set to the division factor 1, 2, 4, 8, 16, or 32 of HCLK. For details, refer to the "CMU ADC Clock Configuration Register (CMU_PERICKSEL)" section in the "Clock Module Unit (CMU)" chapter.



Notice

The ADCLK frequency should be set within 1MHz-32MHz.

16.3.3 Channel Selection

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. Refer to the ADC_CHSELRA, B0 descriptions for the correspondence between register bits and analog channels. 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.



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.

16.3.4 Trigger Source Selection

Sequences A, B independently select trigger sources. The selectable trigger sources include external ports ADTRG and internal events IN_TRG0/1. Among them, port ADTRG triggers are active on falling edges. IN_TRG0/1 are set by registers AOS_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.

16.3.5 Sequence A Single Scan Mode

Set ADC_CR0.MS to 0b00 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.EOCAR 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.

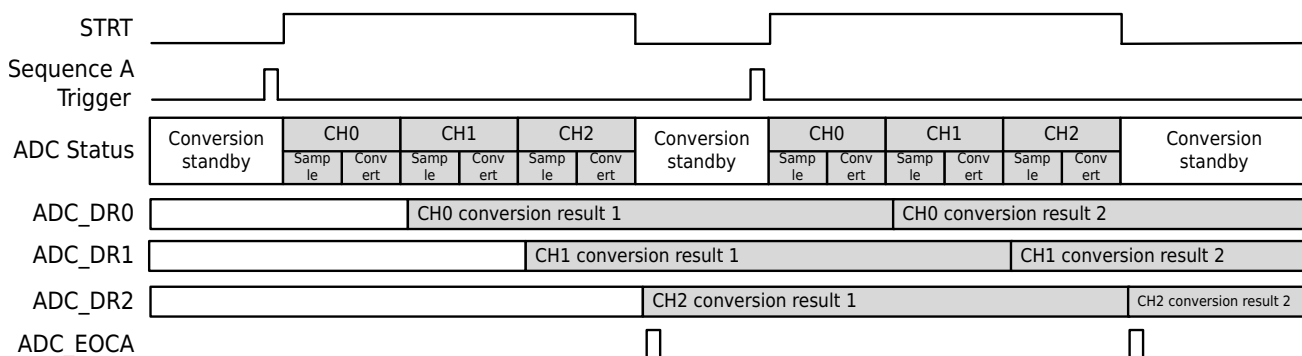


Figure 16-2 Sequence A Single Scan Mode

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 0b00 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.

16.3.6 Sequence A Continuous Scan Mode

Set ADC_CR0.MS to 0b01 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.

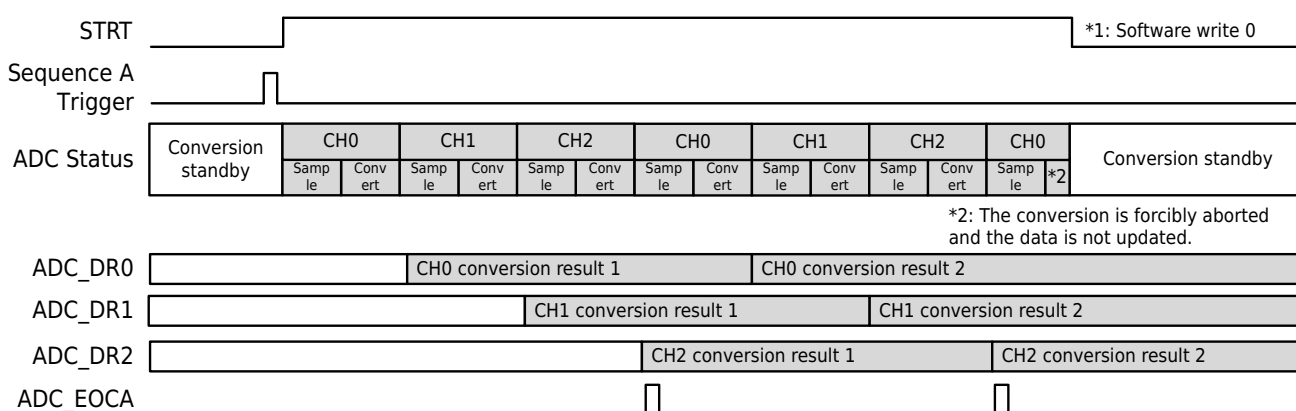


Figure 16-3 Continuous Scan

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 0b01 to select sequence A continuous scan mode;

Step3. Set sequence A channel selection register ADC_CHSELR;

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.

16.3.7 Dual Sequence Scan Mode

Set ADC_CR0.MS to 0b10 or 0b11 to select dual sequence scan mode, allowing both sequences A and B to be started by their respective trigger conditions.

When MS=0b10, sequences A and B are equivalent to two independent single scan sequences.

MS=0b11, sequence A is continuous scan mode, sequence 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.EOCAP 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 16-2 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 2. After Sequence B conversion completes, sequence A continues from interrupted channel	1. Sequence A conversion interrupted, sequence B conversion started 2. After Sequence B conversion completes, sequence A restarts from first channel
During Sequence B Conversion	Sequence A 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	

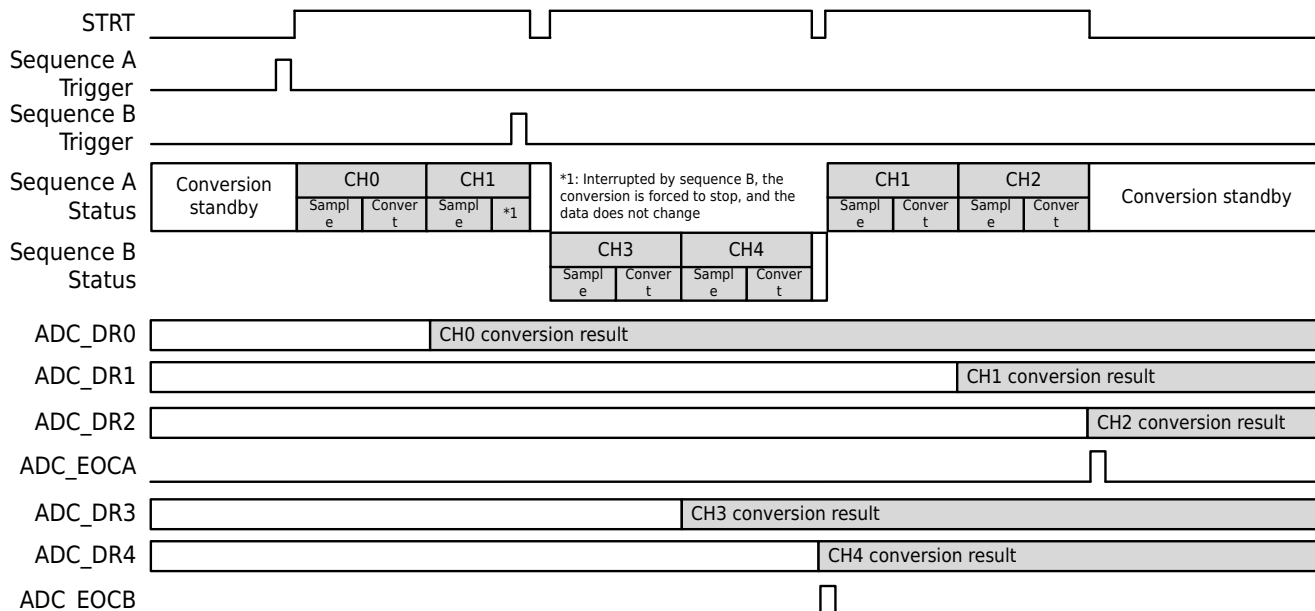


Figure 16-4 Dual Sequence Scan Mode (Sequence A Restarts from Interrupted Channel)

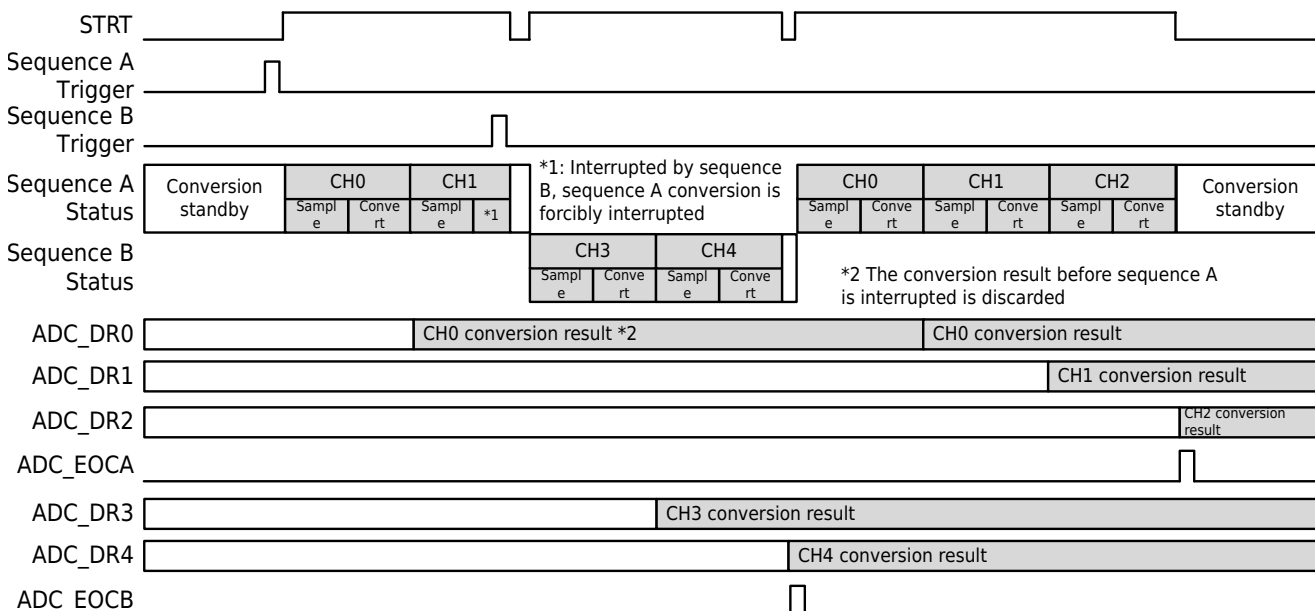


Figure 16-5 Dual Sequence Scan Mode (Sequence A Restarts from First Channel)

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 0b10 or 0b11 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.

16.3.8 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.

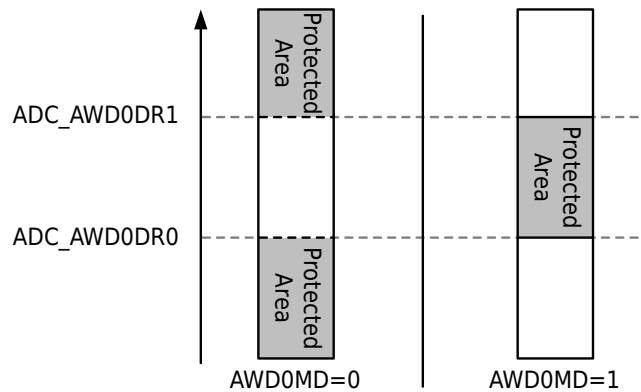


Figure 16-6 Analog Watchdog Protected Area (Comparative Mode)

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.

16.3.9 Sampling Time and Conversion Time for Analog Inputs

In single scan mode, ADC conversion can be started by software settings, internal triggers IN_TRG0/1, and external pin triggers 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 ADCLK for 12-bit precision, 11 ADCLK for 10-bit precision, and 9 ADCLK 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.

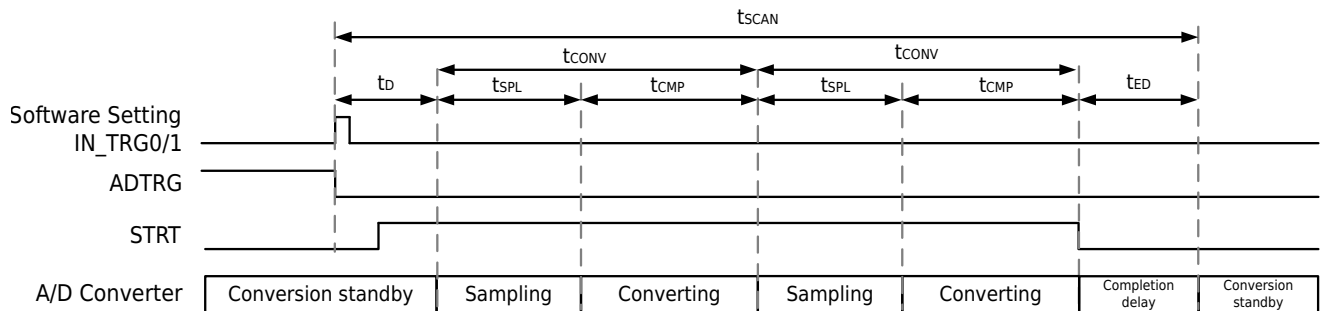


Figure 16-7 ADC Conversion Time Diagram

Table 16-3 ADC Conversion Time

Name		Description		Conditions		
				Synchronous Peripheral Trigger	External Pin Trigger	Software Trigger
t _D		Scan start processing time	ADC is idle, start conversion	4 ADCLK	2 HCLK + 4 ADCLK	3 ADCLK
			Sequence A conversion interrupted, start sequence B conversion	5 ADCLK	2 HCLK + 5 ADCLK	-
t _{CONV}	t _{SPL}	Sample time		ADC_SSTR.SST * ADCLK		
		Successive conversion time	12-bit resolution	13 ADCLK		
	10-bit resolution		11 ADCLK			
	8-bit resolution		9 ADCLK			
t _{ED}		Scan completion processing time		3 ADCLK		
t _{TD}		Minimum continuous trigger interval		Σt _{CONV} + 6 ADCLK		

16.3.10 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.

16.3.11 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.

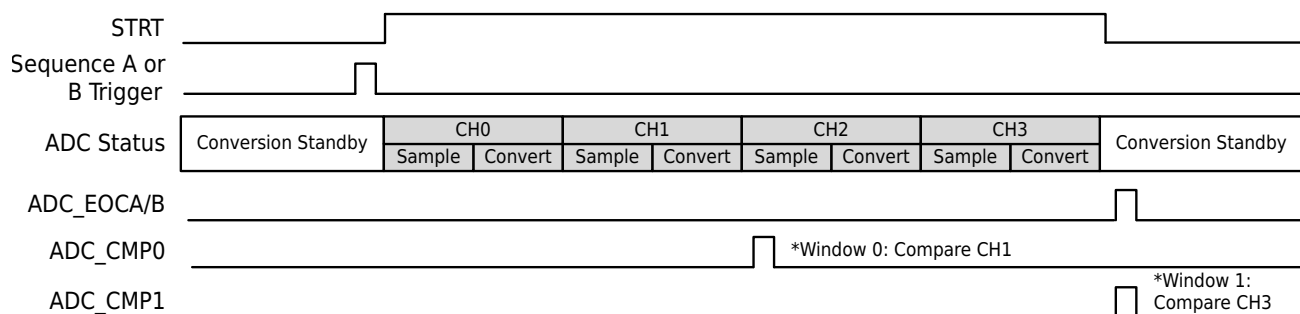


Figure 16-8 ADC Interrupt and Event Output Timing

16.4 Register Description

16.4.1 Register List

Base Address: 0x4000B800

Table 16-4 Analog-to-Digital Converter (ADC) Register Offset

Offset	Register	Description	Width
0x00	ADC_STR	ADC Start Register	8
0x02	ADC_CR0	ADC Control Register 0	16
0x04	ADC_CR1	ADC Control Register 1	16
0x0A	ADC_TRGSR	ADC Trigger Select Register	16
0x0C	ADC_CHSELRA	ADC Sequence A Channel Select Register	16
0x10	ADC_CHSELRB	ADC Sequence B Channel Select Register	16
0x20	ADC_SSTR	ADC Channel x Sampling Time Register	8
0x44	ADC_ISR	ADC Interrupt Flag Register	8
0x45	ADC_ICR	ADC Interrupt Enable Register	8
0x46	ADC_ISCLRR	ADC Interrupt Flag Clear Register	8
0x50+0x2*x	ADC_DRx	ADC Channel x Convert Data Register (x=0-15)	16
0xA0	ADC_AWDCR	ADC Analog Watchdog Control Register	16
0xA2	ADC_AWDSR	ADC Analog Watchdog Status Register	8
0xA3	ADC_AWDSCLR	ADC Analog Watchdog Status Clear Register	8
0xA4+0x8*x+0x2*y	ADC_AWDxDRy	ADC Analog Watchdog x Compare Data Register y (x=0-1, y=0-1)	16
0xA8+0x8*x	ADC_AWDxCHSR	ADC Analog Watchdog x Channel Select Register (x=0-1)	8

16.4.2 ADC 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	0b0: Stop conversion 0b1: Start conversion Set Conditions: - Software setting - The selected trigger condition occurs - During ADC conversion Clear Conditions: - Software clearing 0 - Automatically clears to 0 after conversion ends  Note - Writing 1 to STRT when STRT is 0 (ADC idle) generates a software trigger to start sequence A. - Writing 1 to STRT has no effect when STRT is 1 (ADC active). - 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. - Write 0 has no effect when STRT is 0.	RW

16.4.3 ADC 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								DFMT	CLREN	ACCSEL	Reserved			MS	
Access									RW	RW	RW				RW	

Bit	Name	Function	Description	Access
15:8	Reserved	Reserved	-	-
7	DFMT	Data Format	0b0: Right-align conversion data 0b1: Left-align conversion data	RW
6	CLREN	Data Register Automatic Clear	0b0: Disable automatic clear 0b1: Enable automatic clear Note 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.	RW
5:4	ACCSEL	Resolution Selection	0b00: 12-bit resolution 0b01: 10-bit resolution 0b10: 8-bit resolution 0b11: Setting prohibited	RW
3:2	Reserved	Reserved	-	-
1:0	MS	Mode Selection	0b00: Sequence A single scan mode, sequence B is invalid 0b01: Sequence A continuous scan mode, sequence B is invalid 0b10: Sequence A single scan mode, sequence B single scan mode 0b11: Sequence A continuous scan mode, sequence B single scan mode	RW

16.4.4 ADC 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	0b0: After being interrupted by sequence B, when sequence A restarts, it continues scanning from the interrupted channel. 0b1: After being interrupted by sequence B, when sequence A restarts, it starts to re-scan from the first channel.	RW
1:0	Reserved	Reserved	-	-

16.4.5 ADC Trigger Select 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	0b0: Disable sequence B internal or external pin trigger 0b1: Enable sequence B internal or external pin trigger Note External pin trigger is selected as active. If ADTRG transitions from "High" to "Low", a falling edge is detected, and the scan conversion starts. Please keep "Low" for at least 1.5*HCLK periods.	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: 0b00: ADTRG 0b01: IN_TRG0 0b10: IN_TRG1 0b11: IN_TRG0 + IN_TRG1 Note This is only valid in sequence B active mode and invalid in other modes. The interval between two triggers must be greater than or equal to the scan period t_{SCAN}, and if less, the trigger is disabled.	RW

Bit	Name	Function	Description	Access
7	TRGENA	Sequence A Trigger Enabled	● 0b0: Disable sequence A internal or external pin trigger ● 0b1: Enable sequence A internal or external pin trigger  Note External pin trigger is selected as active. If ADTRG transitions from "High" to "Low", a falling edge is detected, and the scan conversion starts. Please keep "Low" for at least 1.5*HCLK periods.	RW
6:2	<i>Reserved</i>	<i>Reserved</i>	-	-
1:0	TRGSELA	Sequence A Trigger Condition Selection	Trigger conditions for sequence A: ● 0b00: ADTRG ● 0b01: IN_TRG0 ● 0b10: IN_TRG1 ● 0b11: IN_TRG0 + IN_TRG1  Note 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. The interval between two triggers must be greater than or equal to the scan period t_{SCAN}, and if less, the trigger is disabled.	RW

16.4.6 ADC Sequence A Channel Select Register (ADC_CHSELRA)

**Note**

Please set this register when ADC_STR.STRT is 0.

Offset	Bit Position															
0x0C	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0000															
Name	CHSELA															
Access	RW															

Bit	Name	Function	Description	Access
15:0	CHSELA	Conversion Channel Selection	For sequence A channel selection, each bit represents a channel. CHSELA[x] represents channel CHx, and any combination can be selected. ● 0b0: Corresponding channel is not selected ● 0b1: Corresponding channel is selected Bit14-Bit0 correspond to channels ANI14-ANI0, respectively, and Bit15 corresponds to channel ANI15.  Notice Please do not select the same channel for both sequences A, B. Unsupported channels need to be set to 0.	RW

16.4.7 ADC Sequence B Channel Select Register (ADC_CHSELRB)

**Note**

Please set this register when ADC_STR.STRT is 0.

Offset	Bit Position															
0x10	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0000															
Name	CHSELB															
Access	RW															

Bit	Name	Function	Description	Access
15:0	CHSELB	Conversion Channel Selection	For sequence B channel selection, each bit represents a channel. CHSELB[x] represents channel CHx, and any combination can be selected. ● 0b0: Corresponding channel is not selected ● 0b1: Corresponding channel is selected Bit14-Bit0 correspond to channels ANI14-ANI0, respectively, and Bit15 corresponds to channel ANI15.  Notice This is only valid in dual sequence scan mode. Please do not select the same channel for both sequences A, B. Unsupported channels need to be set to 0.	RW

16.4.8 ADC Channel x Sampling Time Register (ADC_SSTR)

**Note**

Please set this register when ADC_STR.STRT is 0.

Offset: 0x20

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	0x00-0x04: Setting prohibited 0x05-0xFF: Correspond to 5-255 ADCLK clock periods respectively Note When the ADCLK frequency is 20MHz, a sampling period is 50ns, and the initial conversion state has 11 sampling periods. Register can be set to adjust sampling time when external input impedance R_{AIN} is too large with insufficient sampling time or ADCLK frequency is low. The sample time should not be less than 5 periods. $SST \geq (R_{AIN} + R_{ADC}) * C_{ADC} * \ln(2^{N+2}) * f_{ADC} + 1$ Where: R_{AIN} represents external input impedance (Ω), R_{ADC} represents internal sampling switch resistance (Ω), C_{ADC} represents internal sampling and retention capacitance (F), N represents AD resolution (12/10/8), and f_{ADC} represents ADCLK frequency (Hz). For specific information, please refer to the relevant sections of the "Electrical Specifications" chapter in the Datasheet.	RW

16.4.9 ADC Interrupt Flag Register (ADC_ISR)

Offset	Bit Position							
0x44	7	6	5	4	3	2	1	0
Reset	0x00							
Name	Reserved						EOCBF	EOCAF
Access							RO	RO

Bit	Name	Function	Description	Access
7:2	Reserved	Reserved	-	-
1	EOCBF	Sequence B Conversion Complete Flag	0b0: The selected channels in sequence B have not been scanned completely 0b1: All selected channels in sequence B have been scanned completely  Note This bit is set to 1 after all selected channels in sequence B have been scanned completely.	RO
0	EOCAF	Sequence A Conversion Complete Flag	0b0: The selected channels in sequence A have not been scanned completely 0b1: All selected channels in sequence A have been scanned completely  Note This bit is set to 1 after all selected channels in sequence A have been scanned completely.	RO

16.4.10 ADC 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	0b0: Interrupt disabled 0b1: Interrupt enabled	RW
0	EOCAIEN	Sequence A Conversion Complete Interrupt Enabled	0b0: Interrupt disabled 0b1: Interrupt enabled	RW

16.4.11 ADC Interrupt Flag Clear Register (ADC_ISCLRR)

Offset	Bit Position							
0x46	7	6	5	4	3	2	1	0
Reset	0x00							
Name	Reserved						CLREOCBF	CLREOCAF
Access							WO	WO

Bit	Name	Function	Description	Access
7:2	Reserved	Reserved	-	-
1	CLREOCBF	Sequence B Conversion Complete Flag Reset	Write 1 to reset the EOCBF bit, write 0 has no effect; read always returns 0.	WO
0	CLREOCAF	Sequence A Conversion Complete Flag Reset	Write 1 to reset the EOCAF bit, write 0 has no effect; read always returns 0.	WO

16.4.12 ADC Channel x Convert Data Register (ADC_DRx) (x=0-15)

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: $0x50 + 0x2 \times x$

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															

16.4.13 ADC 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	0b00: Window combination inactive, ADC_CMP1 outputs independent comparison results for Window 1 0b01: Window combination active, ADC_CMP1 outputs logical OR of comparison results for Window 0 and Window 1 0b10: Window combination active, ADC_CMP1 outputs logical AND of comparison results for Window 0 and Window 1 0b11: Window combination active, ADC_CMP1 outputs logical XOR of comparison results for Window 0 and Window 1  Note 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.	RW
7	Reserved	Reserved	-	-
6	AWD1MD	Watchdog Window 1 Compare Mode	Window 1 Protected Area Selection: 0b0: Conversion result < AWD1DR0, or conversion result > AWD1DR1 0b1: Conversion result ≥ AWD1DR0, and conversion result ≤ AWD1DR1	RW
5	AWD1IEN	Watchdog Window 1 Interrupt Enabled	0b0: Watchdog Window 1 comparison interrupt ADC_CMP1 inactive 0b1: Watchdog Window 1 comparison interrupt ADC_CMP1 enabled	RW

Bit	Name	Function	Description	Access
4	AWD1EN	Watchdog Window 1 Compare Enabled	0b0: Watchdog Window 1 comparison function inactive 0b1: Watchdog Window 1 comparison function enabled	RW
3	<i>Reserved</i>	<i>Reserved</i>	-	-
2	AWD0MD	Watchdog Window 0 Compare Mode	Window 0 Protected Area Selection: 0b0: Conversion result < AWD0DR0, or conversion result > AWD0DR1 0b1: Conversion result ≥ AWD0DR0, and conversion result ≤ AWD0DR1	RW
1	AWD0IEN	Watchdog Window 0 Interrupt Enabled	0b0: Watchdog Window 0 comparison interrupt ADC_CMP0 inactive 0b1: Watchdog Window 0 comparison interrupt ADC_CMP0 enabled	RW
0	AWD0EN	Watchdog Window 0 Compare Enabled	0b0: Watchdog Window 0 comparison function inactive 0b1: Watchdog Window 0 comparison function enabled	RW

16.4.14 ADC 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	0b0: Compare result of Window 0 and Window 1 does not meet the combination condition 0b1: Compare result of Window 0 and Window 1 meets the combination condition  Note 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.	RO
3:2	<i>Reserved</i>	<i>Reserved</i>	-	-
1	AWD1F	Watchdog Window 1 Comparative Status Bit	0b0: Conversion result does not meet the comparison condition 0b1: Conversion result meets the comparison condition  Note 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.	RO
0	AWD0F	Watchdog Window 0 Comparative Status Bit	0b0: Conversion result does not meet the comparison condition 0b1: Conversion result meets the comparison condition  Note 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.	RO

16.4.15 ADC 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

16.4.16 ADC Analog Watchdog x Compare Data Register y (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	AWD0DR0 sets the low threshold for Window 0, and AWD0DR1 sets the high threshold for Window 0. AWD1DR0 sets the low threshold for Window 1, and AWD1DR1 sets the high threshold for Window 1. 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). ● Right-aligned data-12-bit resolution: Low 12 bits [11:0] are available ● Right-aligned data-10-bit resolution: Low 10 bits [9:0] are available ● Right-aligned data-8-bit resolution: Low 8 bits [7:0] are available ● Left-aligned data-12-bit resolution: High 12 bits [15:4] are available ● Left-aligned data-10-bit resolution: High 10 bits [15:6] are available ● Left-aligned data-8-bit resolution: High 8 bits [15:8] are available  Note The unused bits of AWD0DR0, AWD0DR1, AWD1DR0, AWD1DR1 need to be filled with 0.	RW

16.4.17 ADC Analog Watchdog x Channel Select 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	Reserved	Reserved	-	-
4:0	AWDCH	Watchdog Compare Channel Selection	Select the comparison channel for Window x. ● 0x00: ANI0 ● 0x01: ANI1 ● ● 0x0F: ANI15 ● Others: Setting prohibited  Notice Please do not set non-existent channels.	RW

16.5 Precautions

16.5.1 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.

16.5.2 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.

16.5.3 Precautions for Module Stop and Low Power Settings

The ADC module can be stopped and power consumption reduced by setting the CMU_FCG register. The ADC is initially in the stopped state. To start the ADC module, first set the corresponding bit in the CMU_FCG register to cancel the stop, and wait for 8μs 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.

16.5.4 ADC Conversion Analog Channel Input Pin Settings

When setting a chip pin as an ADC analog channel input, please first disable the digital function (PFSRxy.FSEL=1) of the corresponding pin. For specific descriptions, please refer to the "General-Purpose I/O (GPIO)" chapter.

16.5.5 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.

17 General Control Timer (Timer4)

17.1 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. 1 unit of Timer4 is carried in this product family.

17.2 Features

The following table outlines the basic functions and features of Timer4.

Table 17-1 Basic Functions and Features of Timer4

Waveform mode	Sawtooth wave, triangle wave
Basic function	Up/Down-counting direction
	Buffer function
	General PWM output
	Special event output triggers ADC conversion
	Hardware counter start
	Software-synchronized counter start
	EMB control
Interrupt type	General compare match interrupt
	Count period match interrupt
	Reload count match interrupt
	Special compare match interrupt

17.3 Functional Description

17.3.1 Timer4 Block Diagram

The figure below describes the basic architecture of the general control timer Timer4.

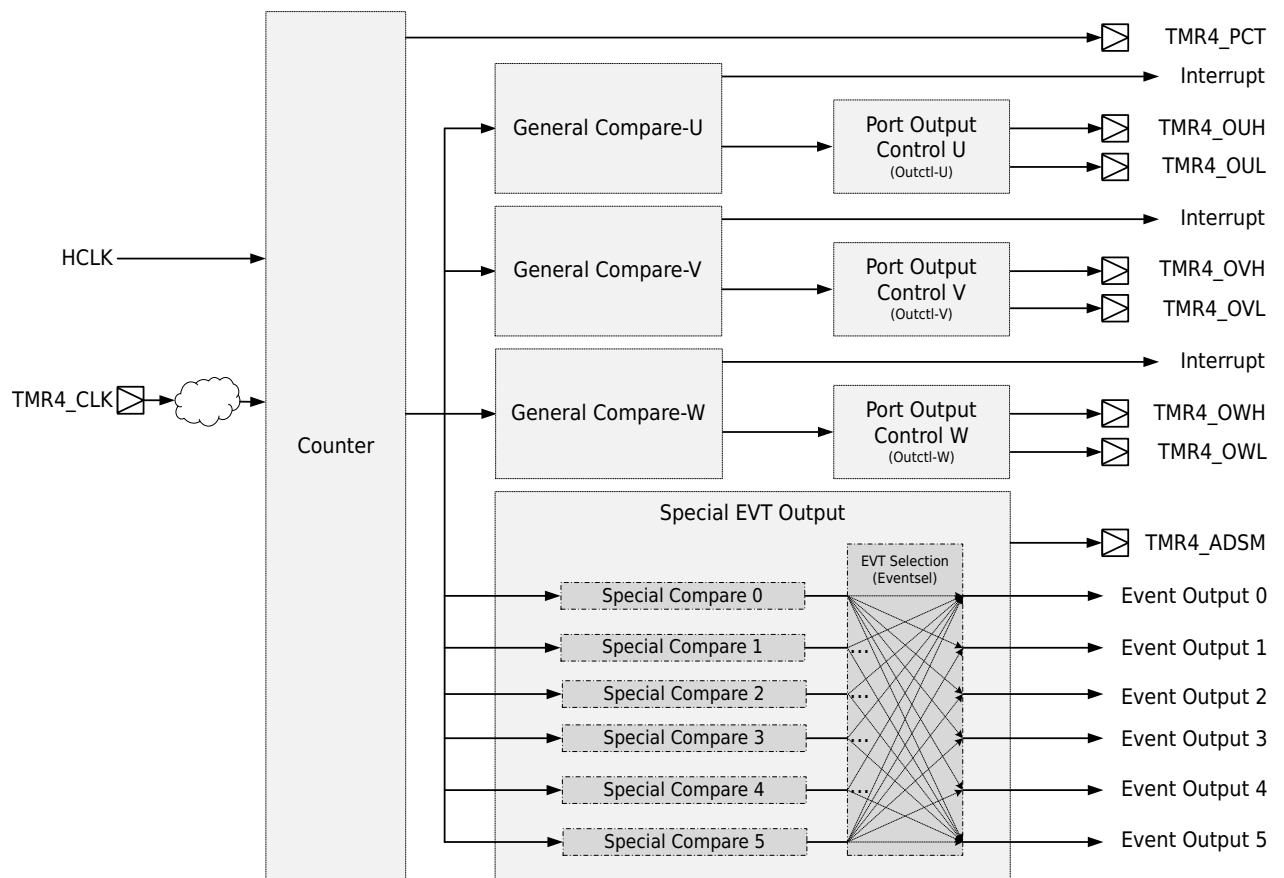


Figure 17-1 Timer4 Basic Block Diagram

The following table lists the input and output ports of Timer4.

Table 17-2 Timer4 Port List

Port Name	Direction	Function
TMR4_CLK	Input	Count clock input port
TMR4_PCT	Output	PWM periodic output monitoring
TMR4_ADSTM	Output	Special event output monitoring
TMR4_OUH	Output	PWM output port
TMR4_OUL		
TMR4_OVH		
TMR4_OVL		
TMR4_OWH		
TMR4_OWL		

17.3.2 Basic Actions

17.3.2.1 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.

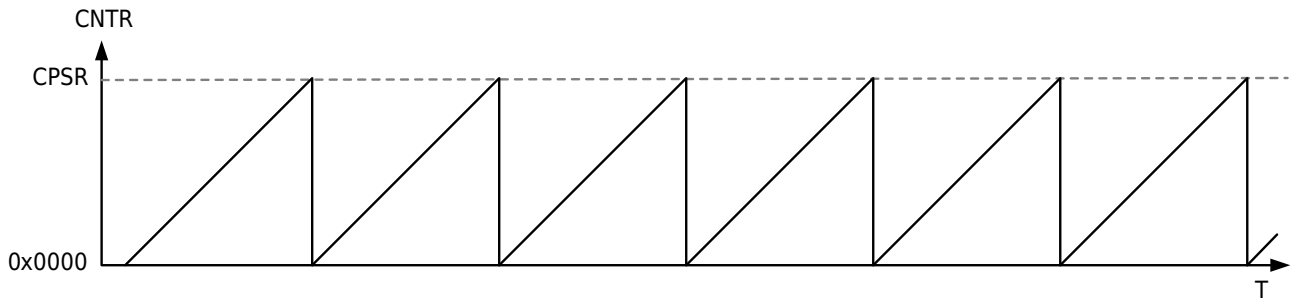


Figure 17-2 Timer4 Sawtooth Waveform

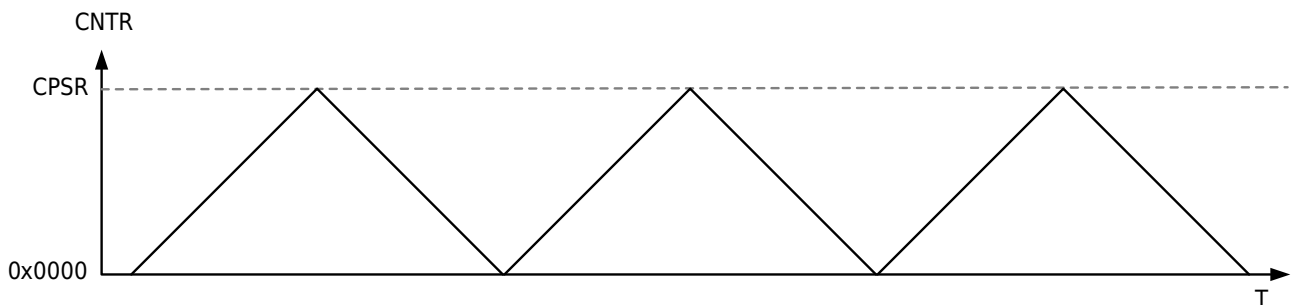


Figure 17-3 Timer4 Triangle Waveform

17.3.2.2 Counting Action

- The sawtooth wave counting operation and control process are shown in the figure below:

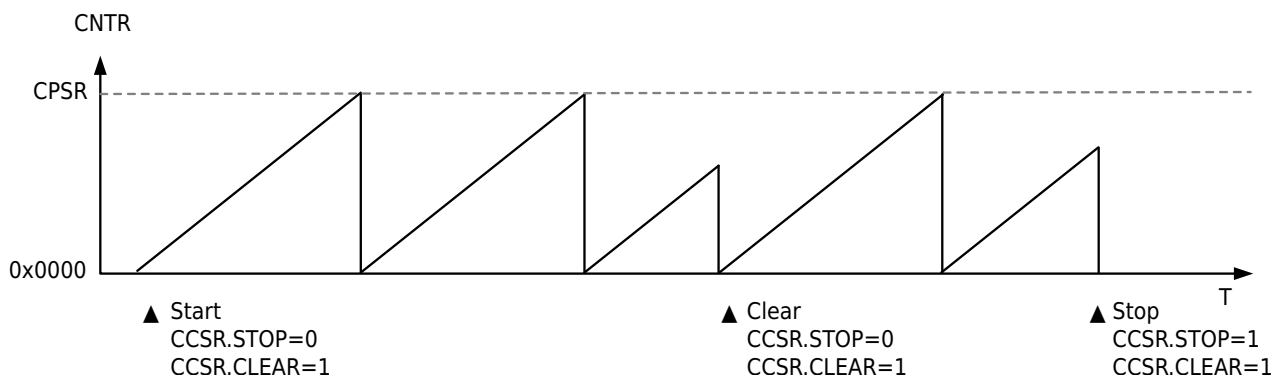


Figure 17-4 Timer4 Sawtooth Wave Mode Counting Action

- Setting mode `CCSR.MODE=0`.
- Set the count peak CPSR register.
- 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.
- 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:

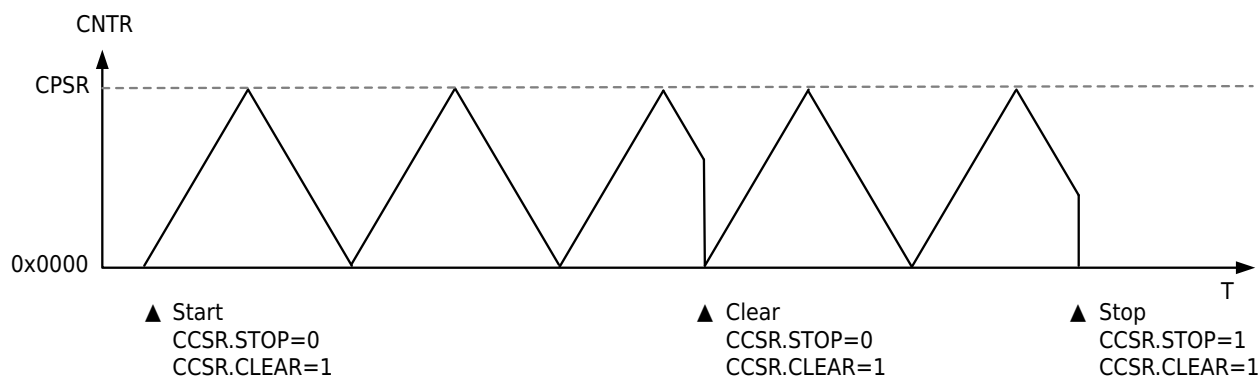


Figure 17-5 Timer4 Triangle Wave Mode Counting Action

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.

17.3.2.3 Compare Output

The compare output function consists of universal comparators U, V, and W. The general comparator includes TMR4_OCCR register, TMR4_OCSR register, TMR4_OCER register, and TMR4_OCMR register. 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 17-3 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> ; 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> ; 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	Internal Signal	Description
General Compa-rator V	in_opvh	When OCSRv.OCEH=0, the output high or low level is specified by OCSRv.OCPL; When OCSRv.OCEH=1, the PWM output is configured by the relevant bits of registers OCCRvh, OCMRvh and OCERv;
	in_opvl	When OCSRv.OCEL=0, the output high or low level is specified by OCSRv.OCPL; When OCSRv.OCEL=1, the PWM output is configured by the relevant bits of registers OCCRvl, OCMRvl and OCERv;
General Compa-rator W	in_opwh	When OCSRw.OCEH=0, the output high or low level is specified by OCSRw.OCPL; When OCSRw.OCEH=1, the PWM output is configured by the relevant bits of registers OCCRwh, OCMRwh and OCERw;
	in_opwl	When OCSRw.OCEL=0, the output high or low level is specified by OCSRw.OCPL; When OCSRw.OCEL=1, the PWM output is configured by the relevant bits of registers OCCRwl, OCMRwl and OCERw;

- The figure below is an example of the waveform output of the comparison output module in sawtooth counting mode.

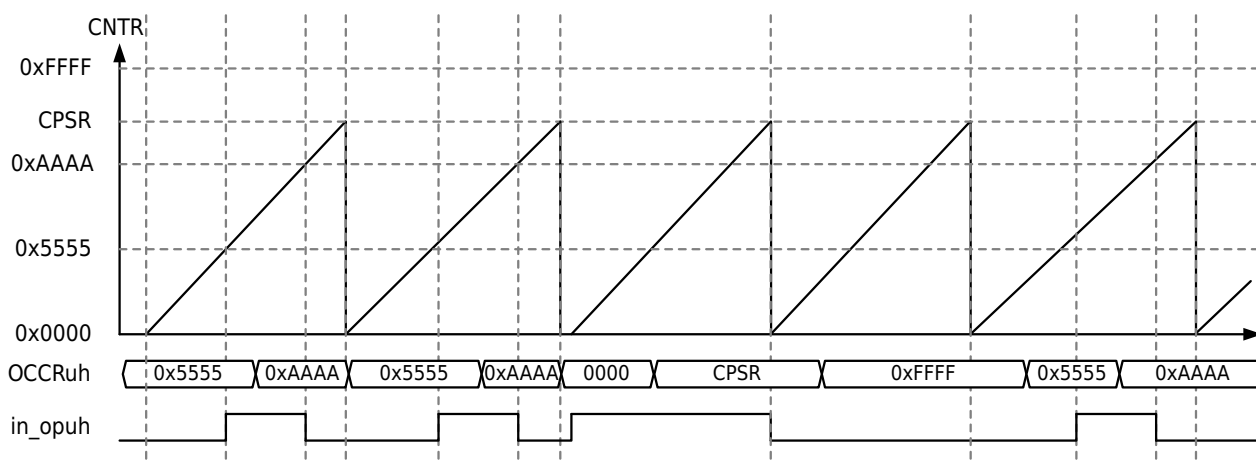


Figure 17-6 Sawtooth Mode Waveform Output Example

- The figure below is an example of the waveform output of the comparison output module in triangle wave counting mode.

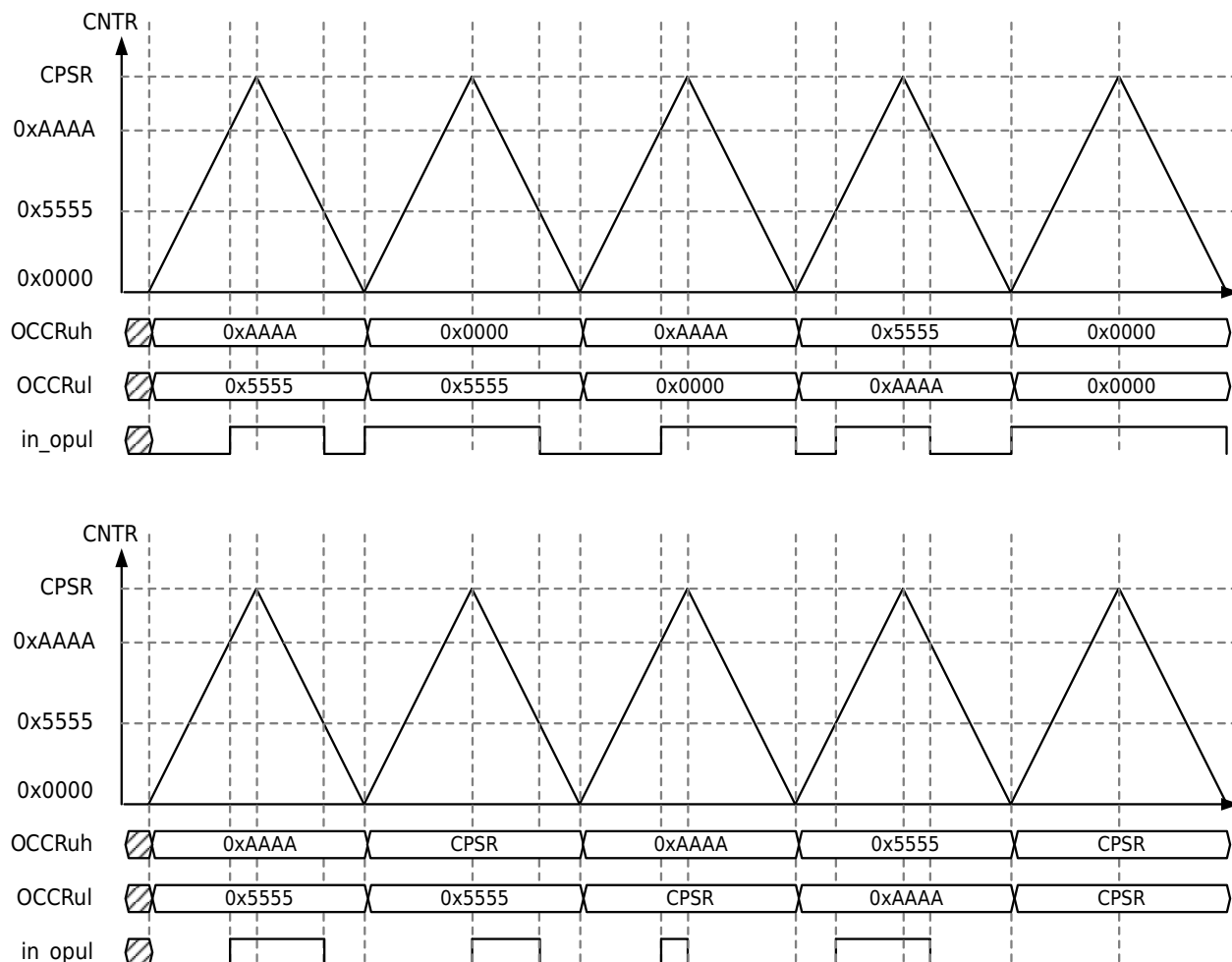


Figure 17-7 Triangular Wave Mode Waveform Output Example

17.3.3 Buffer Function

The Timer4 module provides buffer function for the CPSR Register, the OCCR Register, the OCMR Register, the SCCR Register, and the SCMR Register.

17.3.3.1 Period Value 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:

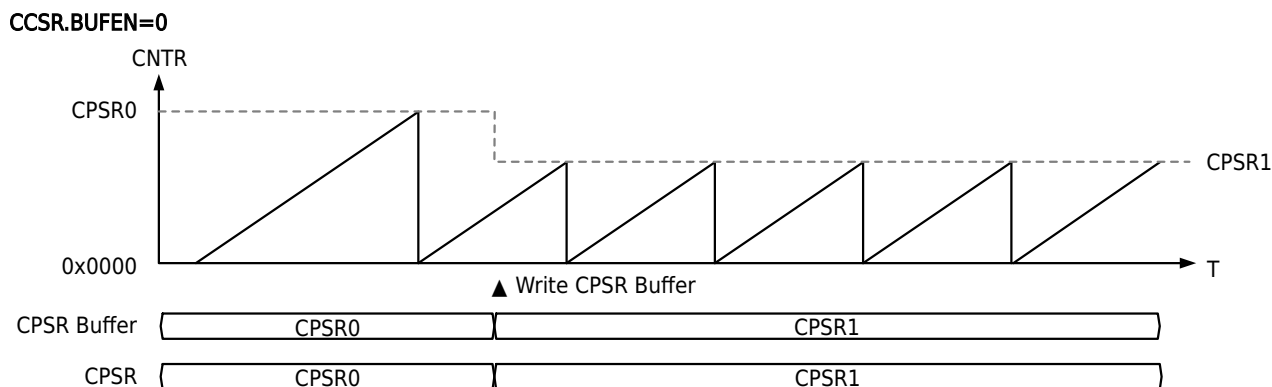


Figure 17-8 Modify Sawtooth Count Period when the Buffer is Disabled

**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, the buffer function is enabled, modify the operation of the count peak CPSR:

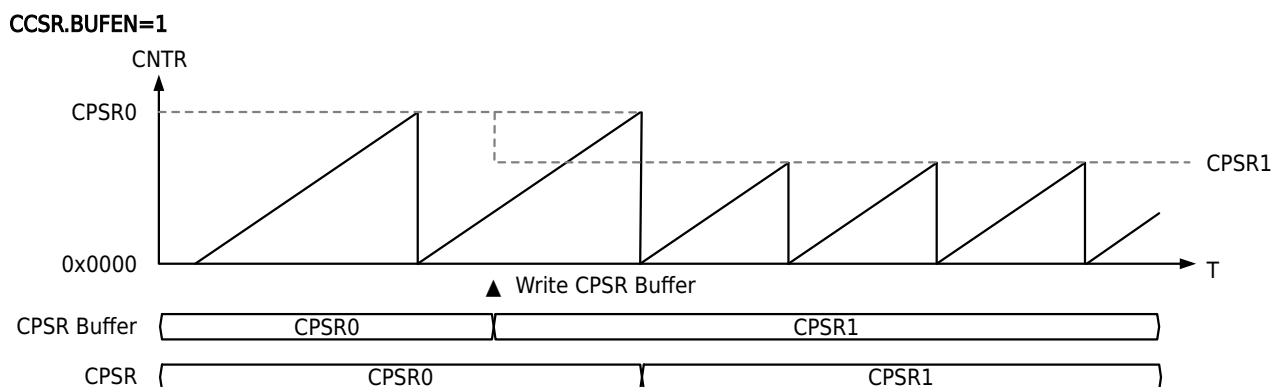


Figure 17-9 Modify Sawtooth Wave Count Period when the Buffer is Enabled

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, the buffer function is enabled, modify the operation of the count peak CPSR:

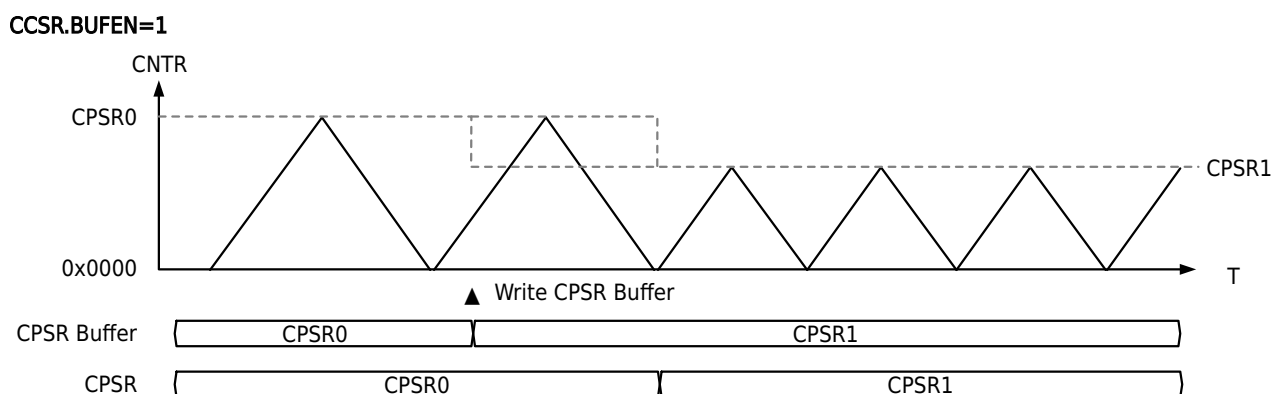


Figure 17-10 Modify Triangular Wave Count Period when Buffer is Enabled

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.

17.3.3.2 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).

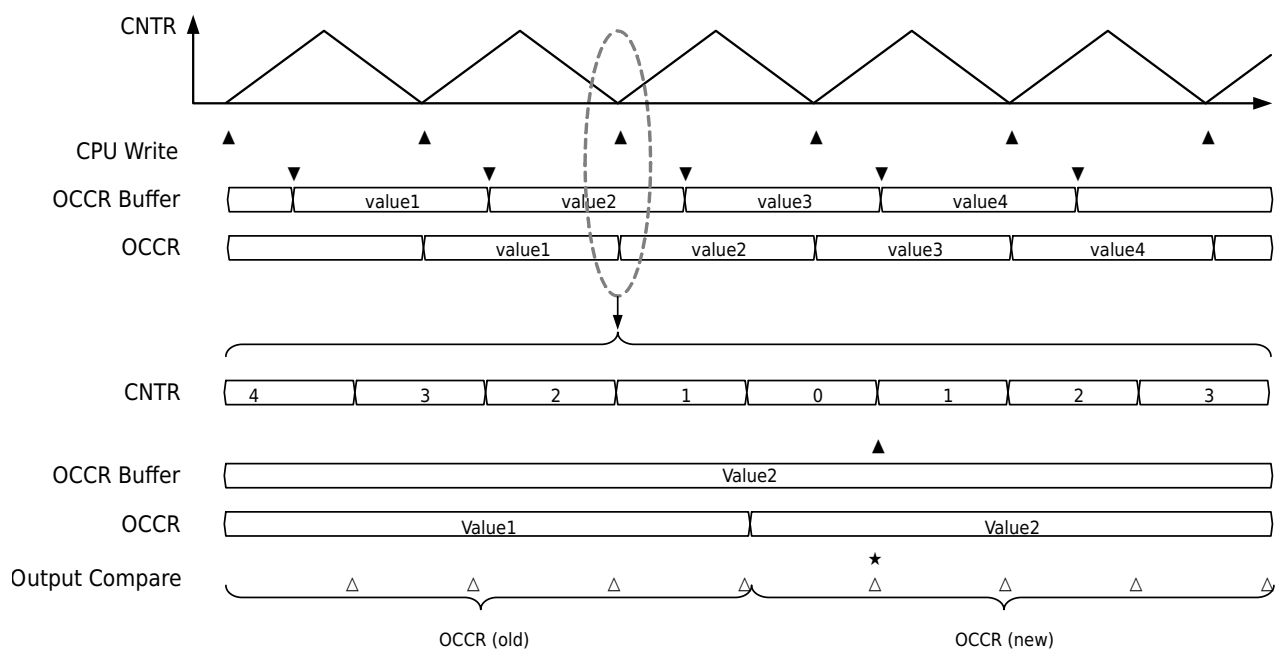


Figure 17-11 OCCR Buffer Data Transmission (Period Interval Response Link Disabled)

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 △, 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).

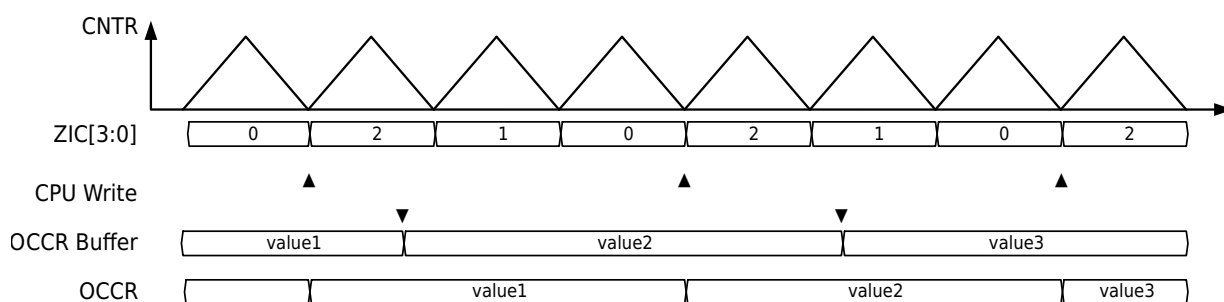


Figure 17-12 OCCR Buffer Data Transmission (Period Interval Response Link Enabled)

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.

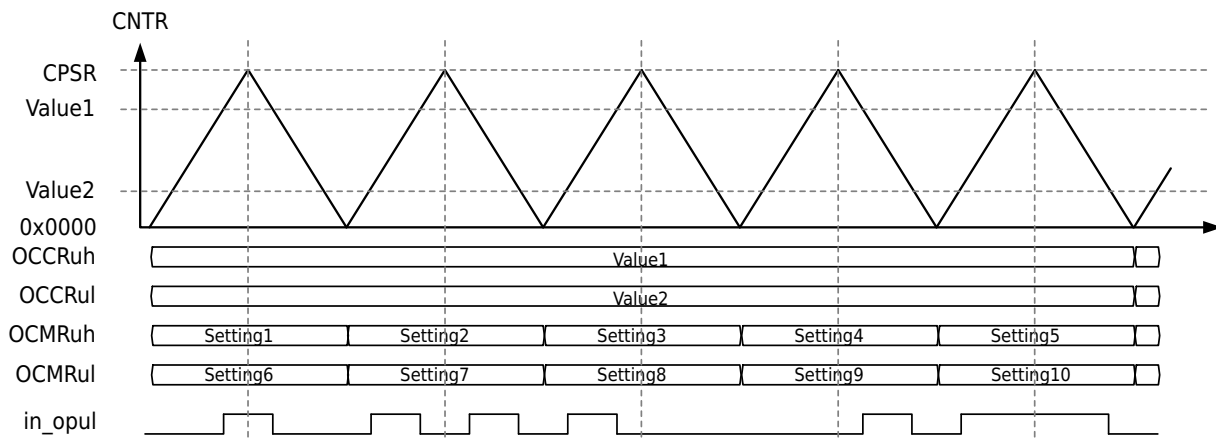


Figure 17-13 Output Compare Buffer Data Transfer (OCMR Buffer Enabled)

17.3.3.3 Special Compare Register Buffer Function

The SCCR register and SCMR register 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.

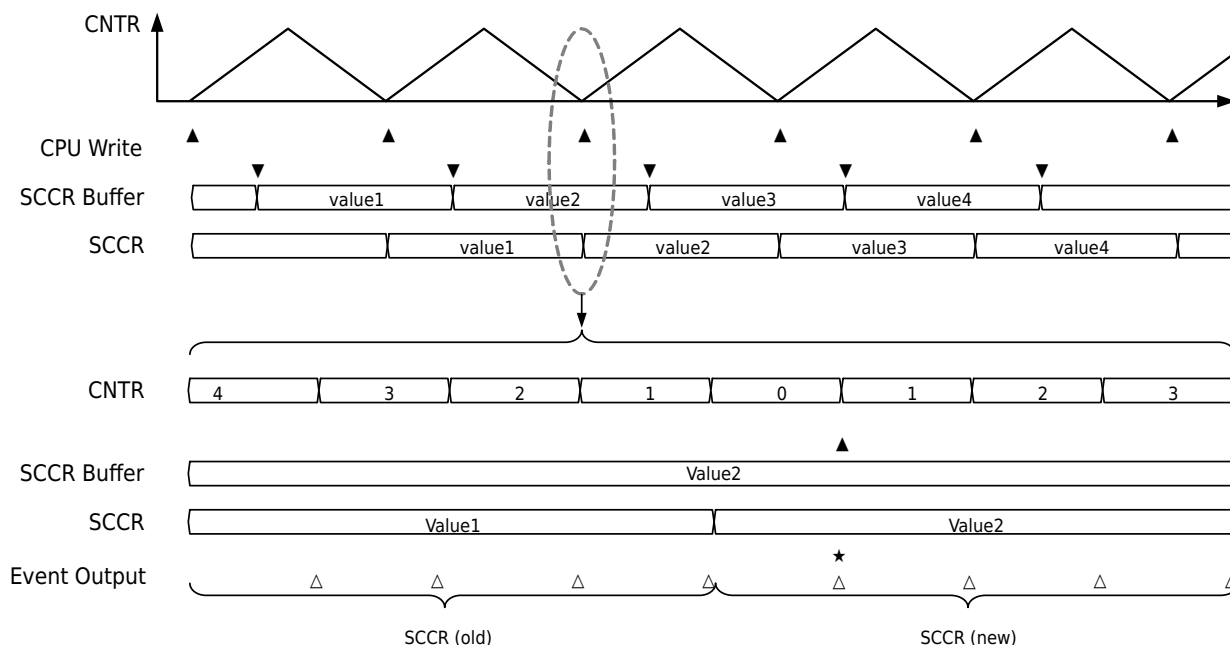


Figure 17-14 SCCR Buffer Transmission Operation (Period Interval Response Link Transmission Disabled)

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 Δ , a comparison operation is performed based on the configured value in the SCCR register and the counter value. After time $\star$ (when the counter value equals 0x0000), the special event output signal operates according to the new SCCR data. Before the $\star$ 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).

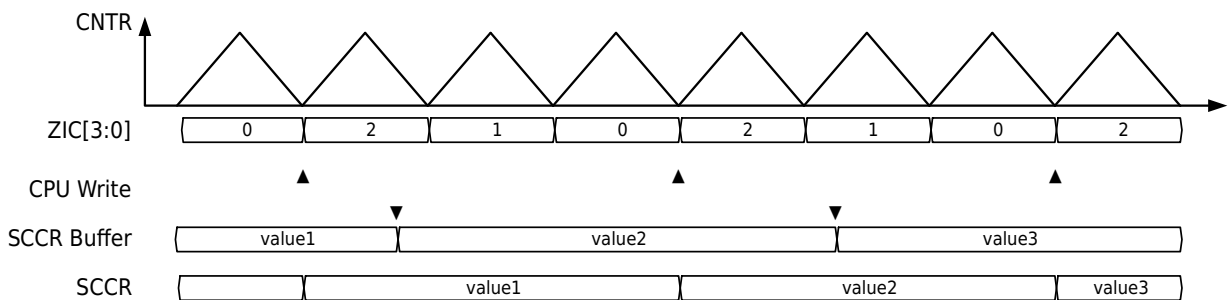


Figure 17-15 SCCR Buffer Transmission Operation (when Periodic Interval Response Link Transmission Enabled)

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 $\blacktriangle$ (valley point). At $\blacktriangledown$, 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.

17.3.4 General PWM Output

17.3.4.1 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_O*h, TMR4_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_O*h and TMR4_O*l. The port output control includes the TMR4_POCR register, TMR4_PSCR register, TMR4_PFSR register, TMR4_PDAR 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 17-4 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	

17.3.4.2 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.

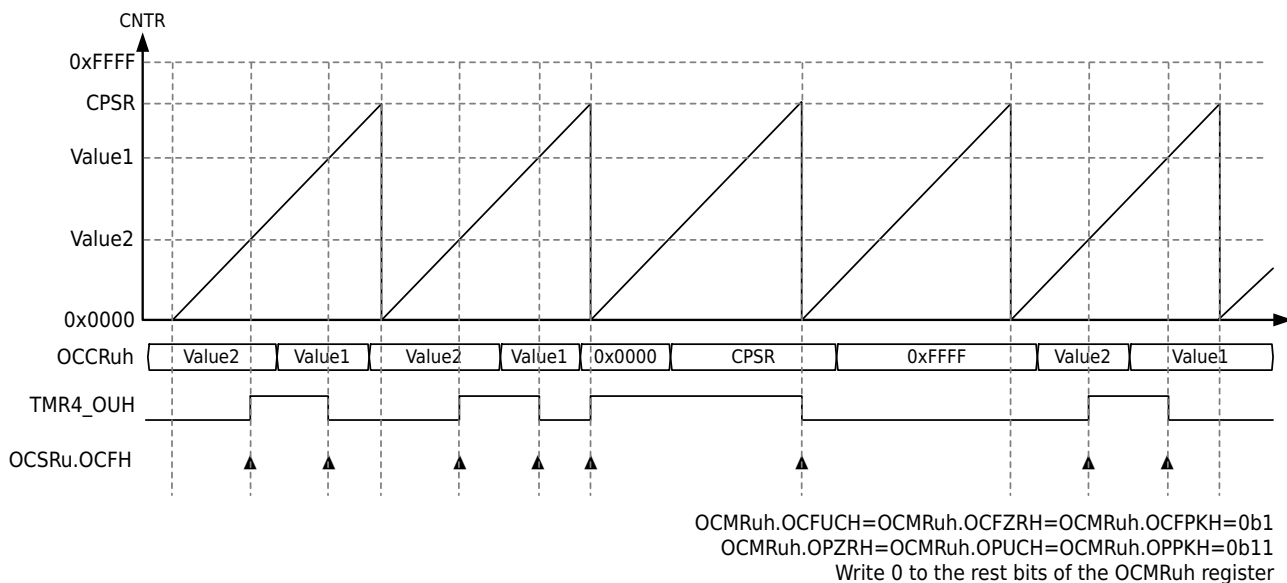


Figure 17-16 Sawtooth Independent PWM Output Example

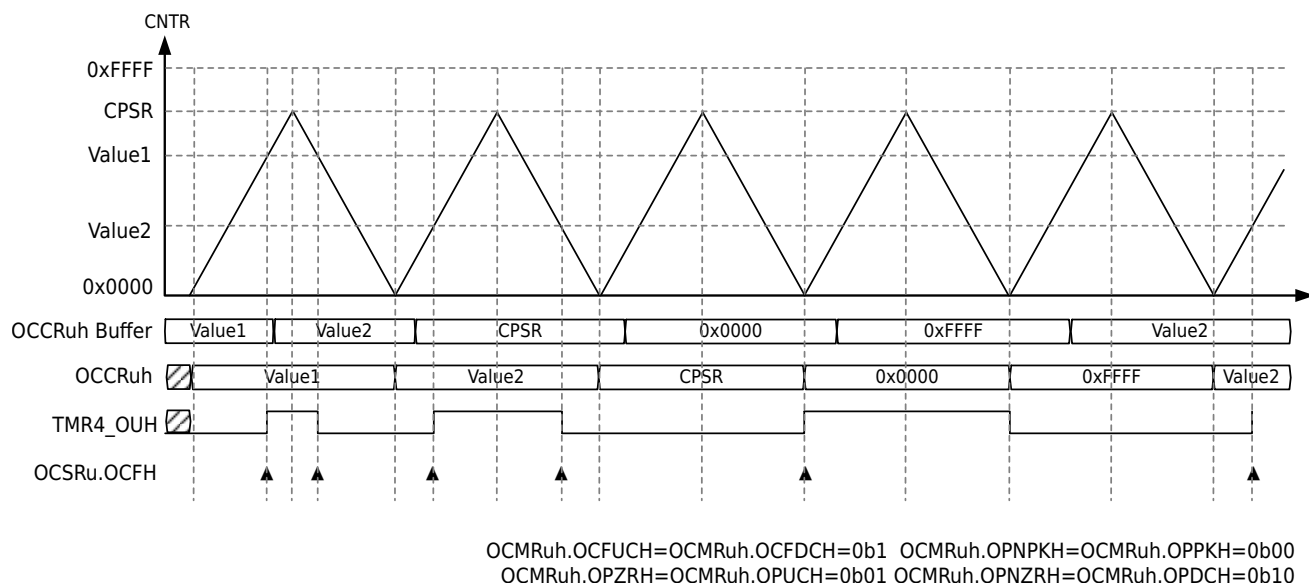


Figure 17-17 Triangle Independent PWM Output Example

**Note**

The pass-through mode means that the internal output signal (in_op*h, in_op*l) generated by the comparison match of the value of the general compare reference register (OCCR*h, OCCR*l) is directly output to the corresponding port (TMR4_O*h, TMR4_O*l).

17.3.4.3 Extended PWM Output

In pass-through mode (POCR.PWMMD=0b00), the port output status of TMR4_O*L can also be determined by the extension bit (Bit32-16) in the OCMR*l register. The setting of this extension bit is related to the OCCR*h reference, thereby realizing extended PWM output on the TMR4_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_OUH and TMR4_OUL ports in this mode.

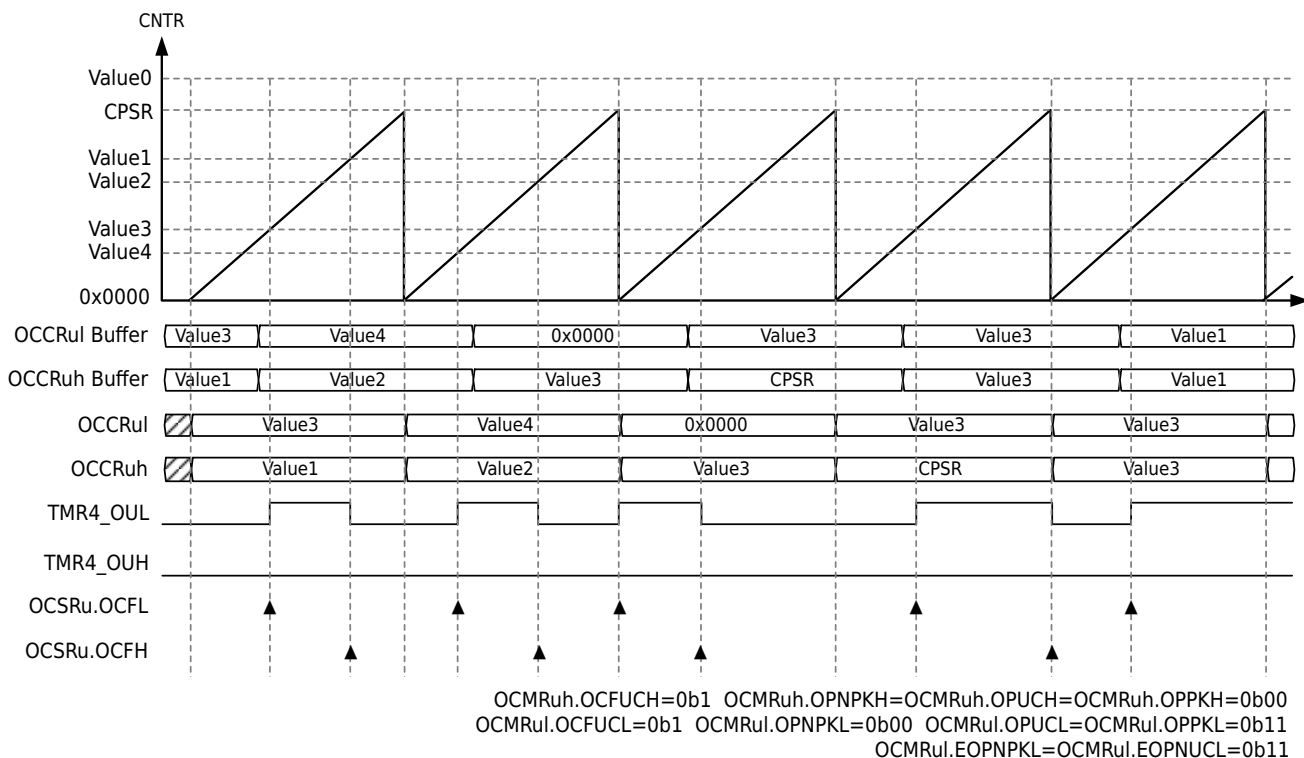


Figure 17-18 Sawtooth Wave Extended PWM Output

**Note**

In independent PWM output mode, the port status of TMR4_O*L is determined by Bit15-Bit4 of the OCMRxI register and is only related to the OCCR*I reference.

17.3.4.4 Complementary PWM Output

Software Setting for Complementary PWM Output

In the pass-through mode (POCR.PWMMD=0b00), directly set the OCCR*h and OCCR*I 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.

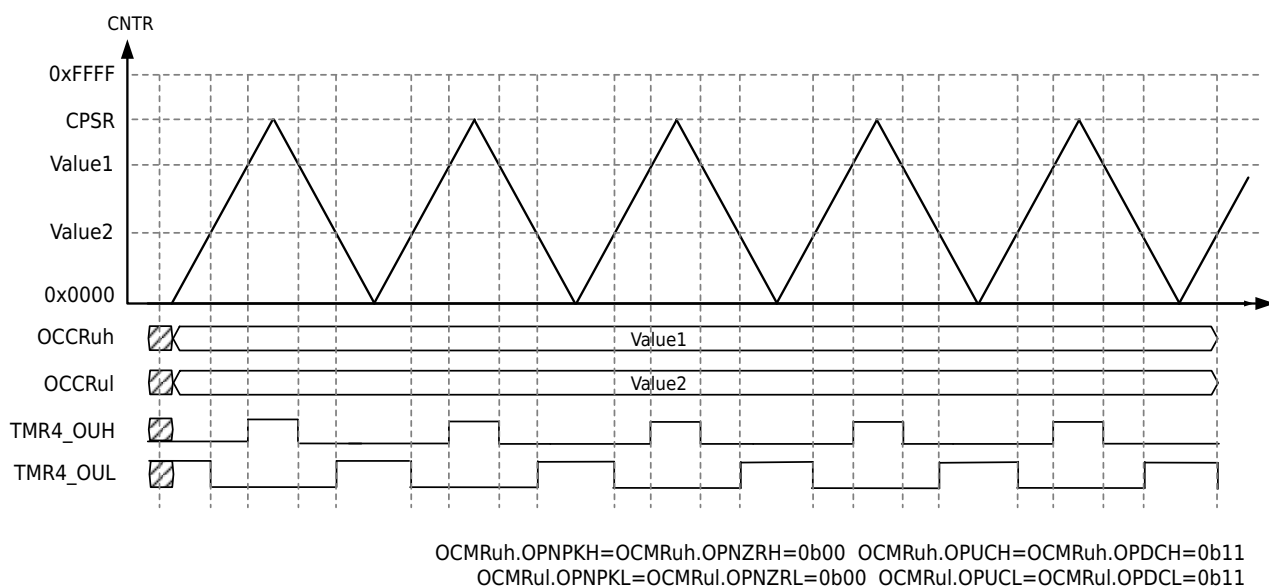


Figure 17-19 Software Implementation of Complementary PWM Output

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_O*H port is the same as that of in_op*I, and the polarity of the output of the TMR4_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_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_O*H outputs high level; If the falling edge of in_op*I is detected, the TMR4_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_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.

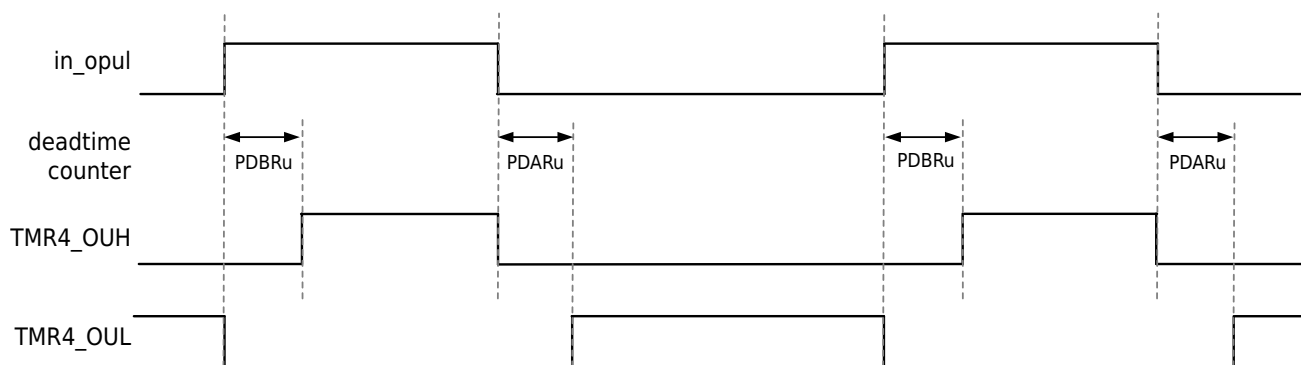


Figure 17-20 Complementary PWM Output in Dead-Zone Generator Mode

When the high-level pulse width of in_op*I is less than the dead time set by PDBR, only the TMR4_O*L output becomes low level. The condition for the TMR4_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_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 PDAR, only the TMR4_O*H output becomes low level. The condition for the TMR4_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_O*L output will continue to remain at a low level. As shown in the figure below.

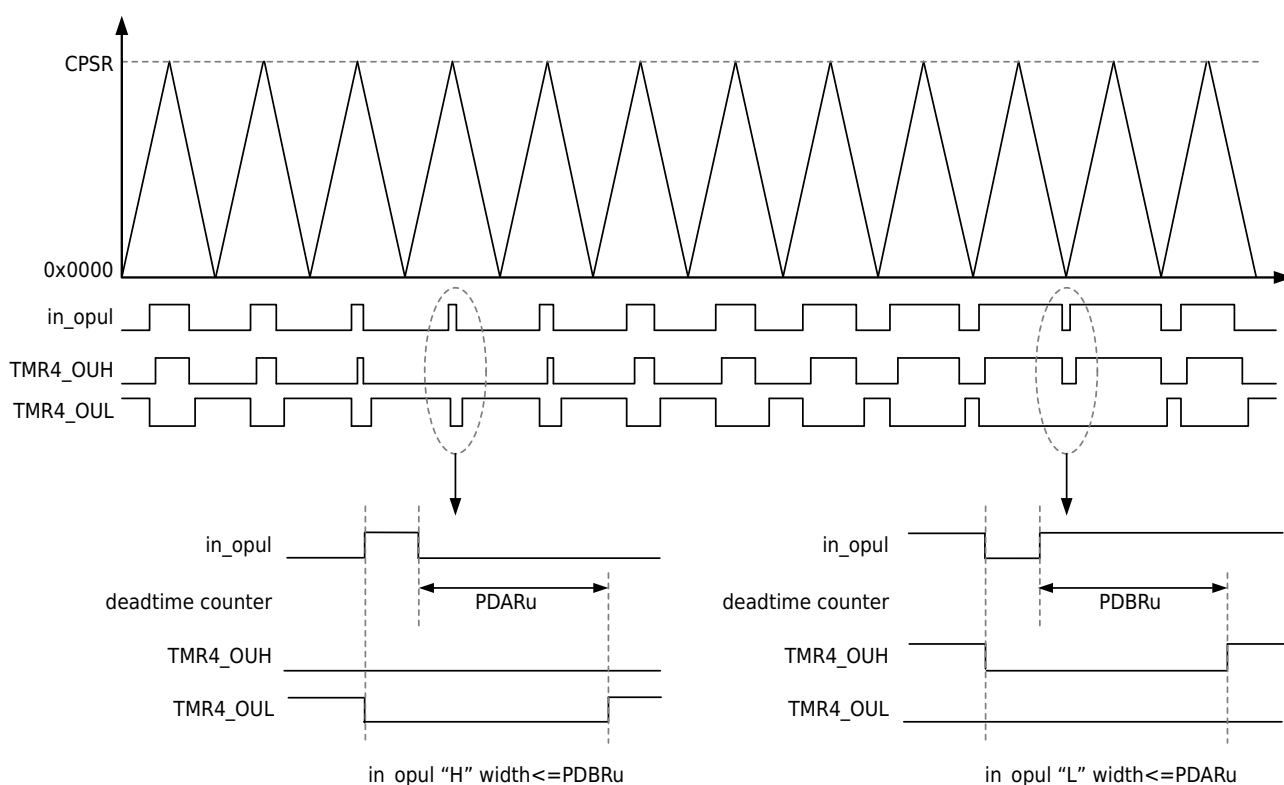


Figure 17-21 Waveform Output in Dead-Zone Timer Mode with Pulse Width Anomalies

Based on the hardware implementation of dead-zone output, pulse width monitoring can also be applied to the internal compare match signal (in_op^*l), enabling filtering control of the in_op^*l signal. This method, where the in_op^*l 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 PFSR register. When the pulse width of in_op^*l exceeds the time set in the PFSR register, the filter counter will delay the in_op^*l signal by the time set in PFSR 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^*l is detected, the filter counter loads the PFSR register value and starts measuring the high level width of in_op^*l . When the high level pulse width of in_op^*l is greater than the time set by the register PFSR, after the time set by PFSR, the TMR4_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_O*H output high level. If the falling edge of the signal in_op^*l is detected, the filter counter loads the PFSR register value and starts to measure the low-level width of in_op^*l . When the low-level pulse width of in_op^*l is greater than the time set by the register PFSR, after PFSR After the set time, the TMR4_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_O*L output high level. When the level pulse width of in_op^*l is less than the time set by PFSR register, the outputs TMR4_O*H and TMR4_O*L will remain unchanged.

The following figure shows an example of complementary PWM output in dead-zone counter filter mode.

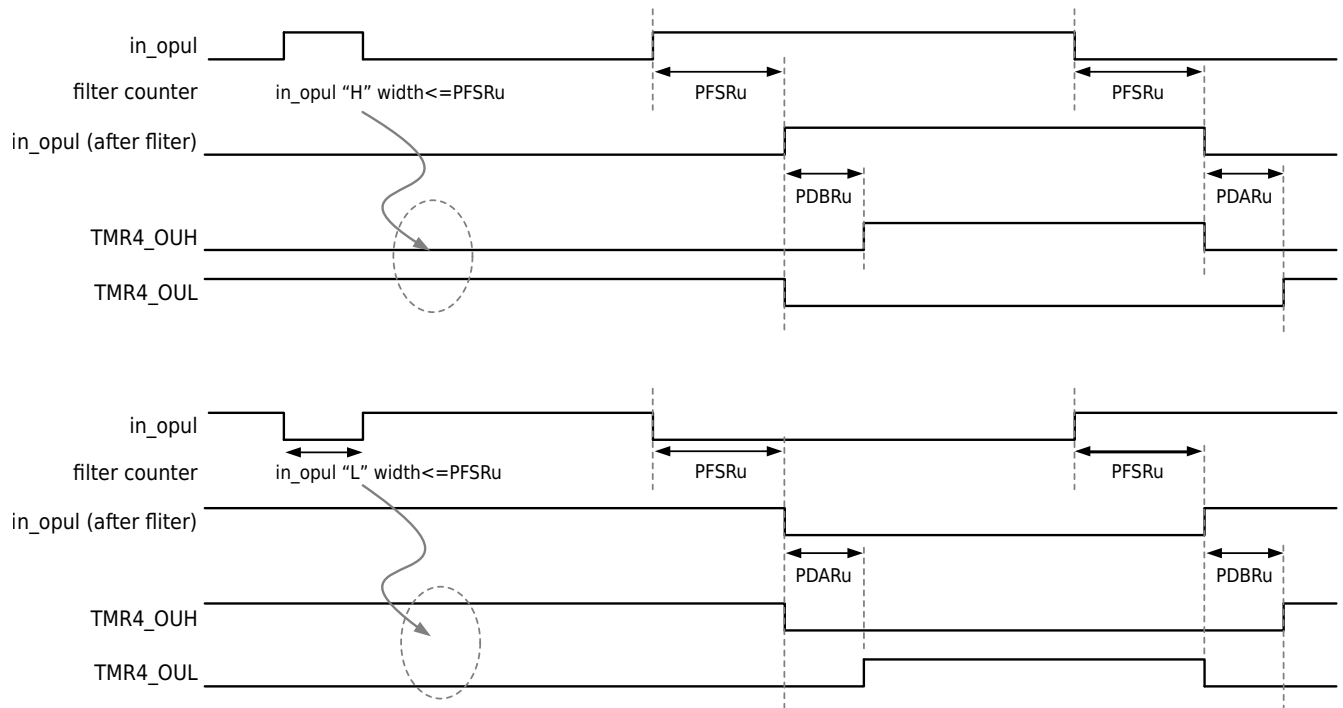


Figure 17-22 Complementary PWM Output in Dead-Zone Timer Filter Mode

**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 \(TMR4_PSCR\)](#)".

17.3.5 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.

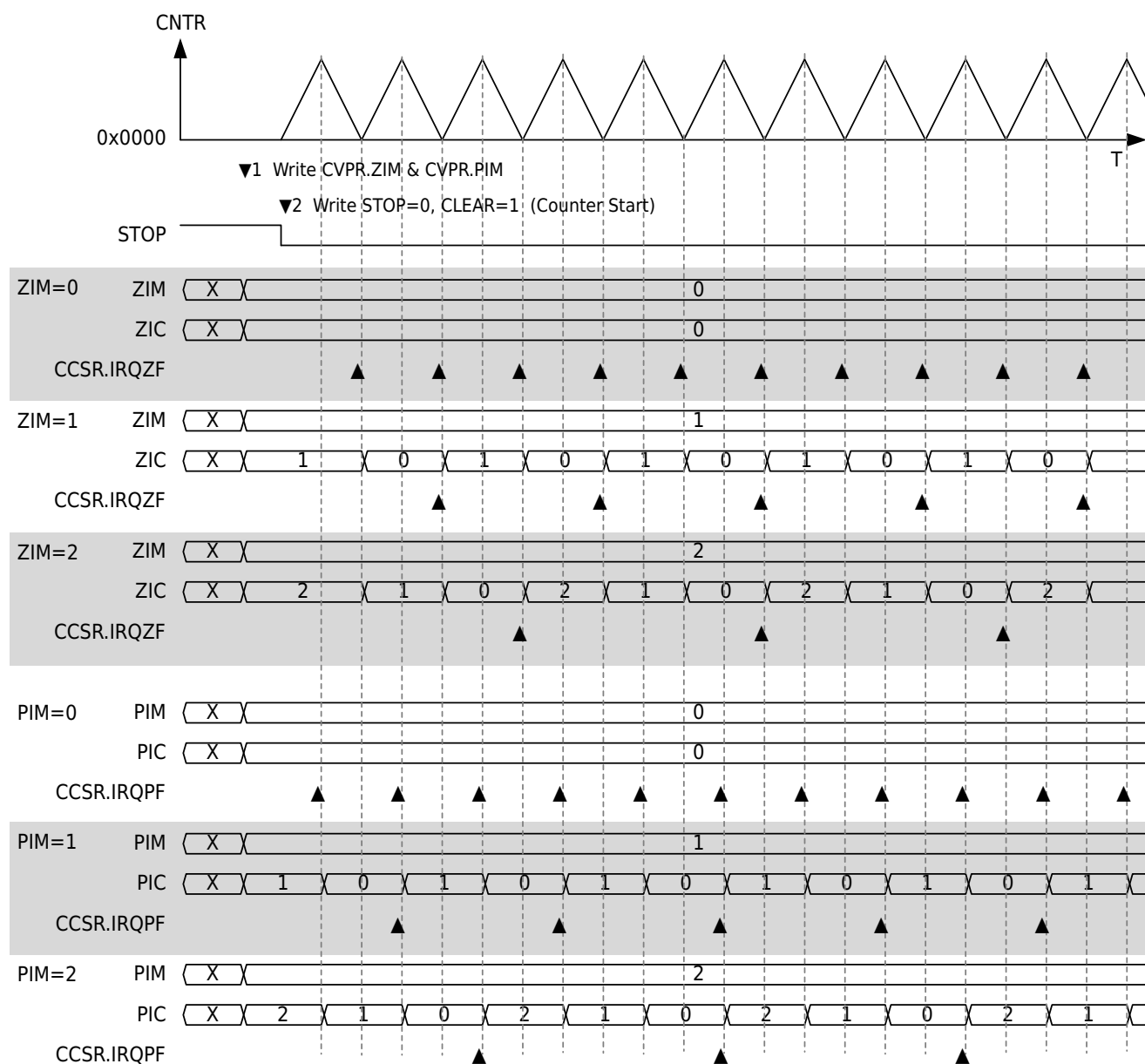


Figure 17-23 Periodic Interval Response Timing Diagram

- 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.

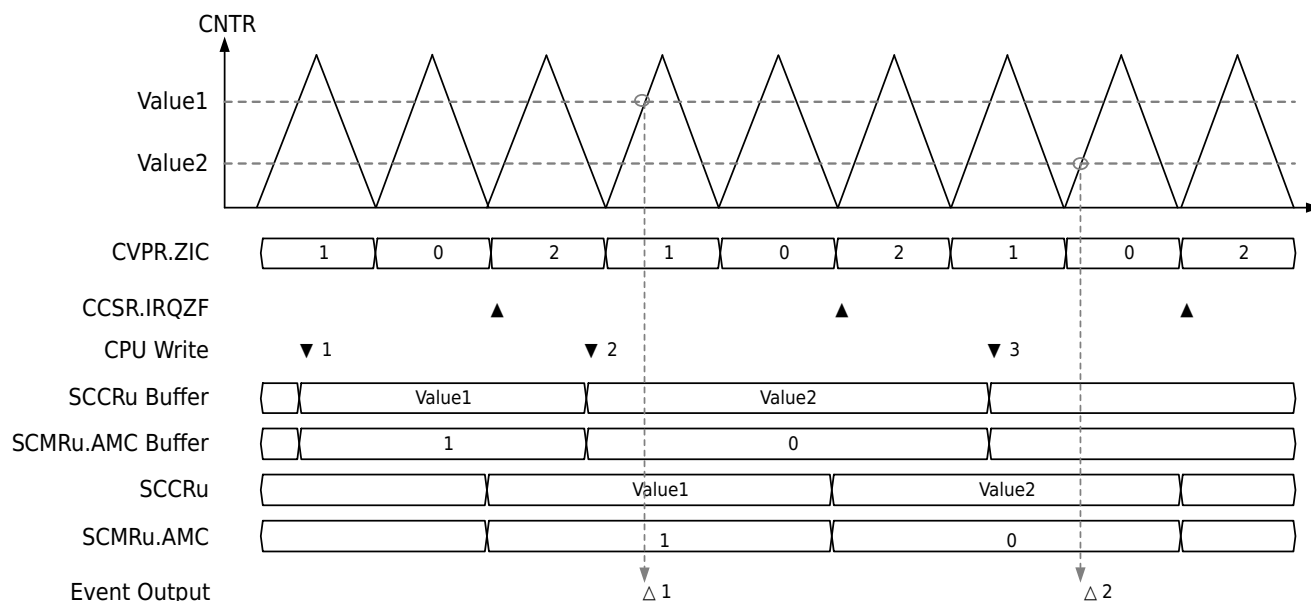


Figure 17-24 Special Event Output Signal Periodic Interval Response Output

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.

17.3.6 EMB Control

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 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_OUH port of Timer4, and the output on the TMR4_OUH port changes to a high impedance state.

After the EMB abnormal 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.

For detailed operation methods of EMB, please refer to the relevant description of the "Emergency Brake Module (EMB)" chapter.

17.3.7 Monitoring Output

Timer4 unit has a PWM period output monitoring port TMR4_PCT and a special event output monitoring port TMR4_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_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_PCT. The following figure is an example of the counting direction signal.

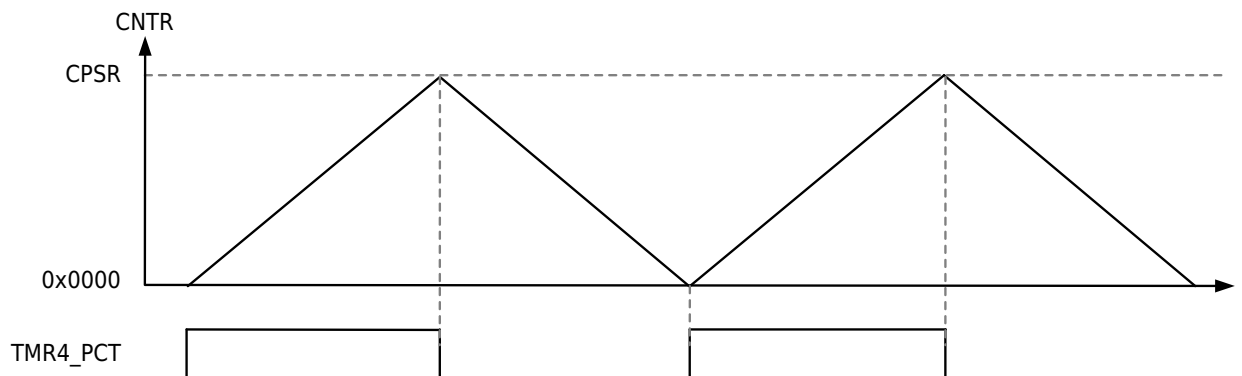


Figure 17-25 PWM Period Output Monitoring Example

The special event output monitoring port TMR4_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_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_ADSM will be reset (the output is low).

17.3.8 Hardware Startup

Timer4 can implement the counter startup function through internal hardware trigger events.

Timer4 has an internal hardware trigger source. The event source can be selected by setting the corresponding number in the hardware trigger event selection register. For details of the register content, please refer to the "Timer4 Hardware Trigger Event Selection Register (TMR4_TRGSEL)" section of "Automatic Operation System (AOS)". For specific event correspondence, please refer to the "Interrupt Controller (INTC)" section. When using the internal hardware trigger function, it is necessary to set the peripheral circuit trigger function enable bit of CMU_FCG to 1 first.

17.3.9 Synchronous Starting

Timer4 can be software-synchronized with TimerA for a simultaneous start. For specific usage, refer to the "Synchronous Starting" in the "General Timer (TimerA)" chapter.

17.3.10 Interrupt and Event

General Count Compare Match Interrupt and Event

There are a total of 6 OCCRM registers, 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 OCSRn register 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_GCMmn, m=U, V, W; n=H, L) will also be triggered.

The compare match event does not have a corresponding enable bit control. When the count compare matches, the compare match event is triggered. The event output (TMR4_GCMmn, m=U, V, W; n=H, L) is used to select triggering other modules.

Count Period Match Interrupt and Event

When the sawtooth wave counts up to the overflow point, the sawtooth wave counts down to the under-flow 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_GOVF and TMR4_GUDF) can be triggered at the corresponding time point.

The period match event does not have a corresponding enable bit control. When the count period matches, the period match event is triggered. The event output (TMR4_GOVF and TMR4_GUDF) is used to select triggering other modules.

Reload Count Match Interrupt and Event

When the reload function is enabled, a compare match signal is generated when PFSRn register matches the counter value. When a compare match occurs, 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_RLOm, where m=U, V, W) will be triggered.

The reload count match event does not have a corresponding enable bit control. When the reload count matches, the reload match event is triggered. The event output (TMR4_RLOm, m=U, V, W) is used to select triggering other modules.

Special Compare Match Interrupts and Events

There are a total of 6 SCCRm registers, which can be compared with the count value to generate a compare match valid signal. When a count compare match occurs, the SFmn (m=U, V, W; n=H, L) bits in the SCFR register are set to 1 respectively. If the SCIR.ITENmn (m=U, V, W; n=H, L) bits in the SCIR register are set to enable the interrupt, the corresponding interrupt request (TMR4_SCMmn, m=U, V, W; n=H, L) will also be triggered.

The 6 SCCRm registers 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 SCCRm register has a count comparison match event (TMR4_SCMmn, m=U, V, W; n=H, L), 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.

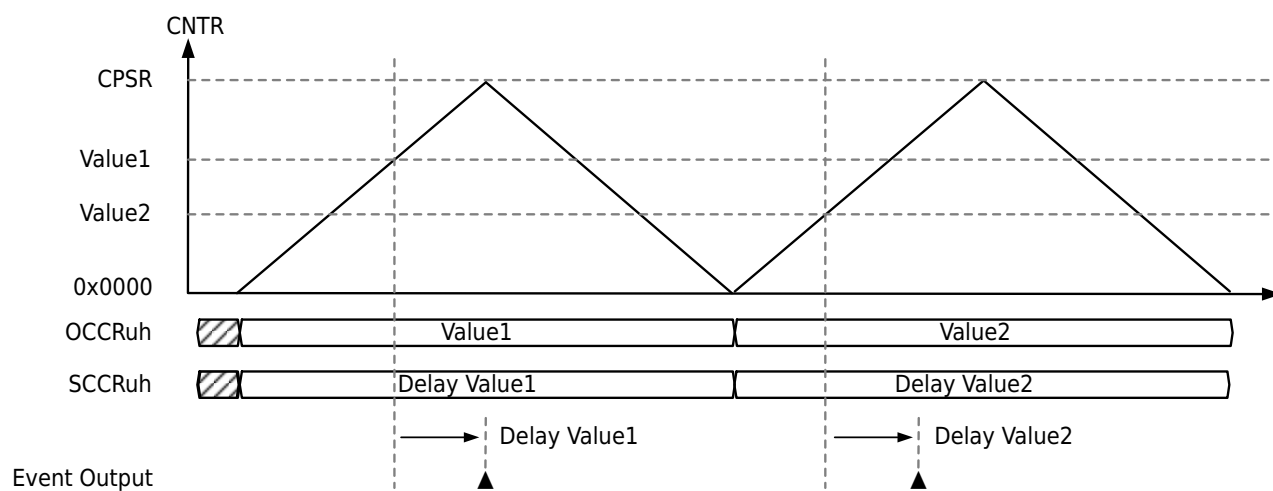


Figure 17-26 Output Timing of Special Event Output Signal in Delayed Start Mode



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.

17.4 Register Description

17.4.1 Register List

Base address: 0x40006800

Table 17-5 General Control Timer (Timer4) Register Offset

Offset	Register	Description	Width
0x0044	TMR4_CNTR	Count Value Register	16
0x0040	TMR4_CPSR	Period Value Register	16
0x0048	TMR4_CCSR	Control Status Register	16
0x004A	TMR4_CVPR	Valid Period Register	16
0x0000	TMR4_OCCRuh	General Compare Reference Register UH	16
0x0004	TMR4_OCCRuI	General Compare Reference Register UL	16
0x0008	TMR4_OCCRvh	General Compare Reference Register VH	16
0x000C	TMR4_OCCRvI	General Compare Reference Register VL	16
0x0010	TMR4_OCCRwh	General Compare Reference Register WH	16
0x0014	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
0x0038	TMR4_OCMRwI	General Mode Control Register WL	32
0x00B0	TMR4_SCCRuh	Special Compare Reference Register UH	16
0x00B4	TMR4_SCCRuI	Special Compare Reference Register UL	16

Offset	Register	Description	Width
0x00B8	TMR4_SCCRvh	Special Compare Reference Register VH	16
0x00BC	TMR4_SCCRvl	Special Compare Reference Register VL	16
0x00C0	TMR4_SCCRwh	Special Compare Reference Register WH	16
0x00C4	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_SCSRvl	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
0x00E0	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
0x004C	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
0x008C	TMR4_PDARv	PWM Dead Zone Control Register AV	16
0x008E	TMR4_PDBRv	PWM Dead Zone Control Register BV	16
0x0094	TMR4_PDARw	PWM Dead Zone Control Register AW	16

Offset	Register	Description	Width
0x0096	TMR4_PDBRw	PWM Dead Zone Control Register BW	16
0x00A4	TMR4_RCSR	Reload Control Status Register	
0x00E4	TMR4_SCIR	Special Interrupt Control Register	16
0x00E8	TMR4_SCFR	Special Flag Control 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_OUH, TMR4_OUL, TMR4_OVH, TMR4_OVL, TMR4_OWH, TMR4_OWL when the function is implemented; The registers indicated by n correspond to the output controls of ports TMR4_OUx, TMR4_OVx, TMR4_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.

17.4.2 Count Value Register (TMR4_CNTR)

Offset	Bit Position															
0x0044	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	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.  Notice Writing to this register is prohibited while counting is in progress.	RW

17.4.3 Period Value Register (TMR4_CPSR)

Offset	Bit Position															
0x0040	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	Counter period value  Note When reading data from this address area, the value read is not from the buffer register, but from the CPSR register.	RW

17.4.4 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	Reserved	HST	Reserved	IRQZF	IRQZEN	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	0b0: Internal HCLK clock 0b1: External TMR4_CLK port input clock  Note - This bit is set when the counter stops. - 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.	RW
14	Reserved	Reserved	-	-
13	HST	Hardware Start Enabled	Condition: The event specified in the TMR4_TRGSEL register occurs 0b0: Hardware start disabled upon condition match 0b1: When the condition matches, the hardware start is enabled, and the counter begins counting	RW
12	Reserved	Reserved	-	-
11	IRQZF	Underflow Status	0b0: No count underflow has occurred 0b1: Count underflow occurred  Note - 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. - When the counter is reset by the bus or when CLEAR=1 is written, the IRQZF bit will not be set.	RW
10	IRQZEN	Underflow Interrupt Enabled	0b0: Disable the IRQZF interrupt from being generated to the CPU 0b1: Enable the IRQZF interrupt from being generated to the CPU	RW

Bit	Name	Function	Description	Access
9	IRQPF	Overflow Status	● 0b0: No count overflow occurred ● 0b1: Count overflow occurred  Note ● 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. ● When the counter is reset by the bus or when CLEAR=1 is written, the IRQPF bit will not be set.	RW
8	IRQPEN	Overflow Interrupt Enabled	● 0b0: Disable the IRQPF interrupt from being generated to the CPU ● 0b1: Enable the IRQPF interrupt from being generated to the CPU	RW
7	BUFEN	Buffer Enabled	● 0b0: Disable CPSR buffer function ● 0b1: Enable CPSR buffer function	RW
6	STOP	Counter Enabled	● 0b0: Counter started ● 0b1: Counter stopped	RW
5	MODE	Waveform Mode	● 0b0: Sawtooth wave mode (supports only up counting) ● 0b1: Triangle wave mode	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	This bit indicates the count clock frequency division of the basic counter ● 0b0000: Count clock is HCLK ● 0b0001: Count clock is HCLK/2 ● 0b0010: Count clock is HCLK/4 ● 0b0011: Count clock is HCLK/8 ● 0b0100: Count clock is HCLK/16 ● 0b0101: Count clock is HCLK/32 ● 0b0110: Count clock is HCLK/64 ● 0b0111: Count clock is HCLK/128 ● 0b1000: Count clock is HCLK/256 ● 0b1001: Count clock is HCLK/512 ● 0b1010: Count clock is HCLK/1024 ● Others: Setting prohibited	RW

17.4.5 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	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. Write condition: counter stopped state (CCSR.STOP=1) The value of PIM is loaded into PIC if one of the following is met: 1. Counter clearing action occurs (CCSR.CLEAR writes 1) 2. PIC=0 and CNTR count overflow interrupt occurs	RW
11:8	ZIC	Underflow Interrupt Mask Status	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. Write condition: counter stopped state (CCSR.STOP=1) The value of ZIM is loaded into ZIC if one of the following is met: 1. Counter clearing action occurs (CCSR.CLEAR writes 1) 2. ZIC=0 and CNTR count underflow interrupt occurs	RW
7:4	PIM	Overflow Interrupt Mask Setting	Set the number of overflow interrupts to be masked The value of PIM is loaded into PIC if one of the following is met: 1. Counter clearing action occurs (CCSR.CLEAR writes 1) 2. PIC=0 and CNTR count overflow interrupt occurs	RW

Bit	Name	Function	Description	Access
3:0	ZIM	Underflow Interrupt Mask Setting	Set the number of underflow interrupts to be masked The value of ZIM is loaded into ZIC if one of the following is met: 1. Counter clearing action occurs (CCSR.CLEAR writes 1) 2. ZIC=0 and CNTR count underflow interrupt occurs	RW

17.4.6 General Compare Reference Register (TMR4_OCCRm)

Offset: 0x0000 (TMR4_OCCRUh), 0x0004 (TMR4_OCCRUl), 0x0008 (TMR4_OCCRVh), 0x000C (TMR4_OCCRVl), 0x0010 (TMR4_OCCRWh), 0x0014 (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	General compare reference  Note When reading data from this address area, the value read is not from the buffer register, but from the OCCR register.	RW

17.4.7 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	0b0: The counter value is not equal to the OCCR*L setting value 0b1: The counter value is equal to the OCCR*L setting value  Note This bit is enabled only when OCEL=1.	RW
6	OCFH	Count Match Flag H	0b0: The counter value is not equal to the OCCR*h setting value 0b1: The counter value is equal to the OCCR*h setting value  Note This bit is enabled only when OCEH=1.	RW
5	OCIEL	Count Match Interrupt Enable L	0b0: When OCFL is set, no interrupt occurs 0b1: When OCFL is set, an interrupt occurs	RW
4	OCIEH	Count Match Interrupt Enable H	0b0: When OCFH is set, no interrupt occurs 0b1: When OCFH is set, an interrupt occurs	RW
3	OCPL	Port Status L When Comparison Output Is Disabled	Write: 0b0: When OCEL=0, in_op*L outputs low level 0b1: When OCEL=0, in_op*L outputs high level Read: 0b0: The current state of in_op*L is low level 0b1: The current state of in_op*L is high level	RW
2	OCPL	Port Status H When Comparison Output Is Disabled	Write: 0b0: When OCEH=0, in_op*h outputs low level 0b1: When OCEH=0, in_op*h outputs high level Read: 0b0: The current state of in_op*h is low level 0b1: The current state of in_op*h is high level	RW

Bit	Name	Function	Description	Access
1	OCEL	Port Output Selection L	0b0: When compare output is disabled, the state of in_op*I port is determined by OCPL. 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.	RW
0	OCEH	Port Output Selection H	0b0: When compare output is disabled, the state of in_op*h port is determined by OCPH. 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.	RW

17.4.8 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	Additional Condition: Count overflow (CNTR=CPSR) and OCCR*I > CPSR. 0b0: Disable this condition as a matching condition. 0b1: Enable this condition as a matching condition. When the condition is met, OCCR*I will generate a count match.  Note - This bit can only be modified when the output comparison function is disabled (OCSR.OCEL=0). - For details, refer to the precautions in the TMR4_OCMRm register description.	RW
12	MCECH	Count Match Additional Condition Enable H	Additional Condition: Count overflow (CNTR=CPSR) and OCCR*h > CPSR. 0b0: Disable this condition as a matching condition. 0b1: Enable this condition as a matching condition. When the condition is met, OCCR*h will generate a count match.  Note - This bit can only be modified when the output comparison function is disabled (OCSR.OCEH=0). - For detailed information, refer to the notes in the TMR4_OCMRm register description.	RW

Bit	Name	Function	Description	Access
11	LMML	OCMR*I Periodic Interval Response Function Link L	0b0: The periodic interval response function link is disabled. The buffer transmission of OCMR*I is determined by the MLBUFEN setting. 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).	RW
10	LMMH	OCMR*h Periodic Interval Response Function Link H	0b0: The periodic interval response function link is disabled. The buffer transmission of OCMR*h is determined by the MLBUFEN setting. 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).	RW
9	LMCL	OCCR*I Periodic Interval Response Function Link L	0b0: The periodic interval response function link is disabled. The buffer transmission of OCMR*I is determined by the MLBUFEN setting. 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).	RW
8	LMCH	OCCR*h Periodic Interval Response Function Link H	0b0: The periodic interval response function link is disabled. The buffer transmission of OCCR*h is determined by the MLBUFEN setting. 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).	RW

Bit	Name	Function	Description	Access
7:6	MLBUFEN	OCMR*I Buffer Transmission	● 0b00: The value of the OCMR*I buffer register is directly written to OCMR*I ● 0b01: The value of the OCMR*I buffer register is written to OCMR*I when the counter underflows ● 0b10: The value of the OCMR*I buffer register is written to OCMR*I when the counter overflows ● 0b11: The value of the OCMR*I buffer register is written to OCMR*I when the counter either overflows or underflows	RW
5:4	MHBUFEN	OCMR*h Buffer Transmission	● 0b00: The value of the OCMR*h buffer register is directly written to OCMR*h ● 0b01: The value of the OCMR*h buffer register is written to OCMR*h when the counter underflows ● 0b10: The value of the OCMR*h buffer register is written to OCMR*h when the counter overflows ● 0b11: The value of the OCMR*h buffer register is written to OCMR*h when the counter either overflows or underflows	RW
3:2	CLBUFEN	OCCR*I Buffer Transmission	● 0b00: The value of the OCCR*I buffer register is directly written to OCCR*I. ● 0b01: The value of the OCCR*I buffer register is written to OCCR*I when the counter underflows. ● 0b10: The value of the OCCR*I buffer register is written to OCCR*I when the counter overflows. ● 0b11: The value of the OCCR*I buffer register is written to OCCR*I when the counter either overflows or underflows.	RW
1:0	CHBUFEN	OCCR*h Buffer Transmission	● 0b00: The value of the OCCR*h buffer register is directly written to OCCR*h ● 0b01: The value of the OCCR*h buffer register is written to OCCR*h when the counter underflows ● 0b10: The value of the OCCR*h buffer register is written to OCCR*h when the counter overflows ● 0b11: The value of the OCCR*h buffer register is written to OCCR*h when the counter either overflows or underflows	RW

17.4.9 General Mode 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 17-21 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 counts do not match. 0b00: When the condition is satisfied, in_op*h remains unchanged 0b01: When the condition is satisfied, in_op*h outputs high level 0b10: When the condition is satisfied, in_op*h outputs low level 0b11: When the condition is satisfied, in_op*h outputs inverted state	RW
13:12	OPNPKH	Overflow Condition 1 Port State H	Overflow condition 1 (Simultaneously satisfied): Count overflow and OCCR*h counts do not match. 0b00: When the condition is satisfied, in_op*h remains unchanged 0b01: When the condition is satisfied, in_op*h outputs high level 0b10: When the condition is satisfied, in_op*h outputs low level 0b11: When the condition is satisfied, in_op*h outputs inverted state	RW

Bit	Name	Function	Description	Access
11:10	OPZRH	Underflow Condition 2 Port State H	Underflow condition 2 (Simultaneously satisfied): Count underflow and OCCR*h counts do not match. 0b00: When the condition is satisfied, in_op*h remains unchanged 0b01: When the condition is satisfied, in_op*h outputs high level 0b10: When the condition is satisfied, in_op*h outputs low level 0b11: When the condition is satisfied, in_op*h outputs inverted state	RW
9:8	OPUCH	Up Counting Condition 2 Port State H	Up counting condition 2 (Simultaneously satisfied): the counter counts up, and the OCCR*h counts match 0b00: When the condition is satisfied, in_op*h remains unchanged 0b01: When the condition is satisfied, in_op*h outputs high level 0b10: When the condition is satisfied, in_op*h outputs low level 0b11: When the condition is satisfied, in_op*h outputs inverted state	RW
7:6	OPPKH	Overflow Condition 2 Port State H	Overflow Condition 2 (Simultaneously satisfied): Count overflow and OCCR*h counts match 0b00: When the condition is satisfied, in_op*h remains unchanged 0b01: When the condition is satisfied, in_op*h outputs high level 0b10: When the condition is satisfied, in_op*h outputs low level 0b11: When the condition is satisfied, in_op*h outputs inverted state	RW
5:4	OPDCH	Down Counting Condition 2 Port State H	Down counting condition 2 (Simultaneously satisfied): the counter counts down, and the OCCR*h counts match 0b00: When the condition is satisfied, in_op*h remains unchanged 0b01: When the condition is satisfied, in_op*h outputs high level 0b10: When the condition is satisfied, in_op*h outputs low level 0b11: When the condition is satisfied, in_op*h outputs inverted state	RW
3	OCFZRH	Underflow Point OCFH Status H	Condition (Simultaneously satisfied): count underflow, OCCR*h counts match 0b0: When the condition is met, the OCSR.OCFH bit remains unchanged 0b1: When the condition is met, the OCSR.OCFH bit is set	RW

Bit	Name	Function	Description	Access
2	OCFUCH	Up Counting OCFH State H	Conditions (Simultaneously satisfied): counter counts up, OCCR*h counts match ● 0b0: When the condition is met, the OCSR.OCFH bit remains unchanged ● 0b1: When the condition is met, the OCSR.OCFH bit is set	RW
1	OCFPKH	Overflow Point OCFH State H	Condition (Simultaneously satisfied): count overflow, OCCR*h counts match ● 0b0: When the condition is met, the OCSR.OCFH bit remains unchanged ● 0b1: When the condition is met, the OCSR.OCFH bit is set	RW
0	OCFDCH	Down Counting OCFH State H	Conditions (Simultaneously satisfied): counter counts down, OCCR*h counts match ● 0b0: When the condition is met, the OCSR.OCFH bit remains unchanged ● 0b1: When the condition is met, the OCSR.OCFH bit is set	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 17-24 OCCR*h/l 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 counts do not match, OCCR*h counts match 0b00: When the condition is satisfied, in_op*l remains unchanged 0b01: When the condition is satisfied, in_op*l outputs high level 0b10: When the condition is satisfied, in_op*l outputs low level 0b11: When the condition is satisfied the output of in_op*l is reversed	RW
29:28	EOPNPKL	Overflow Condition 3 Port State L	Overflow condition 3 (Simultaneously satisfied): Counting overflow, OCCR*l counts do not match, OCCR*h counts match 0b00: When the condition is satisfied, in_op*l remains unchanged 0b01: When the condition is satisfied, in_op*l outputs high level 0b10: When the condition is satisfied, in_op*l outputs low level 0b11: When the condition is satisfied the output of in_op*l is reversed	RW

Bit	Name	Function	Description	Access
27:26	EOPZRL	Underflow Condition 4 Port State L	Underflow condition 4 (Simultaneously satisfied): Counting underflow, OCCR*I counts match, OCCR*h counts match ● 0b00: When the condition is satisfied, in_op*I remains unchanged ● 0b01: When the condition is satisfied, in_op*I outputs high level ● 0b10: When the condition is satisfied, in_op*I outputs low level ● 0b11: When the condition is satisfied the output of in_op*I is reversed	RW
25:24	EOPUCL	Up counting Condition 4 Port State L	Up counting condition 4 (Simultaneously satisfied): counter counts up, OCCR*I counts match, OCCR*h counts match ● 0b00: When the condition is satisfied, in_op*I remains unchanged ● 0b01: When the condition is satisfied, in_op*I outputs high level ● 0b10: When the condition is satisfied, in_op*I outputs low level ● 0b11: When the condition is satisfied the output of in_op*I is reversed	RW
23:22	EOPPKL	Overflow Condition 4 Port State L	Overflow condition 4 (Simultaneously satisfied): count overflow, OCCR*I counts match, OCCR*h counts match ● 0b00: When the condition is satisfied, in_op*I remains unchanged ● 0b01: When the condition is satisfied, in_op*I outputs high level ● 0b10: When the condition is satisfied, in_op*I outputs low level ● 0b11: When the condition is satisfied the output of in_op*I is reversed	RW
21:20	EOPDCL	Down Counting Condition 4 Port State L	Down counting condition 4 (Simultaneously satisfied): counter counts down, OCCR*I counts match, OCCR*h counts match ● 0b00: When the condition is satisfied, in_op*I remains unchanged ● 0b01: When the condition is satisfied, in_op*I outputs high level ● 0b10: When the condition is satisfied, in_op*I outputs low level ● 0b11: When the condition is satisfied the output of in_op*I is reversed	RW

Bit	Name	Function	Description	Access
19:18	EOPNUCL	Up Counting Condition 3 Port State L	Up counting condition 3 (Simultaneously satisfied): the counter counts up, OCCR*I counts do not match, OCCR*h counts match ● 0b00: When the condition is satisfied, in_op*I remains unchanged ● 0b01: When the condition is satisfied, in_op*I outputs high level ● 0b10: When the condition is satisfied, in_op*I outputs low level ● 0b11: When the condition is satisfied the output of in_op*I is reversed	RW
17:16	EOPNDCL	Down Counting Condition 3 Port State L	Down counting condition 3 (Simultaneously satisfied): counter counts down, OCCR*I counts do not match, OCCR*h counts match ● 0b00: When the condition is satisfied, in_op*I remains unchanged ● 0b01: When the condition is satisfied, in_op*I outputs high level ● 0b10: When the condition is satisfied, in_op*I outputs low level ● 0b11: When the condition is satisfied the output of in_op*I is reversed	RW
15:14	OPNZRL	Underflow Condition 1 Port State L	Underflow condition 1 (Simultaneously satisfied): Counting underflow, OCCR*I counts do not match, OCCR*h counts match, ● 0b00: When the condition is satisfied, in_op*I remains unchanged ● 0b01: When the condition is satisfied, in_op*I outputs high level ● 0b10: When the condition is satisfied, in_op*I outputs low level ● 0b11: When the condition is satisfied the output of in_op*I is reversed	RW
13:12	OPNPKL	Overflow Condition 1 Port State L	Overflow condition 1 (Simultaneously satisfied): count overflow, OCCR*I counts match, OCCR*h counts do not match ● 0b00: When the condition is satisfied, in_op*I remains unchanged ● 0b01: When the condition is satisfied, in_op*I outputs high level ● 0b10: When the condition is satisfied, in_op*I outputs low level ● 0b11: When the condition is satisfied the output of in_op*I is reversed	RW

Bit	Name	Function	Description	Access
11:10	OPZRL	Underflow Condition 2 Port State L	Underflow condition 2 (Simultaneously satisfied): Counting underflow, OCCR*I counts match, OCCR*h counts do not match 0b00: When the condition is satisfied, in_op*I remains unchanged 0b01: When the condition is satisfied, in_op*I outputs high level 0b10: When the condition is satisfied, in_op*I outputs low level 0b11: When the condition is satisfied the output of in_op*I is reversed	RW
9:8	OPUCL	Up counting Condition 2 Port State L	Up counting condition 2 (Simultaneously satisfied): The counter counts up, OCCR*I counts match, OCCR*h counts do not match 0b00: When the condition is satisfied, in_op*I remains unchanged 0b01: When the condition is satisfied, in_op*I outputs high level 0b10: When the condition is satisfied, in_op*I outputs low level 0b11: When the condition is satisfied the output of in_op*I is reversed	RW
7:6	OPPKL	Overflow Condition 2 Port State L	Overflow condition 2 (Simultaneously satisfied): Count overflow, OCCR*I counts match, OCCR*h counts do not match 0b00: When the condition is satisfied, in_op*I remains unchanged 0b01: When the condition is satisfied, in_op*I outputs high level 0b10: When the condition is satisfied, in_op*I outputs low level 0b11: When the condition is satisfied the output of in_op*I is reversed	RW
5:4	OPDCL	Down Counting Condition 2 Port State L	Down counting condition 2 (Simultaneously satisfied): counter counts down, OCCR*I counts match, OCCR*h counts do not match 0b00: When the condition is satisfied, in_op*I remains unchanged 0b01: When the condition is satisfied, in_op*I outputs high level 0b10: When the condition is satisfied, in_op*I outputs low level 0b11: When the condition is satisfied the output of in_op*I is reversed	RW
3	OCFZRL	Underflow Point OCFL State L	Condition (Simultaneously satisfied): count underflow, OCCR*I counts match 0b0: When the condition is met, the OCSR.OCFL bit remains unchanged 0b1: When the condition is met, the OCSR.OCFL bit is set	RW

Bit	Name	Function	Description	Access
2	OCFUCL	Up Counting OCFL State L	Conditions (Simultaneously satisfied): counter counts up, OCCR*I counts match 0b0: When the condition is met, the OCSR.OCFL bit remains unchanged 0b1: When the condition is met, the OCSR.OCFL bit is set	RW
1	OCFPKL	Overflow Point OCFL State L	Conditions (Simultaneously satisfied): count over-flow, OCCR*I counts match 0b0: When the condition is met, the OCSR.OCFL bit remains unchanged 0b1: When the condition is met, the OCSR.OCFL bit is set	RW
0	OCFDCL	Down Counting OCFL State L	Conditions (Simultaneously satisfied): counter counts down, OCCR*I counts match 0b0: When the condition is met, the OCSR.OCFL bit remains unchanged 0b1: When the condition is met, the OCSR.OCFL bit is set	RW

**Notice**

- When reading data from this address area, the value read is not from the buffer register, but from the OCMR register.
- TMR4_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

17.4.10 Special Compare Reference Register (TMR4_SCCRm)

Offset: 0x00B0 (TMR4_SCCRuh), 0x00B4 (TMR4_SCCRul), 0x00B8 (TMR4_SCCRvh), 0x00BC (TMR4_SCCRvl), 0x00C0 (TMR4_SCCRwh), 0x00C4 (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)  Note When reading data from this address area, the value read is not from the buffer register, but from the SCCR register.	RW

17.4.11 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	0b0: When the count underflows, EVT has no operation 0b1: When the count underflows: - When EVTMS=0&SCCR compare match&SCMR setting match, EVT starts output - When EVTMS=1&OCCR compare match&SCMR setting match, EVT delay mode starts	RW
14	UEN	Up Counting EVT Enabled	0b0: When counting up, EVT has no operation 0b1: When counting up: - When EVTMS=0&SCCR compare match&SCMR setting match, EVT starts output - When EVTMS=1&OCCR compare match&SCMR setting match, EVT delay mode starts	RW
13	PEN	Overflow Point EVT Enabled	0b0: When the count overflows, EVT has no operation 0b1: When the count overflows: - When EVTMS=0&SCCR compare match&SCMR setting match, EVT starts output - When EVTMS=1&OCCR compare match&SCMR setting match, EVT delay mode starts	RW
12	DEN	Down Counting EVT Enabled	0b0: When counting down, EVT has no operation 0b1: When counting down: - When EVTMS=0&SCCR compare match&SCMR setting match, EVT starts output - When EVTMS=1&OCCR compare match&SCMR setting match, EVT delay mode starts	RW

Bit	Name	Function	Description	Access
11:10	Reserved	Reserved	-	-
9	EVTDS	EVT Delay Object Selection	0b0: In delayed start mode, OCCR*h is used as the delayed comparison match object 0b1: In delayed start mode, OCCR*l is used as the delayed comparison match object  Note This bit is invalid when EVTMS=0.	RW
8	EVTMS	EVT Mode Selection	0b0: Compare start mode (triggered by the comparison result of CNTR and SCCR) 0b1: Delay start mode (triggered after the comparison match event is delayed by SCCR)	RW
7:6	Reserved	Reserved	-	-
5	LMC	Periodic Interval Response Function Link	0b0: Periodic interval response function link is disabled, and the buffer transfer of SCCR is determined by the BUFEN setting 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	RW
4:2	EVTOS	EVT Output Selection	0b000: EVT output of Special Event 0 is enabled 0b001: EVT output of Special Event 1 is enabled 0b010: EVT output of Special Event 2 is enabled 0b011: EVT output of Special Event 3 is enabled 0b100: EVT output of Special Event 4 is enabled 0b101: EVT output of Special Event 5 is enabled - Others: Setting prohibited	RW
1:0	BUFEN	SCCR&SCMR Buffer Transmission	0b00: The values of the SCCR and SCMR buffer registers are directly written to SCCR and SCMR 0b01: The values of the SCCR and SCMR buffer registers are written to SCCR and SCMR when the count underflows 0b10: The values of the SCCR and SCMR buffer registers are written to SCCR and SCMR when the count overflows 0b11: The values of the SCCR and SCMR buffer registers are written to SCCR and SCMR when the count underflows or overflows	RW

17.4.12 Special Extended Control Register (TMR4_SCER)

Offset	Bit Position															
0x00E0	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 0b0: Disable the count direction signal output to the TMR4_PCT port 0b1: Enable the count direction signal output to the TMR4_PCT port	RW
2:0	EVTRS	Special Event Output Selection	0b000: Disable special event output to port 0b001: The EVT of Special Event 0 is output to TMR4_ADSM port 0b010: The EVT of Special Event 1 is output to the TMR4_ADSM port 0b011: The EVT of Special Event 2 is output to the TMR4_ADSM port 0b100: The EVT of Special Event 3 is output to the TMR4_ADSM port 0b101: The EVT of Special Event 4 is output to the TMR4_ADSM port 0b110: The EVT of Special Event 5 is output to the TMR4_ADSM port 0b111: Setting prohibited	RW

17.4.13 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	0b0: Disable AMC and CVPR.PIC comparison 0b1: Enable AMC and CVPR.PIC comparison	RW
6	MZCE	Periodic Interval Response Enabled	0b0: Disable comparison between AMC and CVPR.ZIC 0b1: Enable comparison between AMC and CVPR.ZIC	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.

17.4.14 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	0b00: Neither TMR4_O*H nor TMR4_O*L outputs are inverted 0b01: The outputs of TMR4_O*H and TMR4_O*L are both inverted 0b10: The output of TMR4_O*H is inverted, and the output of TMR4_O*L is not inverted 0b11: The output of TMR4_O*H is not inverted, and the output of TMR4_O*L is inverted	RW
5:4	PWMMD	PWM Output Mode	0b00: Pass-through mode 0b01: Dead-zone timer mode 0b10: Dead-zone timer filter mode 0b11: Setting prohibited	RW
3	Reserved	Reserved	-	-
2:0	DIVCK	Count Clock Division Frequency	This bit indicates the count clock division of the filter counter, dead zone counter and reload counter. 0b000: Count clock is HCLK 0b001: Count clock is HCLK/2 0b010: Count clock is HCLK/4 0b011: Count clock is HCLK/8 0b100: Count clock is HCLK/16 0b101: Count clock is HCLK/32 0b110: Count clock is HCLK/64 0b111: Count clock is HCLK/128	RW

17.4.15 PWM Status Control Register (TMR4_PSCR)

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	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_OWL Output Disable Status	Conditions: 1. EMB event occurs; 2. Software writes MOE=0; 3. MOE=1&OEWL=0 0b00: When one of the above conditions is met, the TMR4_OWL port outputs normally 0b01: When one of the above conditions is met, the TMR4_OWL port output is Hi-Z 0b10: When one of the above conditions is met, the TMR4_OWL port output is fixed at a low level 0b11: When one of the above conditions is met, the TMR4_OWL port output is fixed at a high level  Note This bit can only be modified when Timer4 is stopped (CCSR.STOP=1).	RW
25:24	OSWH	TMR4_OWH Output Disable Status	Conditions: 1. EMB event occurs; 2. Software writes MOE=0; 3. MOE=1&OEWH=0 0b00: When one of the above conditions is met, the TMR4_OWH port outputs normally 0b01: When one of the above conditions is met, the TMR4_OWH port output is Hi-Z 0b10: When one of the above conditions is met, the TMR4_OWH port output is fixed at a low level 0b11: When one of the above conditions is met, the TMR4_OWH port output is fixed at a high level  Note This bit can only be modified when Timer4 is stopped (CCSR.STOP=1).	RW
23:22	OSVL	TMR4_OVL Output Disable Status	Conditions: 1. EMB event occurs; 2. Software writes MOE=0; 3. MOE=1&OEVL=0 0b00: When one of the above conditions is met, the TMR4_OVL port outputs normally 0b01: When one of the above conditions is met, the TMR4_OVL port output is Hi-Z 0b10: When one of the above conditions is met, the TMR4_OVL port output is fixed at a low level	RW

Bit	Name	Function	Description	Access
23:22	OSVL	TMR4_OVL Output Disa- ble Status	0b11: When one of the above conditions is met, the TMR4_OVL port output is fixed at a high level  Note This bit can only be modified when Timer4 is stopped (CCSR.STOP=1).	RW
21:20	OSVH	TMR4_OVH Output Disa- ble Status	Conditions: 1. EMB event occurs; 2. Software writes MOE=0; 3. MOE=1&OEVL=0 0b00: When one of the above conditions is met, the TMR4_OVH port outputs normally 0b01: When one of the above conditions is met, the TMR4_OVH port output is Hi-Z 0b10: When one of the above conditions is met, the TMR4_OVH port output is fixed at a low level 0b11: When one of the above conditions is met, the TMR4_OVH port output is fixed at a high level  Note This bit can only be modified when Timer4 is stopped (CCSR.STOP=1).	RW
19:18	OSUL	TMR4_OUL Output Disa- ble Status	Conditions: 1. EMB event occurs; 2. Software writes MOE=0; 3. MOE=1&OEVL=0 0b00: When one of the above conditions is met, the TMR4_OUL port outputs normally 0b01: When one of the above conditions is met, the TMR4_OUL port output is Hi-Z 0b10: When one of the above conditions is met, the TMR4_OUL port output is fixed at a low level 0b11: When one of the above conditions is met, the TMR4_OUL port output is fixed at a high level  Note This bit can only be modified when Timer4 is stopped (CCSR.STOP=1).	RW
17:16	OSUH	TMR4_OUH Output Disa- ble Status	Conditions: 1. EMB event occurs; 2. Software writes MOE=0; 3. MOE=1&OEVL=0 0b00: When one of the above conditions is met, the TMR4_OUH port outputs normally 0b01: When one of the above conditions is met, the TMR4_OUH port output is Hi-Z 0b10: When one of the above conditions is met, the TMR4_OUH port output is fixed at a low level 0b11: When one of the above conditions is met, the TMR4_OUH port output is fixed at a high level  Note This bit can only be modified when Timer4 is stopped (CCSR.STOP=1).	RW
15:10	Reserved	Reserved	-	-

Bit	Name	Function	Description	Access
9	AOE	Automatic Output Enable	0b0: After the EMB event is cleared, the MOE bit can only be set to 1 by software to restore normal PWM output 0b1: After the EMB event is cleared, the MOE bit is automatically set to 1 by hardware, and the PWM output returns to normal  Note Setting EMB_STATCLR or clearing EMB_SOE can clear the EMB event.	RW
8	MOE	Master Output Enable	0b0: TMR4_O*H/L port outputs the state specified by the OS*H/L bit 0b1: TMR4_O*H, TMR4_O*L port outputs are enabled when the OE*H, OE*L bits are set to enable  Note 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.	RW
7:6	ODT	Port Enable Bit Active Time	Condition: After the value of port enable bit OE*L, OE*H is changed 0b0x: When the condition is met, it takes effect immediately 0b10: When the condition is met, it takes effect when the counter underflows next time 0b11: When the condition is met, it takes effect when the counter overflows next time	RW
5	OEWL	TMR4_OWL Output Enabled	0b0: TMR4_OWL port output state is set by the OSWL bit 0b1: When MOE=1, TMR4_OWL port outputs normal PWM waveform	RW
4	OEWH	TMR4_OWH Output Enabled	0b0: TMR4_OWH port output state is set by the OSWH bit 0b1: When MOE=1, TMR4_OWH port outputs normal PWM waveform	RW
3	OEVL	TMR4_OVL Output Enabled	0b0: TMR4_OVL port output state is set by the OSVL bit 0b1: When MOE=1, TMR4_OVL port outputs normal PWM waveform	RW
2	OEVL	TMR4_OVL Output Enabled	0b0: TMR4_OVL port output state is set by the OSVL bit 0b1: When MOE=1, TMR4_OVL port outputs normal PWM waveform	RW
2	OEVL	TMR4_OVL Output Enabled	0b0: TMR4_OVL port output state is set by the OSVL bit 0b1: When MOE=1, TMR4_OVL port outputs normal PWM waveform	RW

Bit	Name	Function	Description	Access
1	OEUL	TMR4_OUL Output Enabled	0b0: TMR4_OUL port output state is set by the OSUL bit 0b1: When MOE=1, TMR4_OUL port outputs normal PWM waveform	RW
0	OEUH	TMR4_OUH Output Enabled	0b0: TMR4_OUH port output state is set by the OSUH bit 0b1: When MOE=1, TMR4_OUH port outputs normal PWM waveform	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_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

17.4.16 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  Note 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.	RW

17.4.17 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

17.4.18 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

17.4.19 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. Read: 0b0: Reload counter W stopped 0b1: Reload counter W started	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	0b0: The reload counter count value does not match 0 0b1: The reload counter count value matches 0	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. Read: 0b0: Reload counter V stopped 0b1: Reload counter V started	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	Count Match Status V	0b0: The reload counter count value does not match 0 0b1: The reload counter count value matches 0	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. Read: 0b0: Reload counter U stopped 0b1: Reload counter U started	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	Count Match Status U	0b0: The reload counter count value does not match 0 0b1: The reload counter count value matches 0	RO
3	<i>Reserved</i>	<i>Reserved</i>	-	-
2	RTIDW	Reload Interrupt Mask W	0b0: Reload interrupt W output is enabled when the reload function is active. 0b1: Reload interrupt W output is disabled when the reload function is active.	RW
1	RTIDV	Reload Interrupt Mask V	0b0: Reload interrupt V output is enabled when the reload function is active. 0b1: Reload interrupt V output is disabled when the reload function is active.	RW
0	RTIDU	Reload Interrupt Mask U	0b0: Reload interrupt U output is enabled when the reload function is active. 0b1: Reload interrupt U output is disabled when the reload function is active.	RW

17.4.20 Special Interrupt Control Register (TMR4_SCIR)

Offset	Bit Position															
0x00E4	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0xFF00															
Name	Reserved										ITEN5	ITEN4	ITEN3	ITEN2	ITEN1	ITEN0
Access											RW	RW	RW	RW	RW	RW

Bit	Name	Function	Description	Access
15:6	Reserved	Reserved	-	-
5	ITEN5	Special Match Interrupt 5 Enabled	0b0: Disable 0b1: Enable	RW
4	ITEN4	Special Match Interrupt 4 Enabled	0b0: Disable 0b1: Enable	RW
3	ITEN3	Special Match Interrupt 3 Enabled	0b0: Disable 0b1: Enable	RW
2	ITEN2	Special Match Interrupt 2 Enabled	0b0: Disable 0b1: Enable	RW
1	ITEN1	Special Match Interrupt 1 Enabled	0b0: Disable 0b1: Enable	RW
0	ITEN0	Special Match Interrupt 0 Enabled	0b0: Disable 0b1: Enable	RW

17.4.21 Special Flag Control Register (TMR4_SCFR)

Offset	Bit Position															
0x00E8	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0xFF00															
Name	Reserved										SF5	SF4	SF3	SF2	SF1	SF0
Access											RW	RW	RW	RW	RW	RW

Bit	Name	Function	Description	Access
15:6	Reserved	Reserved	-	-
5	SF5	Special Compare Match Flag Bit 5	0b0: No occurrence of special count compare match event 5 0b1: Special count compare match event 5 occurred	RW
4	SF4	Special Compare Match Flag Bit 4	0b0: No occurrence of special count compare match event 4 0b1: Special count compare match event 4 occurred	RW
3	SF3	Special Compare Match Flag Bit 3	0b0: No occurrence of special count compare match event 3 0b1: Special count compare match event 3 occurred	RW
2	SF2	Special Compare Match Flag Bit 2	0b0: No occurrence of special count compare match event 2 0b1: Special count compare match event 2 occurred	RW
1	SF1	Special Compare Match Flag Bit 1	0b0: No occurrence of special count compare match event 1 0b1: Special count compare match event 1 occurred	RW
0	SF0	Special Compare Match Flag Bit 0	0b0: No occurrence of special count compare match event 0 0b1: Special count compare match event 0 occurred	RW

18 Emergency Brake Module (EMB)

18.1 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

18.2 Functional Description

18.2.1 EMB Channel Function Block Diagram

The EMB is used to output a control event signal to the timer module with PWM function (Timer4) when certain conditions are met, notifying the timer module to disable the current PWM output.

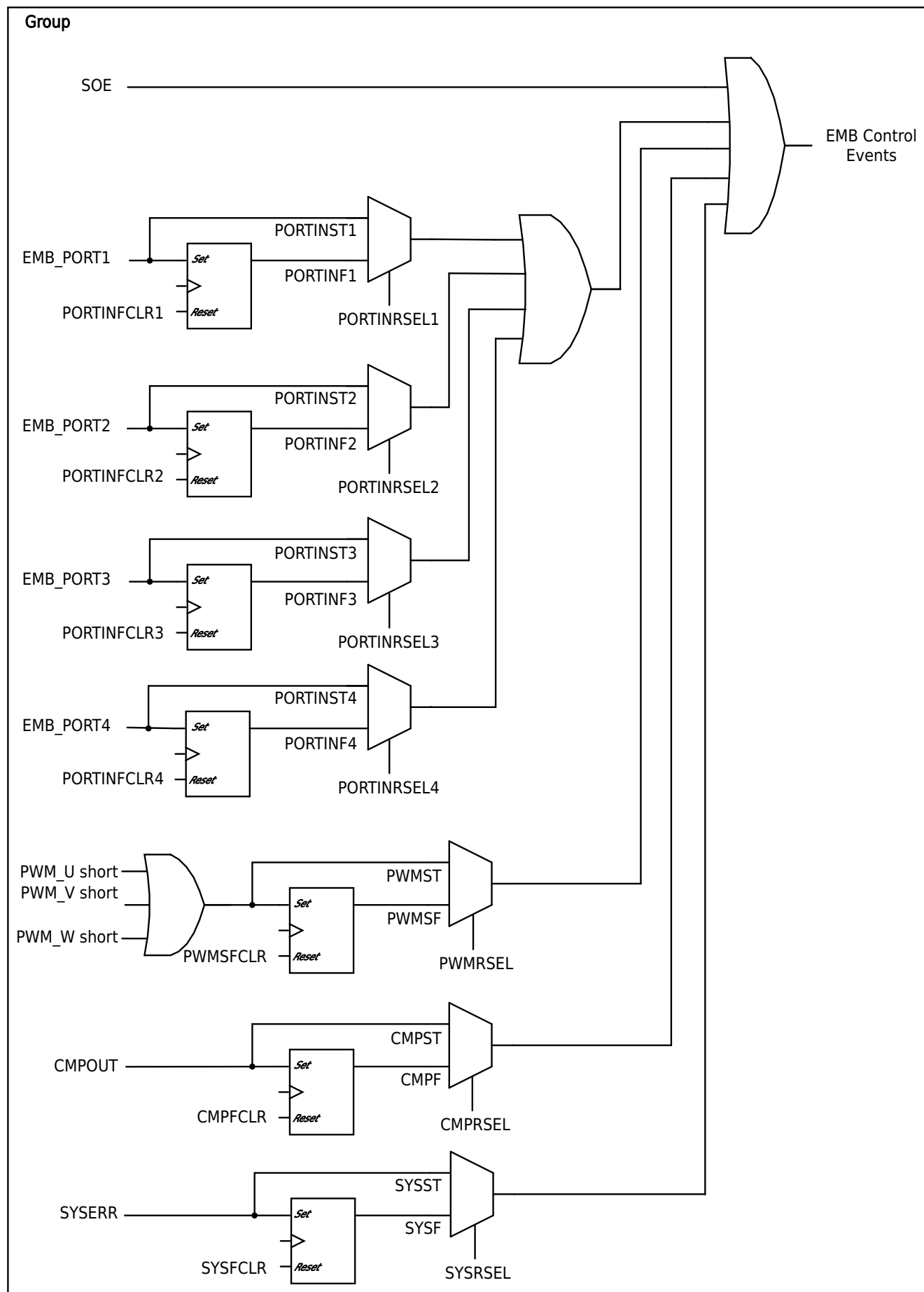


Figure 18-1 EMB Channel Function Block Diagram

Table 18-1 System Error Sources

System Error	Description
SYSERR	Cortex-M0+ Lockup
	LVD1/2 Detection
	SRAM ECC Check Error
	External Oscillator Stop
	D/PFlash ECC Check Error
	FCM Frequency Measurement Anomaly

18.2.2 EMB Structure Block Diagram

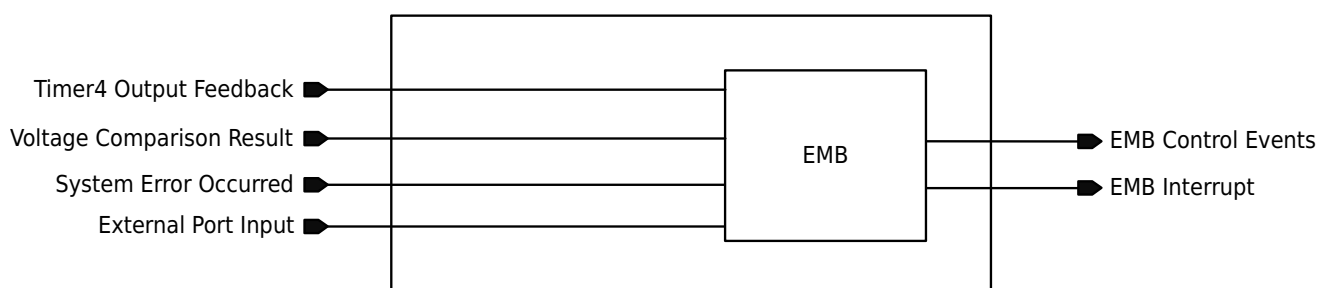


Figure 18-2 EMB Structure Block Diagram

18.2.3 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 18-2 EMB Port Assignment

Port	PIN Name
EMB_PORT0	PC8, PE3
EMB_PORT1	PA6, PC9, PE1
EMB_PORT2	PA7, PC16, PE0
EMB_PORT3	PC17, PD4

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 Timer4. After receiving the control event, 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.

18.2.4 Stop PWM Signal Output When PWM Output Port Level Occurs In-phase (same high or same low)

First, enable the function by setting the enable bit, specifically set the `EMB_CTL1.PWMSENy` ($y=0-2$) to valid. Simultaneously, select the active level to be monitored via the `EMB_CTL2.PWMLVy` ($y=0-2$).

After the settings are completed, EMB monitors the complementary PWM output signals of 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 Timer4 via `EMB_STAT.PWMST` or `EMB_STAT.PWMSF`. After receiving the control event, 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.

The EMB can be used to monitor the complementary PWM output signals of Timer4.

Port Name	Function	EMB_CTL1 Control Bit	EMB_CTL2 Control Bit
TMR4_OUH	Complementary PWM output signal of Timer4	PWMSEN2	PWMLV2
TMR4_OUL			
TMR4_OVH		PWMSEN1	PWMLV1
TMR4_OVL			
TMR4_OWH		PWMSEN0	PWMLV0
TMR4_OWL			

18.2.5 Stop PWM Signal Output According to Voltage Comparator Compare Result

Group of EMB can send control event signals to Timer4 according to the 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.CMPEN 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.CMPINTEN=1.

The control event of EMB selects register EMB_RLSSEL.CMPRSEL through the EMB release. The selection is output to Timer4 via EMB_STAT.CMPST or EMB_STAT.CMPF. After receiving the control event signal, 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.

18.2.6 Stop PWM Signal Output when System Error Occurs

The EMB group can send a notification signal to Timer4 when a system error occurs. The system error signals come from LVD1/2 detection, Cortex-M0+ lock signal, SRAM ECC check error, external oscillator stop, D/PFlash ECC check error, and FCM frequency measurement anomaly.

During use, first enable the enable bit, that is, set the EMB control register EMB_CTL1.SYSEN to valid. Set the EMB_CTL1.LVDEN, LOCKUPEN, SRAMECCERREN, OSCSTPEN, FLASHECCERREN, and FCMERREN bits to enable the system error sources that need to be detected. When the selected system error source flag is set, the system error status EMB_STAT.SYSST in the EMB status register is set, and the EMB system error control flag bit EMB_STAT.SYSF is also set. When the interrupt enable register EMB_INTEN.SYSINTEN=1, an interrupt will also be generated.

The EMB control events are routed to Timer4 through EMB_RLSSEL.SYSRSEL, based on the status of EMB_STAT.SYSST or EMB_STAT.SYSF. After receiving the control event signal, 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.



Notice

When using the external oscillator stop as a system error, you need to enable both the CMU_XTALSTDCR.XTALSTDE and XTALSTIE bits after the external oscillator has stabilized.

18.2.7 Write Register Software Control PWM Signal Output

The EMB software output enable control register (EMB_SOE) allows users to send control signals to Timer4 directly through software setting and resetting. When the software controls the PWM output, no interrupt request is generated.

18.3 Register Description

18.3.1 Register List

Base Address: 0x40006C00

Table 18-4 Emergency Brake Module (EMB) Register Offset

Offset	Register	Description	Width
0x00	EMB_CTL1	EMB Control Register 1	32
0x04	EMB_CTL2	EMB Control Register 2	32
0x08	EMB_SOE	EMB Software Output Enable Register	32
0x0C	EMB_STAT	EMB Status Register	32
0x10	EMB_STATCLR	EMB Status Clear Register	32
0x14	EMB_INTEN	EMB Interrupt Enable Register	32
0x18	EMB_RLSSEL	EMB Control PWM Output Release Select Register	32

18.3.2 EMB Control Register 1 (EMB_CTL1)

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	LVDEN	LOCKUPEN	Reserved	SRAMECCERREN	OSCSTPEN	FLASHECCERREN	INVSEL4	INVSEL3	INVSEL2	INVSEL1	Reserved	PORTINEN4	PORTINEN3	PORTINEN2	PORTINEN1	FCMERREN	Reserved							PWMSEN2	PWMSEN1	PWMSEN0	SYSEN	Reserved			CMPEN	
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	LVDEN	LVD Detection Control Enabled	0b0: LVD detection result output is invalid 0b1: LVD detection result output is valid  Notice This bit is shared by LVD1 and LVD2.	RW
30	LOCKUPEN	Cortex-M0+ Lock-up Control Enabled	0b0: Output control is disabled when lockup occurs 0b1: Output control is enabled when lockup occurs	RW
29	Reserved	Reserved	-	-
28	SRAMECCERREN	SRAM ECC Check Error Control Enabled	0b0: Output control is disabled when SRAM ECC check error occurs 0b1: Output control is enabled when SRAM ECC check error occurs	RW
27	OSCSTPEN	Oscillator Stop Control Enabled	0b0: Output control is disabled when oscillator stop occurs 0b1: Output control is enabled when oscillator stop occurs	RW
26	FLASHECCERREN	FLASH ECC Check Error Control Enabled	0b0: Output control is disabled when FLASH ECC check error occurs 0b1: Output control is enabled when FLASH ECC check error occurs  Notice This bit is shared by PFLASH and DFLASH.	RW
25	INVSEL4	Port Input 4 Active Level Selection	0b0: High level enabled 0b1: Low level enabled	RW

Bit	Name	Function	Description	Access
24	INVSEL3	Port Input 3 Active Level Selection	0b0: High level enabled 0b1: Low level enabled	RW
23	INVSEL2	Port Input 2 Active Level Selection	0b0: High level enabled 0b1: Low level enabled	RW
22	INVSEL1	Port Input 1 Active Level Selection	0b0: High level enabled 0b1: Low level enabled	RW
21:20	Reserved	Reserved	-	-
19	PORTINEN4	Port Input 4 Control Enabled	0b0: Port input control disabled 0b1: Port input control enabled	RW
18	PORTINEN3	Port Input 3 Control Enabled	0b0: Port input control disabled 0b1: Port input control enabled	RW
17	PORTINEN2	Port Input 2 Control Enabled	0b0: Port input control disabled 0b1: Port input control enabled	RW
16	PORTINEN1	Port Input 1 Control Enabled	0b0: Port input control disabled 0b1: Port input control enabled	RW
15	FCMERREN	FCM Frequency Measurement Abnormality Control Enabled	0b0: The output control is disabled when the FCM frequency measurement is abnormal 0b1: The output control is enabled when the FCM frequency measurement is abnormal	RW
14:8	Reserved	Reserved	-	-
7	PWMSEN2	TMR4_OUH/L Short Circuit Output Control Enabled	0b0: Output control is disabled when short circuit occurs 0b1: Output control is enabled when short circuit occurs	RW
6	PWMSEN1	TMR4_OVH/L Short Circuit Output Control Enabled	0b0: Output control is disabled when short circuit occurs 0b1: Output control is enabled when short circuit occurs	RW
5	PWMSEN0	TMR4_OWH/L Short Circuit Output Control Enabled	0b0: Output control is disabled when short circuit occurs 0b1: Output control is enabled when short circuit occurs	RW
4	SYSEN	System Error Control Enabled	0b0: Output control is disabled when system error occurs 0b1: Output control is enabled when system error occurs	RW
3:1	Reserved	Reserved	-	-

Bit	Name	Function	Description	Access
0	CMPEN	CMP Voltage Comparator Comparison Result Control Enabled	● 0b0: Voltage comparator comparison result output control disabled ● 0b1: Voltage comparator comparison result output control enabled	RW

18.3.3 EMB Control Register 2 (EMB_CTL2)

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	0b0: Filter disabled 0b1: Filter enabled	RW
26:25	NFSEL4	Port Input 4 Digital Filter Clock Selection	0b00: Use bus clock filtering 0b01: Filtering with 8-division of bus clock 0b10: Filtering with 32-division of bus clock 0b11: Filtering with 128-division of bus clock	RW
24	NFEN3	Port Input 3 Digital Filter Enable	0b0: Filter disabled 0b1: Filter enabled	RW
23:22	NFSEL3	Port Input 3 Digital Filter Clock Selection	0b00: Use bus clock filtering 0b01: Filtering with 8-division of bus clock 0b10: Filtering with 32-division of bus clock 0b11: Filtering with 128-division of bus clock	RW
21	NFEN2	Port Input 2 Digital Filter Enable	0b0: Filter disabled 0b1: Filter enabled	RW
20:19	NFSEL2	Port Input 2 Digital Filter Clock Selection	0b00: Use bus clock filtering 0b01: Filtering with 8-division of bus clock 0b10: Filtering with 32-division of bus clock 0b11: Filtering with 128-division of bus clock	RW
18	NFEN1	Port Input 1 Digital Filter Enable	0b0: Filter disabled 0b1: Filter enabled	RW
17:16	NFSEL1	Port Input 1 Digital Filter Clock Selection	0b00: Use bus clock filtering 0b01: Filtering with 8-division of bus clock 0b10: Filtering with 32-division of bus clock 0b11: Filtering with 128-division of bus clock	RW
15:3	Reserved	Reserved	-	-
2	PWMLV2	TMR4_OUH/L Output Active Level Selection	0b0: Low level is the active level 0b1: High level is the active level	RW

Bit	Name	Function	Description	Access
1	PWMLV1	TMR4_OVH/L Output Active Level Selection	0b0: Low level is the active level 0b1: High level is the active level	RW
0	PWMLV0	TMR4_OVH/L Output Active Level Selection	0b0: Low level is the active level 0b1: High level is the active level	RW

18.3.4 EMB Software Output Enable 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	0b0: Normal PWM output 0b1: PWM output stops	RW

18.3.5 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	0b0: Port input control is in the disable level state 0b1: Port input control is in the enable level state	RO
16	PORTINST3	Port Input 3 Control Status	0b0: Port input control is in the disable level state 0b1: Port input control is in the enable level state	RO
15	PORTINST2	Port Input 2 Control Status	0b0: Port input control is in the disable level state 0b1: Port input control is in the enable level state	RO
14	PORTINST1	Port Input 1 Control Status	0b0: Port input control is in the disable level state 0b1: Port input control is in the enable level state	RO
13:12	Reserved	Reserved	-	-
11	PORTINF4	Port Input 4 Control Flag Bit	Set when EMB_STAT.PORTINST4=1. Clear 0 when EMB_STATCLR.PORTINFCLR4 writes 1	RO
10	PORTINF3	Port Input 3 Control Flag Bit	Set when EMB_STAT.PORTINST3=1. Clear 0 when EMB_STATCLR.PORTINFCLR3 writes 1	RO
9	PORTINF2	Port Input 2 Control Flag Bit	Set when EMB_STAT.PORTINST2=1. Clear 0 when EMB_STATCLR.PORTINFCLR2 writes 1	RO
8	PORTINF1	Port Input 1 Control Flag Bit	Set when EMB_STAT.PORTINST1 is 1 Clear 0 when EMB_STATCLR.PORTINFCLR1 writes 1	RO
7	SYSST	System Error Status	0b0: No system error 0b1: System error occurs	RO

Bit	Name	Function	Description	Access
6	CMPST	Voltage Com- parator Status	● 0b0: The voltage comparator comparison result flag bit is not set ● 0b1: The voltage comparator comparison result flag bit is set	RO
5	PWMST	PWM Output Status	● 0b0: No PWM output in-phase occurs ● 0b1: PWM output in-phase occurs	RO
4	<i>Reserved</i>	<i>Reserved</i>	-	-
3	SYSF	System Error Flag Bit	Set when EMB_STAT.SYSST=1. Clear 0 when EMB_STATCLR.SYSFCLR writes 1	RO
2	CMPF	CMP Control Flag Bit	Set when EMB_STAT.CMPST is 1 Clear 0 when EMB_STATCLR.CMPFCLR writes 1	RO
1	PWMSF	PWM Output In- phase Control Flag Bit	Set when EMB_STAT.PWMST=1. Clear 0 when EMB_STATCLR.PWMSCLR writes 1	RO
0	<i>Reserved</i>	<i>Reserved</i>	-	-

18.3.6 EMB Status Clear 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	-	-

18.3.7 EMB Interrupt Enable 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																				PORTININTEN4	PORTININTEN3	PORTININTEN2	PORTININTEN1	Reserved				SYSINTEN	CMPINTEN	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	0b0: Port input control does not generate interrupt 0b1: Port input control generates interrupt	RW
10	PORTININTEN3	Port Input 3 Control Interrupt Enabled	0b0: Port input control does not generate interrupt 0b1: Port input control generates interrupt	RW
9	PORTININTEN2	Port Input 2 Control Interrupt Enabled	0b0: Port input control does not generate interrupt 0b1: Port input control generates interrupt	RW
8	PORTININTEN1	Port Input 1 Control Interrupt Enabled	0b0: Port input control does not generate interrupt 0b1: Port input control generates interrupt	RW
7:4	Reserved	Reserved	-	-
3	SYSINTEN	System Error Control Interrupt Enabled	0b0: System error control does not generate interrupt 0b1: System error control generates interrupt	RW
2	CMPINTEN	CMP Comparison Result Control Interrupt Enabled	0b0: CMP comparison result control does not generate interrupt 0b1: CMP comparison result control generates interrupt	RW
1	PWMSINTEN	PWM Output In-phase Control Interrupt Enabled	0b0: PWM output in-phase control does not generate interrupt 0b1: PWM output in-phase control generates interrupt	RW
0	Reserved	Reserved	-	-

18.3.8 EMB Control PWM Output Release Select 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	0b0: Release PWM output control when EMB_STAT.PORTINF4=0 0b1: Release PWM output control when EMB_STAT.PORTINST4=0	RW
10	PORTINRSEL3	Port Input 3 Control Release Selection	0b0: Release PWM output control when EMB_STAT.PORTINF3=0 0b1: Release PWM output control when EMB_STAT.PORTINST3=0	RW
9	PORTINRSEL2	Port Input 2 Control Release Selection	0b0: Release PWM output control when EMB_STAT.PORTINF2=0 0b1: Release PWM output control when EMB_STAT.PORTINST2=0	RW
8	PORTINRSEL1	Port Input 1 Control Release Selection	0b0: Release PWM output control when EMB_STAT.PORTINF1=0 0b1: Release PWM output control when EMB_STAT.PORTINST1=0	RW
7:4	Reserved	Reserved	-	-
3	SYSRSEL	System Error Control Release Selection	0b0: Release PWM output control when EMB_STAT.SYSF=0 0b1: Release PWM output control when EMB_STAT.SYSST=0	RW
2	CMRSEL	CMP Comparison Result Control Release Selection	0b0: Release PWM output control when EMB_STAT.CMPF=0 0b1: Release PWM output control when EMB_STAT.CMPST=0	RW
1	PWMRSEL	PWM Output Control Release Selection	0b0: Release PWM output control when EMB_STAT.PWMSF=0 0b1: Release PWM output control when EMB_STAT.PWMST=0	RW
0	Reserved	Reserved	-	-

19 General Timer (TimerA)

19.1 Introduction

The General Timer A (TimerA) is a timer with a 16-bit counting width and 8-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 2-phase quadrature encoding count and 3-phase quadrature encoding count. This product series is equipped with 1 TimerA.

19.2 Features

The basic functions and features of TimerA are shown in the table below:

Table 19-1 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
	Quadrature encoding count
	8 PWM output
	Compare match event output
Interrupt type	Compare match interrupt
	Period match interrupt

19.3 Functional Description

19.3.1 TimerA Block Diagram

The basic block diagram of TimerA is shown in the figure.

This series of products carries 1 unit of TimerA, the bus clock and count clock are both HCLK.

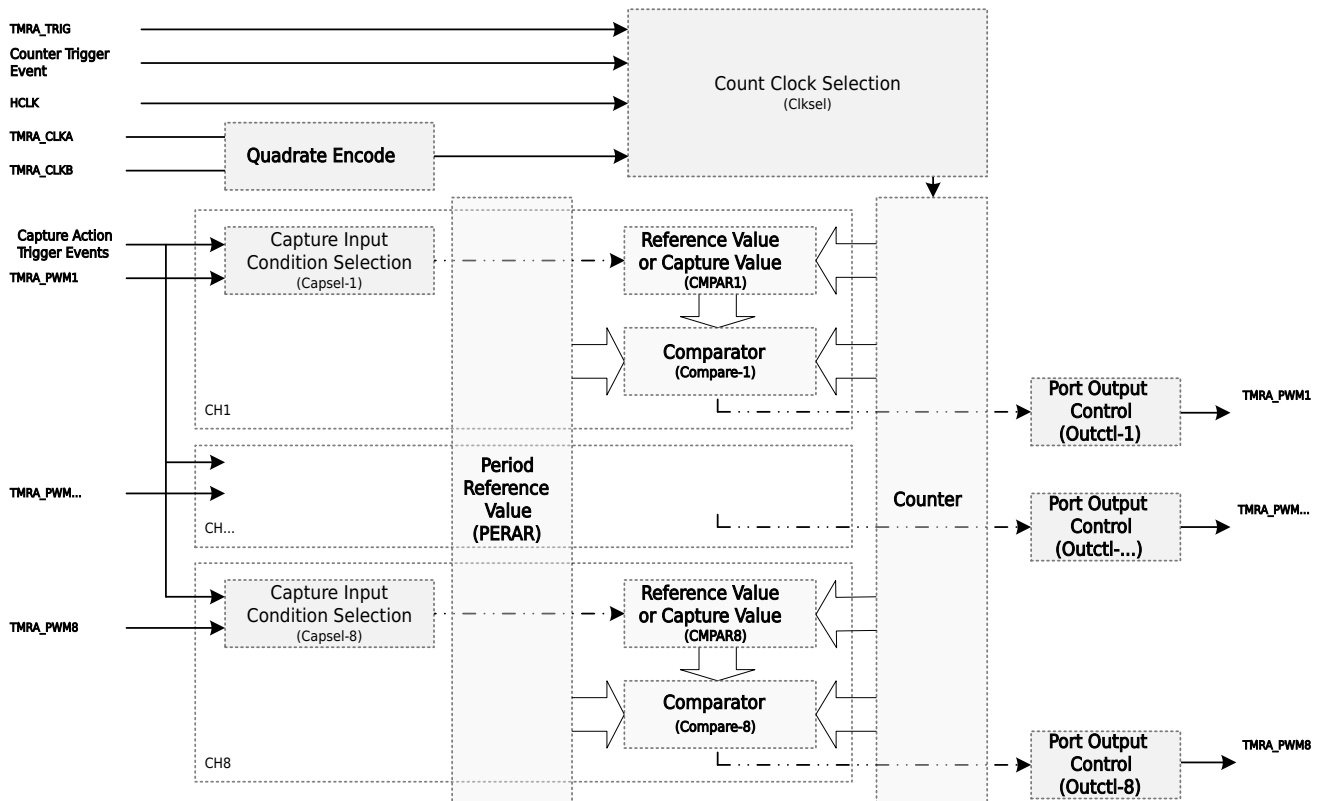


Figure 19-1 TimerA Basic Block Diagram

The following table lists the input and output ports of TimerA.

Table 19-2 TimerA Port List

Port Name	Direction	Function
TMRA_PWMm	Input/Output	Capture input event port or PWM output port (m=1-8)
TMRA_CLKA	Input	Quadrature encoder counting event input port
TMRA_CLKB		
TMRA_TRIG	Input	Hardware-triggered start, stop, clear event input port

19.3.2 Basic Actions

19.3.2.1 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:

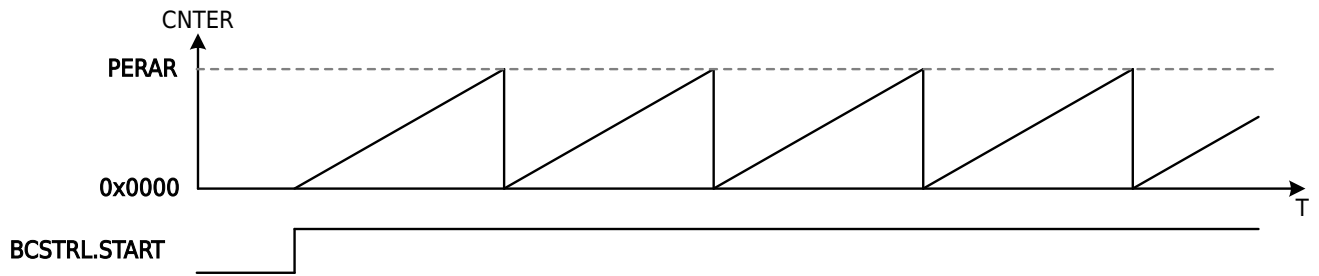


Figure 19-2 Sawtooth Waveform (Counting Up)

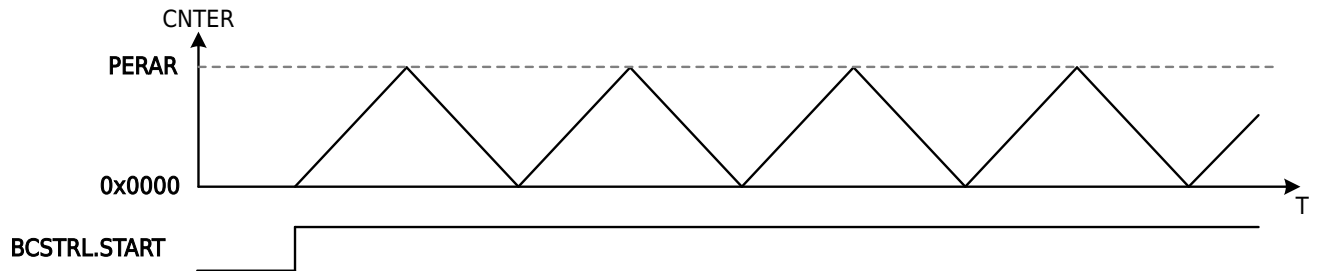


Figure 19-3 Triangle Waveform

19.3.2.2 Compare Output

TimerA unit contains 8 channels compare output (TMRA_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_PWMm output ports. When the timer count value equals TMRA_CMPARM, the TMRA_PWMm port outputs the specified level (m=1-8).

The output level of the TMRA_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=1-8). The diagram below illustrates an example of compare output action.

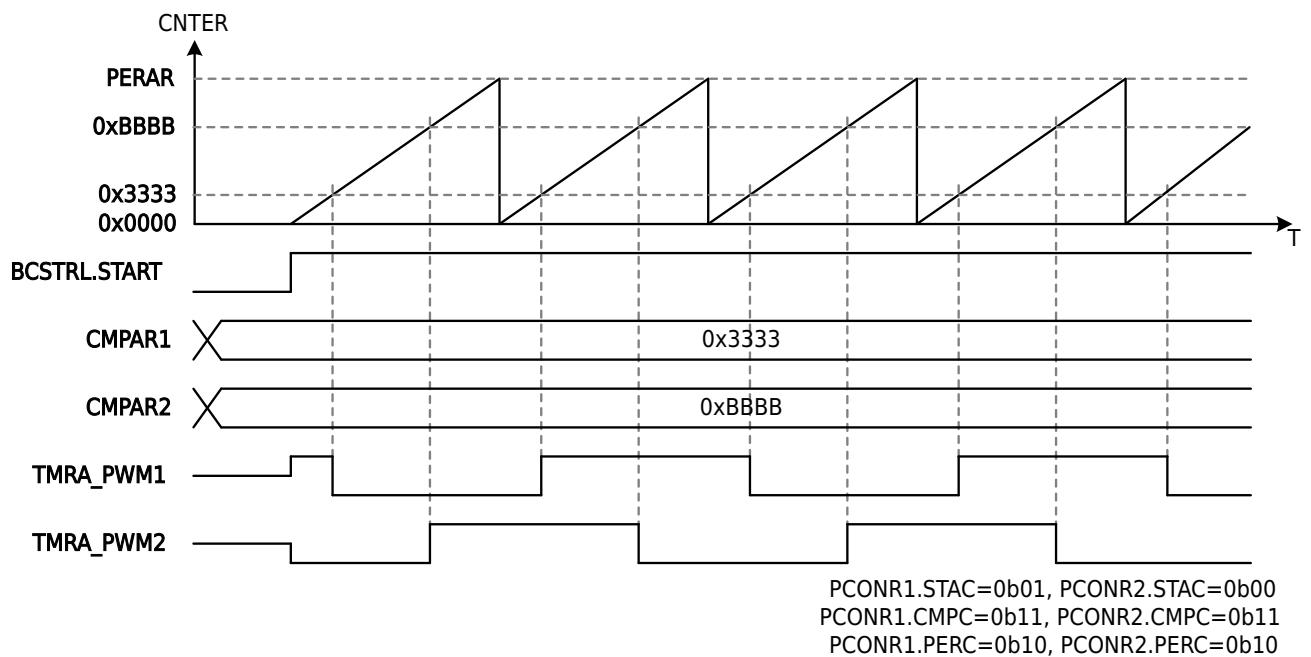


Figure 19-4 Compare Output Action

19.3.2.3 Capture Input

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=1-8).

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=0-1) " in the "Automatic Operation System (AOS) " chapter), TMRA_PWMm port input, etc. The specific condition selection can be set through the CCONRm.HICP bit (m=1-8).

The specific allocation of the internal trigger event selection register is shown in the following table.

Table 19-3 TRGSEL Selection Relationship Table for Internal Trigger Events

TimerA Unit	Event Classification	Corresponding Selection Register
TimerA	Counter Trigger Events: TRGSEL selection events specified by HCONR, HCUPR, and HCDOR	TMRA_TRGSEL0
TimerA	Capture Action Trigger Events: TRGSEL selection events specified by CCONR	TMRA_TRGSEL1

The following figure shows an example of the capture input action.

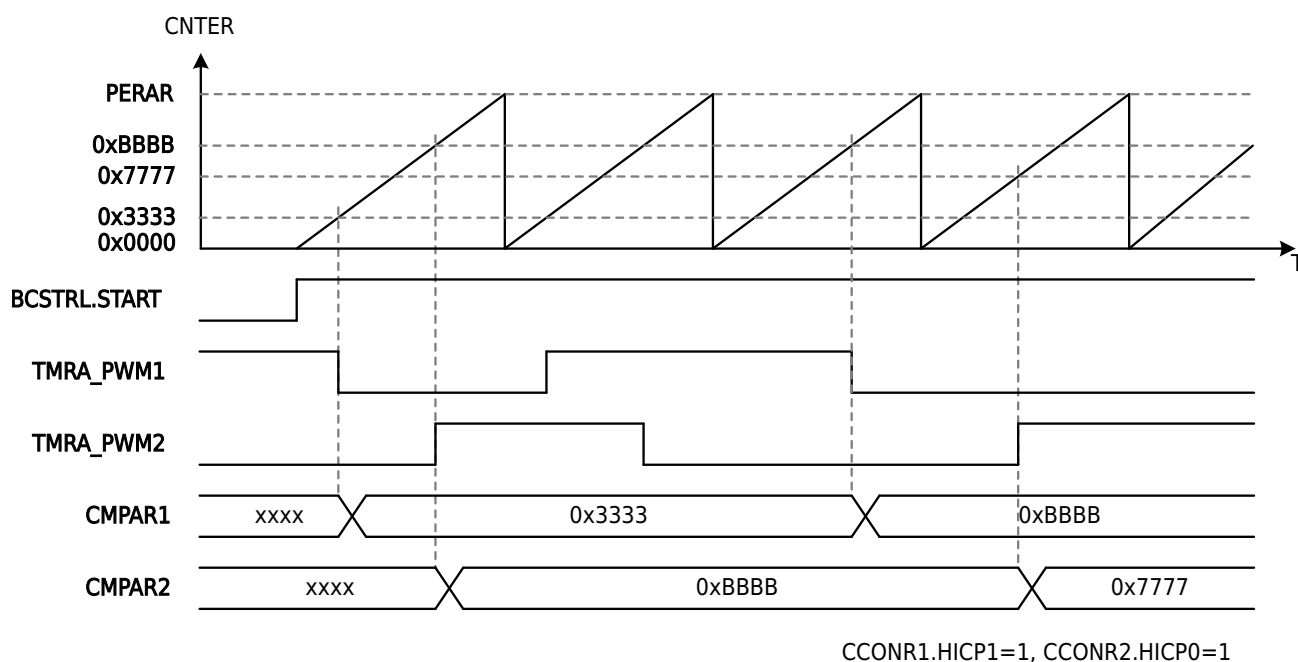


Figure 19-5 Capture Input Action

19.3.3 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 HCLK (configured by BCSTRL.CKDIV)
2. TMRA_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. TMRA_CLKA and TMRA_CLKB port quadrature encoder input (configured by 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.

In hardware counting mode:

- Each count clock source is independent and can be enabled or disabled individually.
- The software clock is automatically disabled.

19.3.4 Count Direction

The count direction of TimerA can be changed via software. The method of changing the counting direction may vary slightly depending on the selected waveform mode.

19.3.4.1 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.

19.3.4.2 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.

19.3.5 Digital Filtering

TMRA_CLKA, TMRA_CLKB, TMRA_TRIG, TMRA_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=1-8).

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_CLKA port filtering action.

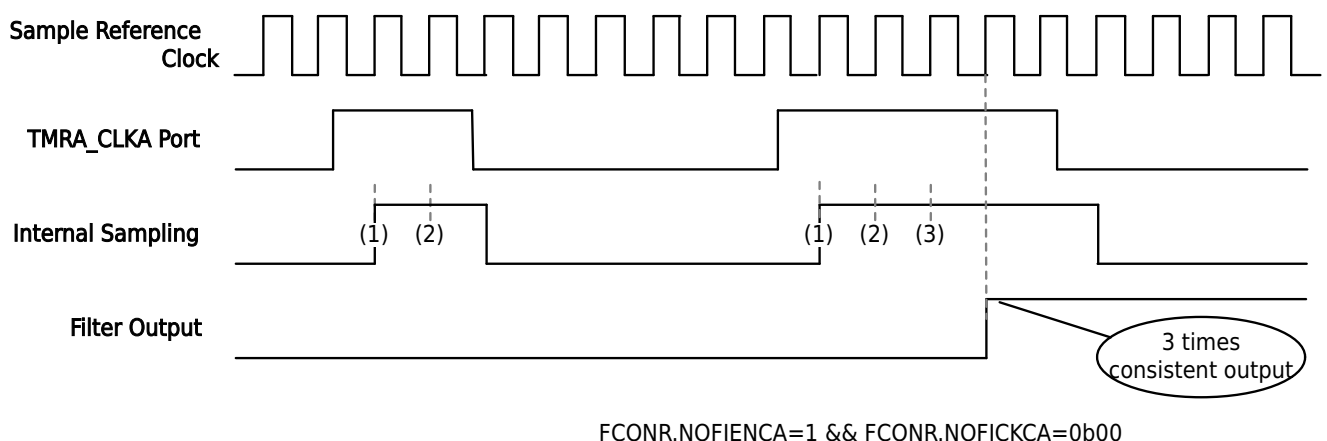


Figure 19-6 Filtering Function of Clock Input Port

19.3.6 Synchronous Starting

The product is equipped with a Counter Synchronization Enable Register TMR_SYNENR to synchronously start all the timers (including Timer4, TimerA) carried by the product. Setting multiple enable bits in the TMR_SYNENR register to 1 allows two timers to start counting simultaneously. For the specific allocation of enable bits, please refer to "[Counter Synchronization Enable Register \(TMR_SYNENR\)](#)".

Usage Process

- Step1.** Set the enable bits in the TMR_SYNENR register to 1;
- Step2.** Set TMRA_BCSTR.L.START to 1;
- Step3.** Clear the enable bits in the TMR_SYNENR register to 0.



Notice

When using the synchronous startup function, ensure that the HCLK clock frequencies of all timers being synchronously started are identical.

19.3.7 Buffer Function

A total of 8 compare reference registers (CMPAR_m) (m=1-8) 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, CMPAR₆ is used as the buffer value of CMPAR₅, and CMPAR₈ is used as the buffer value of CMPAR₇. The buffer control register (BCONR_m) (m=1-4) controls the four groups of buffer functions respectively.

When the BCONR_m.BEN (m=1-4) bit is set, the buffer function becomes active. When the counter reaches a certain time point, a buffer transmission occurs (CMPAR_{2/4/6/8}→CMPAR_{1/3/5/7}). 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 (BCSTR.L.MODE=0), the counter counts to the overflow point (BCSTR.L.DIR=1) or underflow point (BCSTR.L.DIR=0) or the hardware clears the counter.
3. In triangle wave counting mode (BCSTR.L.MODE=1), count to peak point (BCSTR.L.DIR=1 && BCONR_m.BSE0=1) (m=1-4)
4. In triangle wave counting mode (BCSTR.L.MODE=1), count to valley point (BCSTR.L.DIR=0 && BCONR_m.BSE1=1) (m=1-4)

The figure below is a schematic diagram of buffer transmission in sawtooth mode.

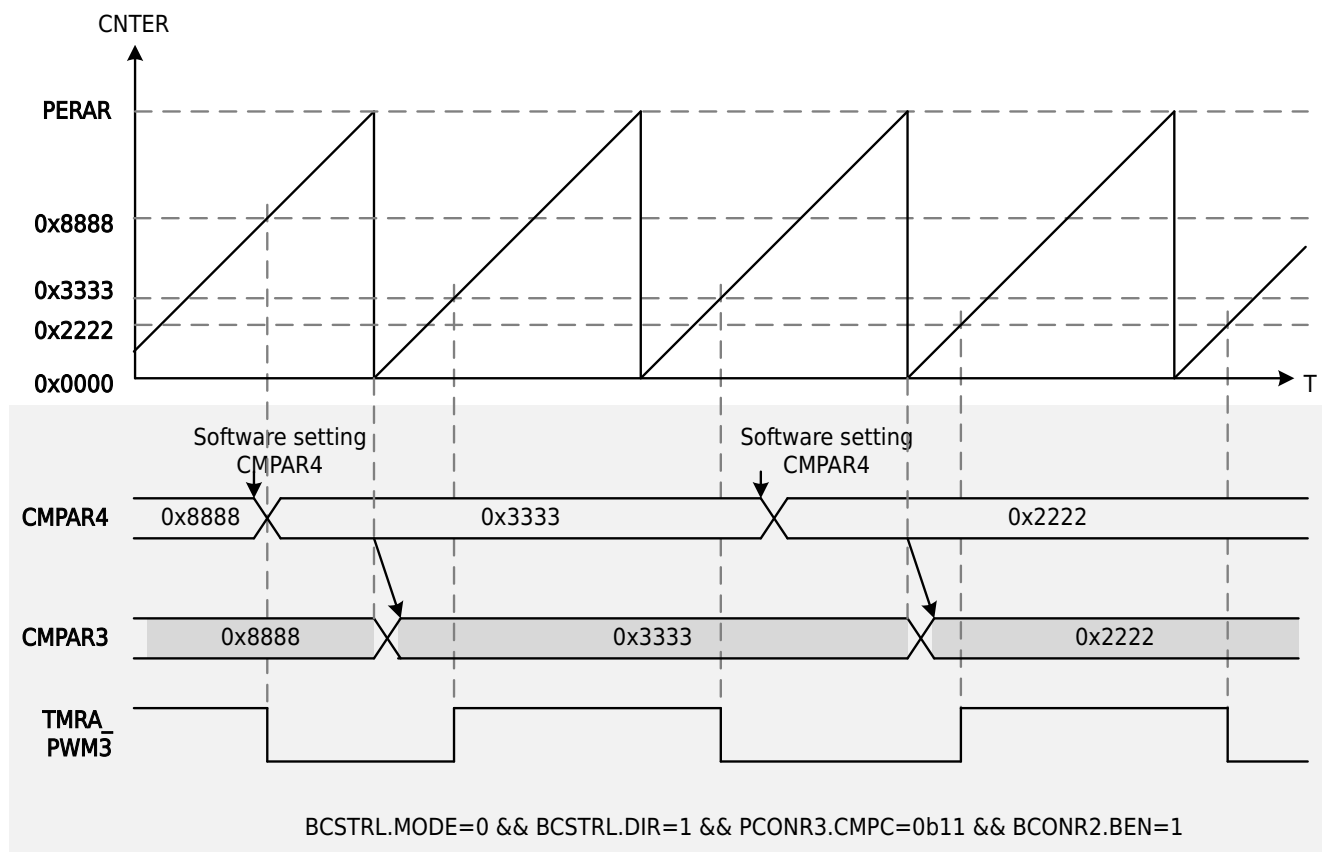


Figure 19-7 Buffer Action in Sawtooth Mode

19.3.8 General PWM Output

19.3.8.1 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 (PCNR.CMPC=0b11) and also to toggle on period match (PCNR.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.

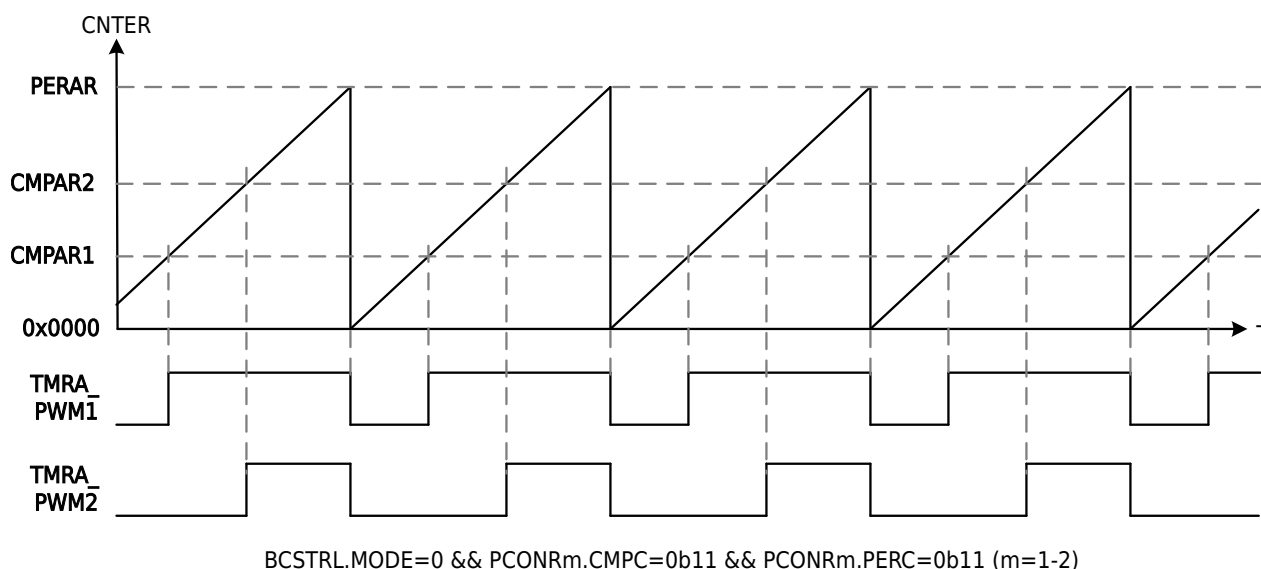


Figure 19-8 Example of Single-Side Aligned PWM Output

19.3.8.2 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.

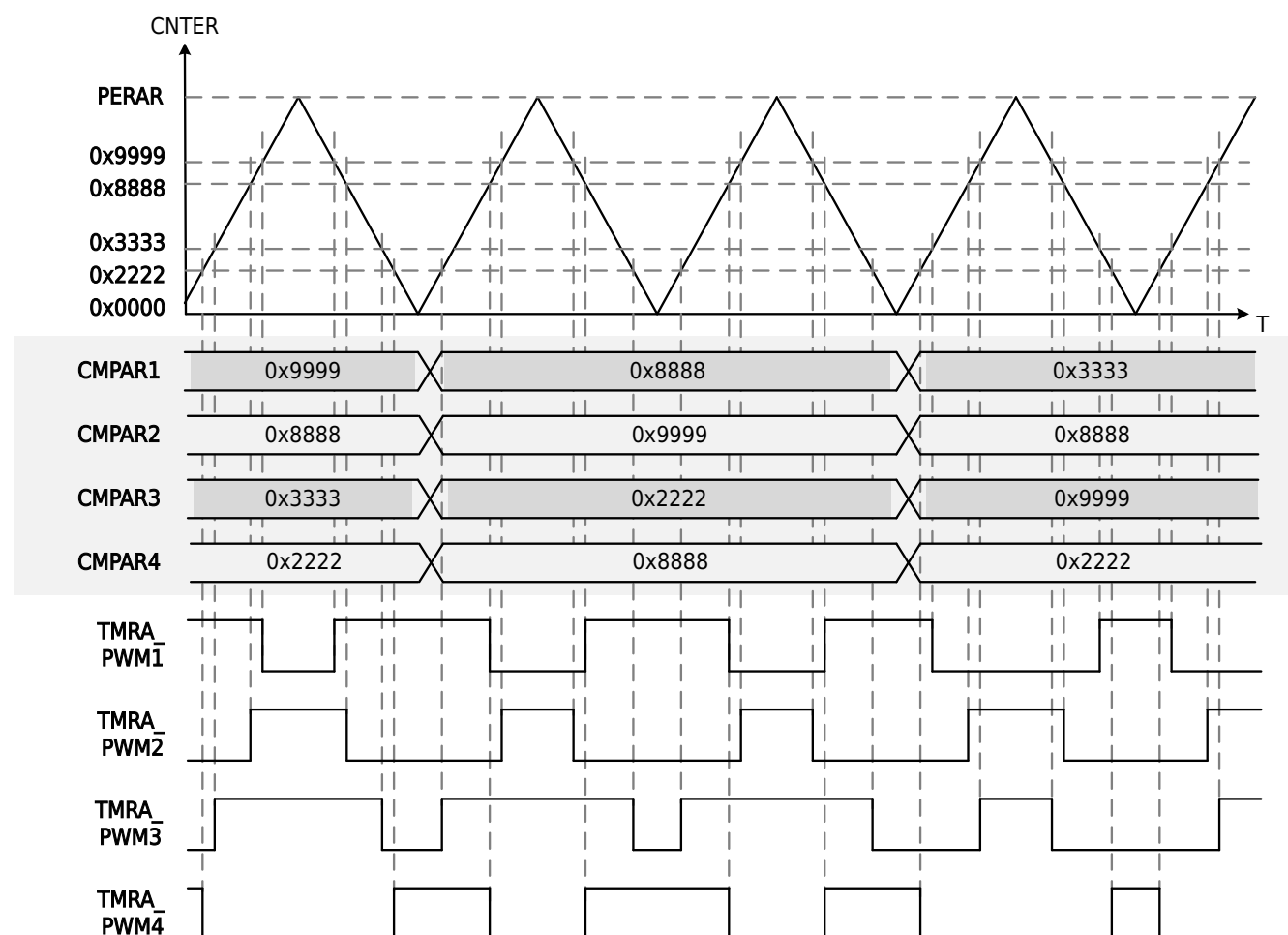


Figure 19-9 Example of Double-Side Symmetric PWM Output

19.3.9 Quadrature Encoding Counting

By considering the TMRA_CLKA input as AIN input, the TMRA_CLKB input as BIN input, and the TMRA_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 counting condition enable of AIN and BIN is realized by setting the quadrature relationship between TMRA_CLKA and TMRA_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.

19.3.9.1 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.

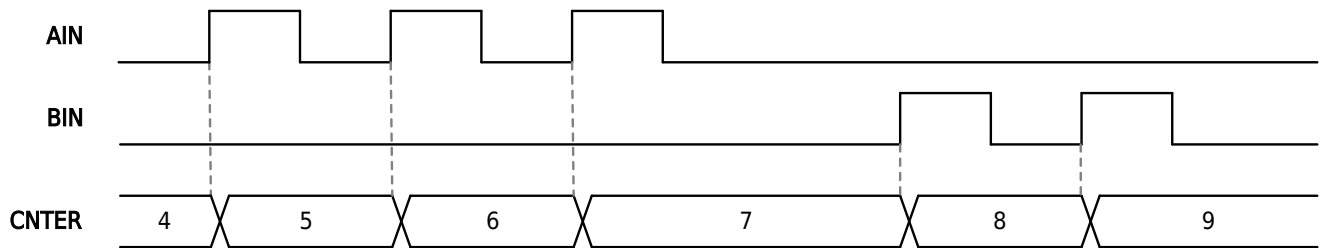
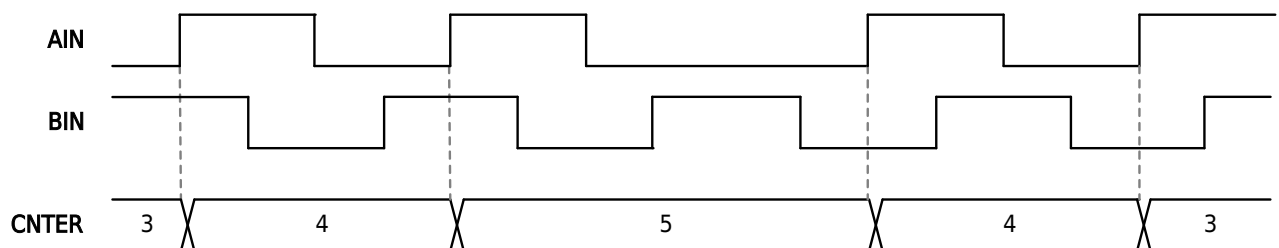


Figure 19-10 Position Mode-Basic Counting

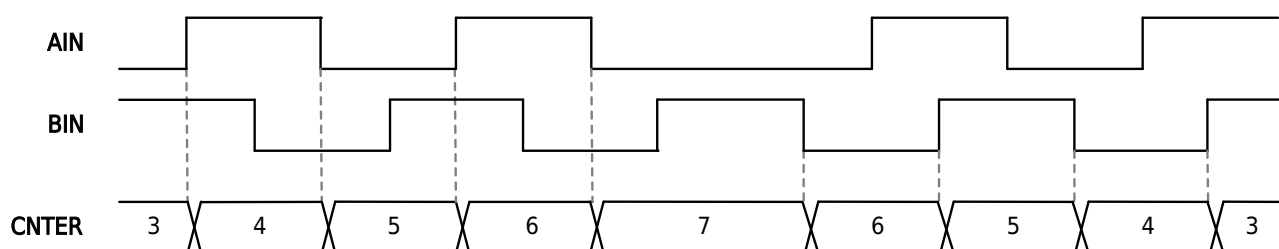
Phase Difference Counting

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.



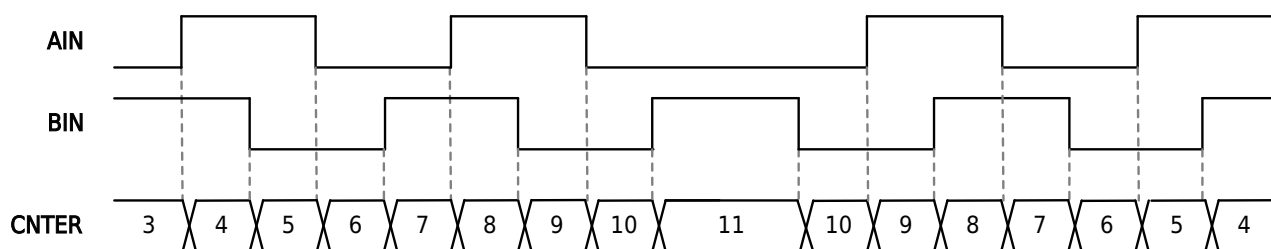
HCUPR.HCUP6=0b1 && HCDOR.HCDO4=0b1

Figure 19-11 Position Counting Mode-Phase Difference Counting (1x Counting)



HCUPR.HCUP6=HCUPR.HCUP5=0b1 && HCDOR.HCDO2=HCDOR.HCDO1=0b1

Figure 19-12 Position Counting Mode-Phase Difference Counting (2x Counting)

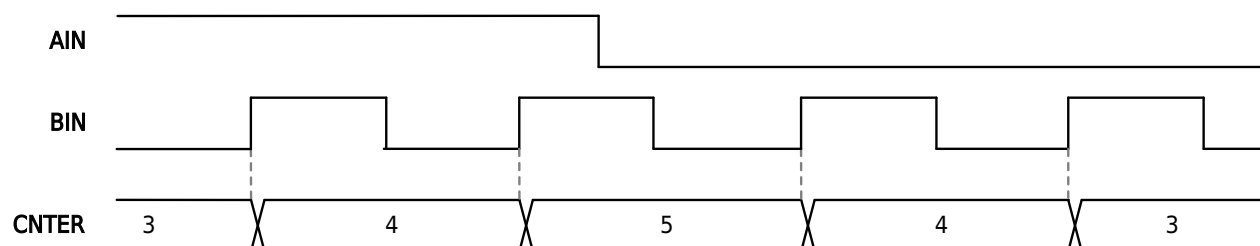


HCUPR.HCUP6=HCUPR.HCUP5=HCUPR.HCUP3=HCUPR.HCUP0=0b1
HCDOR.HCDO7=HCDOR.HCDO4=HCDOR.HCDO2=HCDOR.HCDO1=0b1

Figure 19-13 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 19-14 Position Counting Mode-Direction Counting

19.3.10 Interrupt and Events

TimerA contains 3 interrupt outputs and 3 event outputs, which are 1 comparison match interrupt and event, 2 period match interrupts and events.

19.3.10.1 Comparison 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_CMP) will be triggered; if the ECONR.ETENm bit is set to 1, the corresponding event request (TMRA_CMP) will be triggered (m=1-8).

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_CMP) will be triggered; if the ECONR.ETENm bit is set to 1, the corresponding event request (TMRA_CMP) will be triggered (m=1-8).

The comparison match interrupts and comparison match events of the 8 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.

19.3.10.2 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_OVF and TMRA_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_OVF and TMRA_UDF) output is triggered at the corresponding counting period point to select triggering other modules.

19.4 Register Description

19.4.1 Register List

Base address: 0x40007000

Table 19-4 General Timer (TimerA) Register Offset

Offset	Register	Description	Width
0x0000	TMRA_CNTER	General Counter Value Register	16
0x0004	TMRA_PERAR	Period Reference Value Register	16
0x003C+0x4*m	TMRA_CMPARm	Compare Reference Value Register m (m=1-8)	16
0x0080	TMRA_BCSTRL	Basic Control and Status Register L	8
0x0081	TMRA_BCSTRH	Basic Control and 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=1-4)	16
0x00FC+0x4*m	TMRA_CCONRm	Capture Control Register m (m=1-8)	16
0x013C+0x4*m	TMRA_PCONRm	Port Control Register m (m=1-8)	16
0x0084	TMRA_HCONR	Hardware Control Event Select Register	16
0x0088	TMRA_HCUPR	Hardware Count Up Event Select Register	16
0x008C	TMRA_HCDOR	Hardware Count Down Event Select Register	16

TMR_SYNENR base address: 0x40013400

Table 19-5 TMR_SYNENR Register Offset

Offset	Register	Description	Width
0x0000	TMR_SYNENR	Counter Synchronization Enable Register	32

19.4.2 General Counter 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	Current timer count value  Note 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.	RW

19.4.3 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

19.4.4 Compare Reference Value Register (TMRA_CMPARm) (m=1-8)

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

19.4.5 Basic Control and Status Register L (TMRA_BCSTR_L)

Offset	Bit Position							
0x0080	7	6	5	4	3	2	1	0
Reset	0x02							
Name	CKDIV				Reserved	MODE	DIR	START
Access	RW					RW	RW	RW

Bit	Name	Function	Description	Access
7:4	CKDIV	Count Clock Selection	0b0000: HCLK 0b0001: HCLK/2 0b0010: HCLK/4 0b0011: HCLK/8 0b0100: HCLK/16 0b0101: HCLK/32 0b0110: HCLK/64 0b0111: HCLK/128 0b1000: HCLK/256 0b1001: HCLK/512 0b1010: HCLK/1024 - Others: Setting prohibited	RW
3	Reserved	Reserved	-	-
2	MODE	Counting Mode	0b0: Sawtooth wave mode 0b1: Triangle wave mode	RW
1	DIR	Count Direction	0b0: Counter count down 0b1: Counter count up  Note In triangle wave mode, this bit indicates the direction status of the counter.	RW
0	START	Timer Start	0b0: Timer disabled 0b1: Timer enabled  Note This bit will automatically be cleared to 0 when the hardware stop condition is active.	RW

**Notice**

1. In triangle wave mode, do not write to this register during timer operation.

19.4.6 Basic Control and 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	0b0: No counter underflow occurred during down counting 0b1: Counter underflow occurred during down counting	RW
6	OVFF	Overflow Flag	0b0: No counter overflow occurred during up counting 0b1: Counter overflow occurred during up counting	RW
5	ITENUDF	Underflow Interrupt Enabled	0b0: Disable 0b1: Enable	RW
4	ITENOVF	Overflow Interrupt Enabled	0b0: Disable 0b1: Enable	RW
3:1	<i>Reserved</i>	<i>Reserved</i>	-	-
0	OVSTP	Counting Overflow Stop Control	0b0: The counter continues counting after an overflow or underflow 0b1: The counter stops counting after an overflow or underflow  Note The TMRA_CNTER register must be initialized before restarting the timer after an overflow stop.	RW

19.4.7 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								ITEN8	ITEN7	ITEN6	ITEN5	ITEN4	ITEN3	ITEN2	ITEN1
Access									RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	Function	Description	Access
15:8	Reserved	Reserved	-	-
7	ITEN8	Count Match Interrupt Enable 8	Interrupt Condition: Triggered when the value of the CMPAR8 register matches the counter value (CNTENR), or when an input capture occurs. 0b0: Disable interrupt output 0b1: Enable interrupt output	RW
6	ITEN7	Count Match Interrupt Enable 7	Interrupt Condition: Triggered when the value of the CMPAR7 register matches the counter value (CNTENR), or when an input capture occurs. 0b0: Disable interrupt output 0b1: Enable interrupt output	RW
5	ITEN6	Count Match Interrupt Enable 6	Interrupt Condition: Triggered when the value of the CMPAR6 register matches the counter value (CNTENR), or when an input capture occurs. 0b0: Disable interrupt output 0b1: Enable interrupt output	RW
4	ITEN5	Count Match Interrupt Enable 5	Interrupt Condition: Triggered when the value of the CMPAR5 register matches the counter value (CNTENR), or when an input capture occurs. 0b0: Disable interrupt output 0b1: Enable interrupt output	RW
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. 0b0: Disable interrupt output 0b1: Enable interrupt output	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. 0b0: Disable interrupt output 0b1: Enable interrupt output	RW

Bit	Name	Function	Description	Access
1	ITEN2	Count Match Interrupt Enable 2	Interrupt Condition: Triggered when the value of the CMPAR2 register matches the counter value (CNTERR), or when an input capture occurs. ● 0b0: Disable interrupt output ● 0b1: Enable interrupt output	RW
0	ITEN1	Count Match Interrupt Enable 1	Interrupt Condition: Triggered when the value of the CMPAR1 register matches the counter value (CNTERR), or when an input capture occurs. ● 0b0: Disable interrupt output ● 0b1: Enable interrupt output	RW

19.4.8 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								ETEN8	ETEN7	ETEN6	ETEN5	ETEN4	ETEN3	ETEN2	ETEN1
Access									RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	Function	Description	Access
15:8	Reserved	Reserved	-	-
7	ETEN8	Count Match Events Enable 8	Events Condition: Triggered when the value of the CMPAR8 register matches the counter value (CNTENR), or when an input capture occurs. 0b0: Disable event output 0b1: Enable event output	RW
6	ETEN7	Count Match Events Enable 7	Events Condition: Triggered when the value of the CMPAR7 register matches the counter value (CNTENR), or when an input capture occurs. 0b0: Disable event output 0b1: Enable event output	RW
5	ETEN6	Count Match Events Enable 6	Events Condition: Triggered when the value of the CMPAR6 register matches the counter value (CNTENR), or when an input capture occurs. 0b0: Disable event output 0b1: Enable event output	RW
4	ETEN5	Count Match Events Enable 5	Events Condition: Triggered when the value of the CMPAR5 register matches the counter value (CNTENR), or when an input capture occurs. 0b0: Disable event output 0b1: Enable event output	RW
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. 0b0: Disable event output 0b1: Enable event output	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. 0b0: Disable event output 0b1: Enable event output	RW

Bit	Name	Function	Description	Access
1	ETEN2	Count Match Events Enable 2	Events Condition: Triggered when the value of the CMPAR2 register matches the counter value (CNTER), or when an input capture occurs. ● 0b0: Disable event output ● 0b1: Enable event output	RW
0	ETEN1	Count Match Events Enable 1	Events Condition: Triggered when the value of the CMPAR1 register matches the counter value (CNTER), or when an input capture occurs. ● 0b0: Disable event output ● 0b1: Enable event output	RW

19.4.9 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	Reserved	Reserved	-	-
14:13	NOFICKCB	Filter Sampling Reference Clock Selection CB	0b00: HCLK 0b01: HCLK/4 0b10: HCLK/16 0b11: HCLK/64	RW
12	NOFIENCB	Capture Input Port Filter CB	0b0: Input filtering function for the TMRA_CLKB port is disabled 0b1: Input filtering function for the TMRA_CLKB port is enabled	RW
11	Reserved	Reserved	-	-
10:9	NOFICKCA	Filter Sampling Reference Clock Selection CA	0b00: HCLK 0b01: HCLK/4 0b10: HCLK/16 0b11: HCLK/64	RW
8	NOFIENCA	Capture Input Port Filter CA	0b0: Input filtering function for the TMRA_CLKA port is disabled 0b1: Input filtering function for the TMRA_CLKA port is enabled	RW
7:3	Reserved	Reserved	-	-
2:1	NOFICKTG	Filter Sampling Reference Clock Selection TG	0b00: HCLK 0b01: HCLK/4 0b10: HCLK/16 0b11: HCLK/64	RW
0	NOFIENTG	Capture Input Port Filter TG	0b0: Input filtering function for the TMRA_TRIG port is disabled 0b1: Input filtering function for the TMRA_TRIG port is enabled	RW

19.4.10 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	ICPF8	ICPF7	ICPF6	ICPF5	ICPF4	ICPF3	ICPF2	ICPF1	CMPF8	CMPF7	CMPF6	CMPF5	CMPF4	CMPF3	CMPF2	CMPF1
Access	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	Function	Description	Access
15	ICPF8	Repeat Capture Flag 8	Write 0 to clear the flag, write 1 has no effect Read: 0b0: No capture completion action has occurred after the CMPF8 flag was set to 1 0b1: A capture completion action has occurred again after the CMPF8 flag was set to 1	RW
14	ICPF7	Repeat Capture Flag 7	Write 0 to clear the flag, write 1 has no effect Read: 0b0: No capture completion action has occurred after the CMPF7 flag was set to 1 0b1: A capture completion action has occurred again after the CMPF7 flag was set to 1	RW
13	ICPF6	Repeat Capture Flag 6	Write 0 to clear the flag, write 1 has no effect Read: 0b0: No capture completion action has occurred after the CMPF6 flag was set to 1 0b1: A capture completion action has occurred again after the CMPF6 flag was set to 1	RW
12	ICPF5	Repeat Capture Flag 5	Write 0 to clear the flag, write 1 has no effect Read: 0b0: No capture completion action has occurred after the CMPF5 flag was set to 1 0b1: A capture completion action has occurred again after the CMPF5 flag was set to 1	RW
11	ICPF4	Repeat Capture Flag 4	Write 0 to clear the flag, write 1 has no effect Read: 0b0: No capture completion action has occurred after the CMPF4 flag was set to 1 0b1: A capture completion action has occurred again after the CMPF4 flag was set to 1	RW
10	ICPF3	Repeat Capture Flag 3	Write 0 to clear the flag, write 1 has no effect Read: 0b0: No capture completion action has occurred after the CMPF3 flag was set to 1 0b1: A capture completion action has occurred again after the CMPF3 flag was set to 1	RW

Bit	Name	Function	Description	Access
9	ICPF2	Repeat Capture Flag 2	Write 0 to clear the flag, write 1 has no effect Read: 0b0: No capture completion action has occurred after the CMPF2 flag was set to 1 0b1: A capture completion action has occurred again after the CMPF2 flag was set to 1	RW
8	ICPF1	Repeat Capture Flag 1	Write 0 to clear the flag, write 1 has no effect Read: 0b0: No capture completion action has occurred after the CMPF1 flag was set to 1 0b1: A capture completion action has occurred again after the CMPF1 flag was set to 1	RW
7	CMPF8	Count Match Flag 8	Write 0 to clear the flag, write 1 has no effect Read: 0b0: The value in the CMPAR8 register does not match the counter value, and no capture completion event has occurred 0b1: The value in the CMPAR8 register matches the counter value, or a capture completion event has occurred	RW
6	CMPF7	Count Match Flag 7	Write 0 to clear the flag, write 1 has no effect Read: 0b0: The value in the CMPAR7 register does not match the counter value, and no capture completion event has occurred 0b1: The value in the CMPAR7 register matches the counter value, or a capture completion event has occurred	RW
5	CMPF6	Count Match Flag 6	Write 0 to clear the flag, write 1 has no effect Read: 0b0: The value in the CMPAR6 register does not match the counter value, and no capture completion event has occurred 0b1: The value in the CMPAR6 register matches the counter value, or a capture completion event has occurred	RW
4	CMPF5	Count Match Flag 5	Write 0 to clear the flag, write 1 has no effect Read: 0b0: The value in the CMPAR5 register does not match the counter value, and no capture completion event has occurred 0b1: The value in the CMPAR5 register matches the counter value, or a capture completion event has occurred	RW

Bit	Name	Function	Description	Access
3	CMPF4	Count Match Flag 4	Write 0 to clear the flag, write 1 has no effect Read: 0b0: The value in the CMPAR4 register does not match the counter value, and no capture completion event has occurred 0b1: The value in the CMPAR4 register matches the counter value, or a capture completion event has occurred	RW
2	CMPF3	Count Match Flag 3	Write 0 to clear the flag, write 1 has no effect Read: 0b0: The value in the CMPAR3 register does not match the counter value, and no capture completion event has occurred 0b1: The value in the CMPAR3 register matches the counter value, or a capture completion event has occurred	RW
1	CMPF2	Count Match Flag 2	Write 0 to clear the flag, write 1 has no effect Read: 0b0: The value in the CMPAR2 register does not match the counter value, and no capture completion event has occurred 0b1: The value in the CMPAR2 register matches the counter value, or a capture completion event has occurred	RW
0	CMPF1	Count Match Flag 1	Write 0 to clear the flag, write 1 has no effect Read: 0b0: The value in the CMPAR1 register does not match the counter value, and no capture completion event has occurred 0b1: The value in the CMPAR1 register matches the counter value, or a capture completion event has occurred	RW

19.4.11 Buffer Control Register (TMRA_BCONRm) (m=1-4)

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													BSE1	BSE0	BEN
Access														RW	RW	RW

Bit	Name	Function	Description	Access
15:3	Reserved	Reserved	-	-
2	BSE1	Triangle Wave Buffer Transmission Selection 1	0b0: In triangle wave counting mode, the buffer value is not transmitted when the counter reaches the valley point 0b1: In triangle wave counting mode, the buffer value is transmitted when the counter reaches the valley point, that is: $CMPAR_{2m} \rightarrow CMPAR_{2m-1}$  Note The configuration of this bit is enable only when BEN is 1.	RW
1	BSE0	Triangle Wave Buffer Transmission Selection 0	0b0: In triangle wave counting mode, the buffer value is not transmitted when the counter reaches the peak point 0b1: In triangle wave counting mode, the buffer value is transmitted when the counter reaches the peak point, that is: $CMPAR_{2m} \rightarrow CMPAR_{2m-1}$  Note The configuration of this bit is enable only when BEN is 1.	RW
0	BEN	Buffer enabled	0b0: The buffer function for the $CMPAR_{2m-1}$ reference value is disabled 0b1: The buffer function for the $CMPAR_{2m-1}$ reference value is enabled	RW

 Note

- 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.

19.4.12 Capture Control Register (TMRA_CCONRm) (m=1-8)

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	0b00: HCLK 0b01: HCLK/4 0b10: HCLK/16 0b11: HCLK/64	RW
12	NOFIENCP	Capture Input Port Filtering CP	0b0: TMRA_PWMm port input filter function is disabled 0b1: TMRA_PWMm port input filter function is enabled	RW
11	HICP6	Capture Input Condition Enable 6	0b0: When the falling edge is sampled on the TMRA_XOR port, no capture input action occurs 0b1: When the falling edge is sampled on the TMRA_XOR port, a capture input action occurs  Note 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.	RW
10	HICP5	Capture Input Condition Enable 5	0b0: When the rising edge is sampled on the TMRA_XOR port, no capture input action occurs 0b1: When the rising edge is sampled on the TMRA_XOR port, a capture input action occurs  Note 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.	RW

Bit	Name	Function	Description	Access
9	HICP4	Capture Input Condition Enable 4	0b0: When the falling edge is sampled on the TMRA_TRIG port input, no capture input action occurs 0b1: When the falling edge is sampled on the TMRA_TRIG port input, a capture input action occurs  Note 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.	RW
8	HICP3	Capture Input Condition Enable 3	0b0: When the rising edge is sampled on the TMRA_TRIG port input, no capture input action occurs 0b1: When the rising edge is sampled on the TMRA_TRIG port input, a capture input action occurs  Note 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.	RW
7	<i>Reserved</i>	<i>Reserved</i>	-	-
6	HICP2	Capture Input Condition Enable 2	0b0: When the event specified in the TMRA_TRGSEL register occurs, no capture input action occurs 0b1: When the event specified in the TMRA_TRGSEL register occurs, a capture input action occurs  Note For more information, please refer to "TimerA Hardware Trigger Event Selection Register (TMRA_TRGSELx) (x=0-1) " in "Automatic Operation System (AOS) ".	RW
5	HICP1	Capture Input Condition Enable 1	0b0: When the falling edge is sampled on the TMRA_PWMm port input, no capture input action occurs 0b1: When the falling edge is sampled on the TMRA_PWMm port input, a capture input action occurs	RW

Bit	Name	Function	Description	Access
4	HICP0	Capture Input Condition Enable 0	0b0: When the rising edge is sampled on the TMRA_PWMm port input, no capture input action occurs 0b1: When the rising edge is sampled on the TMRA_PWMm port input, a capture input action occurs	RW
3:1	<i>Reserved</i>	<i>Reserved</i>	-	-
0	CAPMD	Function Mode Selection	0b0: Compare output function 0b1: Capture input function	RW

**Note**

- The TMRA_XOR signal is the signal obtained by performing an XOR operation on the quadrature inputs AIN and BIN of each unit.

19.4.13 Port Control Register (TMRA_PCONRm) (m=1-8)

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	0b0: Output disabled for TMRA_PWMm port in PWM output function mode 0b1: Output enabled for TMRA_PWMm port in PWM output function mode	RW
11:10	Reserved	Reserved	-	-
9:8	FORC	Forced Port Status Setting	0b0x: Automatically release PWM at the period point 0b10: TMRA_PWMm port output will be set to low level at the beginning of next period 0b11: TMRA_PWMm port output will be set to high level at the beginning of next period  Note 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. This bit can be used to achieve control of PWM output duty cycle 0% or 100%.	RW

Bit	Name	Function	Description	Access
7:6	PERC	Port Status Setting at Period Match	● 0b00: When the counter value equals the period value, the TMRA_PWMm output is set to low level. ● 0b01: When the counter value equals the period value, the TMRA_PWMm output is set to high level. ● 0b10: When the counter value equals the period value, the TMRA_PWMm output retains the previous state. ● 0b11: When the counter value equals the period value, the TMRA_PWMm output toggles.  Note 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.	RW
5:4	CMPC	Port Status Setting at Compare Match	● 0b00: When counter value equals CMPARn, the TMRA_PWMm output is set to low level. ● 0b01: When counter value equals CMPARn, the TMRA_PWMm output is set to high level. ● 0b10: When counter value equals CMPARn, the TMRA_PWMm output retains the previous state. ● 0b11: When counter value equals CMPARn, the TMRA_PWMm output toggles.	RW
3:2	STPC	Port Status Setting at Counting Stops	● 0b00: When counting stops, the TMRA_PWMm output is set to low level. ● 0b01: When counting stops, the TMRA_PWMm output is set to high level. ● 0b10: When counting stops, the TMRA_PWMm output retains the previous state. ● 0b11: When counting stops, the TMRA_PWMm output retains the previous state.	RW
1:0	STAC	Port Status Setting at Counting Starts	● 0b00: When counting starts, the TMRA_PWMm output is set to low level. ● 0b01: When counting starts, the TMRA_PWMm output is set to high level. ● 0b10: When counting starts, the TMRA_PWMm output retains the previous state. ● 0b11: When counting starts, the TMRA_PWMm output retains the previous state.	RW

19.4.14 Hardware Control Event Select 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	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

Bit	Name	Function	Description	Access
15	HCLE4	Hardware Clear Condition 4	Condition: Falling edge is sampled on TMRA_PWM3 port 0b0: When the condition is met, hardware clear is disabled 0b1: When the condition is met, hardware clear is enabled	RW
14	HCLE3	Hardware Clear Condition 3	Condition: Rising edge is sampled on TMRA_PWM3 port 0b0: When the condition is met, hardware clear is disabled 0b1: When the condition is met, hardware clear is enabled	RW
13:11	Reserved	Reserved	-	-
10	HCLE2	Hardware Clear Condition 2	Condition: The event specified in the TMRA_TRGSEL register occurs 0b0: When the condition is met, hardware clear is disabled 0b1: When the condition is met, hardware clear is enabled  Note For more information, please refer to "Time-rA Hardware Trigger Event Selection Register (TMRA_TRGSELx) (x=0-1) " in "Automatic Operation System (AOS) ".	RW
9	HCLE1	Hardware Clear Condition 1	Condition: Falling edge is sampled on TMRA_TRIG port 0b0: When the condition is met, hardware clear is disabled 0b1: When the condition is met, hardware clear is enabled	RW

Bit	Name	Function	Description	Access
8	HCLE0	Hardware Clear Condition 0	Condition: Rising edge is sampled on TMRA_TRIG port ● 0b0: When the condition is met, hardware clear is disabled ● 0b1: When the condition is met, hardware clear is enabled	RW
7	<i>Reserved</i>	<i>Reserved</i>	-	-
6	HSTP2	Hardware Stop Condition 2	Condition: The event specified in the TMRA_TRGSEL register occurs ● 0b0: When the condition is met, hardware stop is disabled ● 0b1: When the condition is met, hardware stop is enabled  Note For more information, please refer to "Time-rA Hardware Trigger Event Selection Register (TMRA_TRGSELx) (x=0-1) " in "Automatic Operation System (AOS) ".	RW
5	HSTP1	Hardware Stop Condition 1	Condition: Falling edge is sampled on TMRA_TRIG port ● 0b0: When the condition is met, hardware stop is disabled ● 0b1: When the condition is met, hardware stop is enabled	RW
4	HSTP0	Hardware Stop Condition 0	Condition: Rising edge is sampled on TMRA_TRIG port ● 0b0: When the condition is met, hardware stop is disabled ● 0b1: When the condition is met, hardware stop is enabled	RW
3	<i>Reserved</i>	<i>Reserved</i>	-	-
2	HSTA2	Hardware Start Condition 2	Condition: The event specified in the TMRA_TRGSEL register occurs ● 0b0: When the condition is met, hardware start is disabled ● 0b1: When the condition is met, hardware start is enabled  Note For more information, please refer to "Time-rA Hardware Trigger Event Selection Register (TMRA_TRGSELx) (x=0-1) " in "Automatic Operation System (AOS) ".	RW

Bit	Name	Function	Description	Access
1	HSTA1	Hardware Start Condition 1	Conditions: 1. Falling edge is sampled on TMRA_TRIG port ● 0b0: When the condition is met, hardware start is disabled ● 0b1: When the condition is met, hardware start is enabled	RW
0	HSTA0	Hardware Start Condition 0	Conditions: 1. Rising edge is sampled on TMRA_TRIG port ● 0b0: When the condition is met, hardware start is disabled ● 0b1: When the condition is met, hardware start is enabled	RW

19.4.15 Hardware Count Up Event Select 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					HCUP10	HCUP9	HCUP8	HCUP7	HCUP6	HCUP5	HCUP4	HCUP3	HCUP2	HCUP1	HCUP0
Access						RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	Function	Description	Access
15:11	Reserved	Reserved	-	-
10	HCUP10	Hardware Count Up Condition 10	Condition: The event specified in the TMRA_TRGSEL register occurs 0b0: When this condition is met, hardware count up is disabled. 0b1: When this condition is met, hardware count up is enabled.  Note For more information, please refer to "Time-rA Hardware Trigger Event Selection Register (TMRA_TRGSELx) (x=0-1) " in "Automatic Operation System (AOS) ".	RW
9	HCUP9	Hardware Count Up Condition 9	Condition: Falling edge is sampled on the TMRA_TRIG port 0b0: When this condition is met, hardware count up is disabled. 0b1: When this condition is met, hardware count up is enabled.	RW
8	HCUP8	Hardware Count Up Condition 8	Condition: Rising edge is sampled on the TMRA_TRIG port 0b0: When this condition is met, hardware count up is disabled. 0b1: When this condition is met, hardware count up is enabled.	RW
7	HCUP7	Hardware Count Up Condition 7	Condition: When the TMRA_CLKB port is high level, the falling edge is sampled on the TMRA_CLKA port 0b0: When this condition is met, hardware count up is disabled. 0b1: When this condition is met, hardware count up is enabled.	RW

Bit	Name	Function	Description	Access
6	HCUP6	Hardware Count Up Condition 6	Condition: When the TMRA_CLKB port is high level, the rising edge is sampled on the TMRA_CLKA port 0b0: When this condition is met, hardware count up is disabled. 0b1: When this condition is met, hardware count up is enabled.	RW
5	HCUP5	Hardware Count Up Condition 5	Condition: When the TMRA_CLKB port is low level, the falling edge is sampled on the TMRA_CLKA port 0b0: When this condition is met, hardware count up is disabled. 0b1: When this condition is met, hardware count up is enabled.	RW
4	HCUP4	Hardware Count Up Condition 4	Condition: When the TMRA_CLKB port is low level, the rising edge is sampled on the TMRA_CLKA port 0b0: When this condition is met, hardware count up is disabled. 0b1: When this condition is met, hardware count up is enabled.	RW
3	HCUP3	Hardware Count Up Condition 3	Condition: When the TMRA_CLKA port is high level, the falling edge is sampled on the TMRA_CLKB port 0b0: When this condition is met, hardware count up is disabled. 0b1: When this condition is met, hardware count up is enabled.	RW
2	HCUP2	Hardware Count Up Condition 2	Condition: When the TMRA_CLKA port is high level, the rising edge is sampled on the TMRA_CLKB port 0b0: When this condition is met, hardware count up is disabled. 0b1: When this condition is met, hardware count up is enabled.	RW
1	HCUP1	Hardware Count Up Condition 1	Condition: When the TMRA_CLKA port is low level, the falling edge is sampled on the TMRA_CLKB port 0b0: When this condition is met, hardware count up is disabled. 0b1: When this condition is met, hardware count up is enabled.	RW
0	HCUP0	Hardware Count Up Condition 0	Condition: When the TMRA_CLKA port is low level, the rising edge is sampled on the TMRA_CLKB port 0b0: When this condition is met, hardware count up is disabled. 0b1: When this condition is met, hardware count up is enabled.	RW

19.4.16 Hardware Count Down Event Select 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					HCDO10	HCDO9	HCDO8	HCDO7	HCDO6	HCDO5	HCDO4	HCDO3	HCDO2	HCDO1	HCDO0
Access						RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	Function	Description	Access
15:11	Reserved	Reserved	-	-
10	HCDO10	Hardware Count Down Condition 10	Condition: The event specified in the TMRA_TRGSEL register occurs 0b0: Hardware count down is disabled when the condition is met. 0b1: Hardware count down is enabled when the condition is met.  Note For more information, please refer to "Time-rA Hardware Trigger Event Selection Register (TMRA_TRGSELx) (x=0-1) " in "Automatic Operation System (AOS) ".	RW
9	HCDO9	Hardware Count Down Condition 9	Condition: Falling edge is sampled on the TMRA_TRIG port 0b0: Hardware count down is disabled when the condition is met. 0b1: Hardware count down is enabled when the condition is met.	RW
8	HCDO8	Hardware Count Down Condition 8	Condition: Rising edge is sampled on the TMRA_TRIG port 0b0: Hardware count down is disabled when the condition is met. 0b1: Hardware count down is enabled when the condition is met.	RW
7	HCDO7	Hardware Count Down Condition 7	Condition: When the TMRA_CLKB port is high level, the falling edge is sampled on the TMRA_CLKA port 0b0: Hardware count down is disabled when the condition is met. 0b1: Hardware count down is enabled when the condition is met.	RW

Bit	Name	Function	Description	Access
6	HCDO6	Hardware Count Down Condition 6	Condition: When the TMRA_CLKB port is high level, the rising edge is sampled on the TMRA_CLKA port ● 0b0: Hardware count down is disabled when the condition is met. ● 0b1: Hardware count down is enabled when the condition is met.	RW
5	HCDO5	Hardware Count Down Condition 5	Condition: When the TMRA_CLKB port is low level, the falling edge is sampled on the TMRA_CLKA port ● 0b0: Hardware count down is disabled when the condition is met. ● 0b1: Hardware count down is enabled when the condition is met.	RW
4	HCDO4	Hardware Count Down Condition 4	Condition: When the TMRA_CLKB port is low level, the rising edge is sampled on the TMRA_CLKA port ● 0b0: Hardware count down is disabled when the condition is met. ● 0b1: Hardware count down is enabled when the condition is met.	RW
3	HCDO3	Hardware Count Down Condition 3	Condition: When the TMRA_CLKA port is high level, the falling edge is sampled on the TMRA_CLKB port ● 0b0: Hardware count down is disabled when the condition is met. ● 0b1: Hardware count down is enabled when the condition is met.	RW
2	HCDO2	Hardware Count Down Condition 2	Condition: When the TMRA_CLKA port is high level, the rising edge is sampled on the TMRA_CLKB port ● 0b0: Hardware count down is disabled when the condition is met. ● 0b1: Hardware count down is enabled when the condition is met.	RW
1	HCDO1	Hardware Count Down Condition 1	Condition: When the TMRA_CLKA port is low level, the falling edge is sampled on the TMRA_CLKB port ● 0b0: Hardware count down is disabled when the condition is met. ● 0b1: Hardware count down is enabled when the condition is met.	RW
0	HCDO0	Hardware Count Down Condition 0	Condition: When the TMRA_CLKA port is low level, the rising edge is sampled on the TMRA_CLKB port ● 0b0: Hardware count down is disabled when the condition is met. ● 0b1: Hardware count down is enabled when the condition is met.	RW

19.4.17 Counter Synchronization Enable Register (TMR_SYNENR)

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																														TMR4		TMRA
Access																															RW		RW

Bit	Name	Function	Description	Access
31:2	Reserved	Reserved	-	-
1	TMR4	TMR4 Software Start Enable Bit	0b0: TMR4 software start is disabled 0b1: TMR4 software start is enabled, and the timer starts counting	RW
0	TMRA	TMRA Software Start Enable Bit	0b0: TMRA software start is disabled 0b1: TMRA software start is enabled, and the timer starts counting	RW

20 General Timer (Timer2)

20.1 Introduction

General Timer 2 (Timer2) is a basic timer with a 32/16-bit counting width that can implement synchronous and asynchronous counting modes. This timer contains 2 channels (CH-A and CH-B). Each channel has an output port, enabling basic square wave output; each channel has 2 input ports: one is a clock input port, enabling port asynchronous counting; the other is a trigger input port, enabling timer start, stop, clear, counting actions, and count value capture input; it supports pulse width measurement and period measurement; it supports free-running counting and compare stop counting. This product series incorporates 2 units of Timer2, where unit 1 has a 16-bit counting width and unit 2 has a 32-bit counting width.

20.2 Features

The basic functions and features of Timer2 are as follows:

Table 20-1 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

20.3 Functional Description

20.3.1 Basic Block Diagram

In the block diagram, the symbol "<t>" denotes the unit number, which ranges from 1-2. 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).

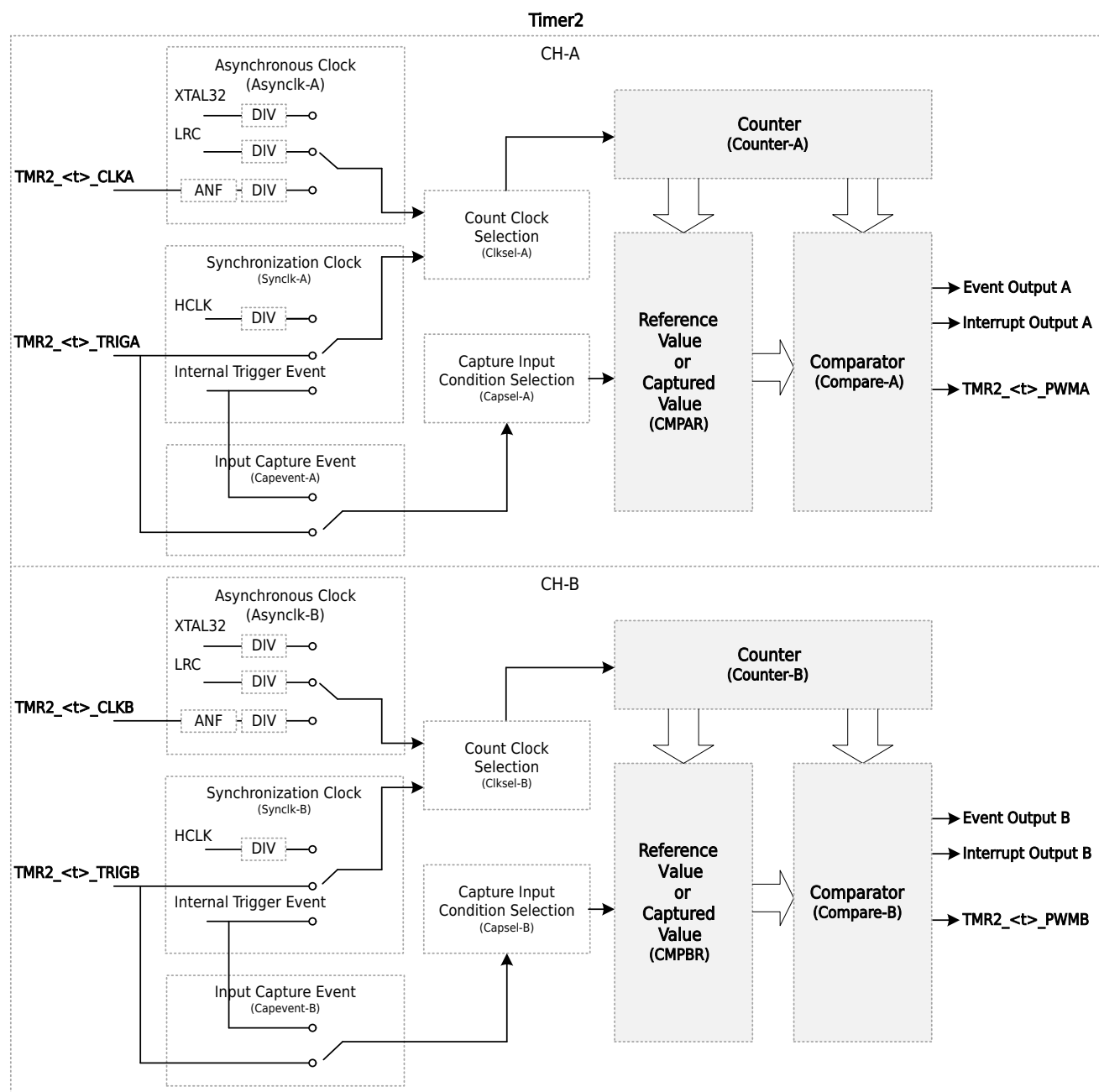


Figure 20-1 Timer2 Basic Block Diagram

The following table lists the input and output ports of Timer2.

Table 20-2 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 2. Capture input port
TMR2_<t>_PWMA/B	Output	Square wave output port

20.3.2 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=0), the following clock sources can be selected (configured via BCONR.SYNCLKA[1:0] or BCONR.SYNCLKAT[1:0]) :

1. HCLK and its 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024 divisions are used as synchronous count clock (BCONR.SYNCLKA[1:0]=0b00 & BCONR.CKDIVA[3:0] settings);
2. The input enable selection edge of TMR2_<t>_TRIGA and TMR2_<t>_TRIGB ports is sampled by the internal HCLK as the synchronous counting clock (BCONR.SYNCLKA[1:0]=0b01 or 0b10);
3. An internal hardware trigger event input can serve as the synchronous count clock (BCONR.SYNCLKA[1:0]=0b11);
4. A counting compare match event occurs on the TMR2_CHA/B channel (BCONR.SYNCLKAT=1).

Asynchronous Count Timer

In asynchronous counting mode (BCONR.SYNSA=1), the clock source can be selected from the following options (BCONR.ASYNCLKA[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=0 & BCONR.CKDIVA[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=1 & BCONR.CKDIVA[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=0b10 & BCONR.CKDIVA[3:0] settings).

20.3.3 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[1:0], PCONR.STPCA[1:0], and PCONR.CMPCA[1:0]. The figure below illustrates an example of the compare output behavior.

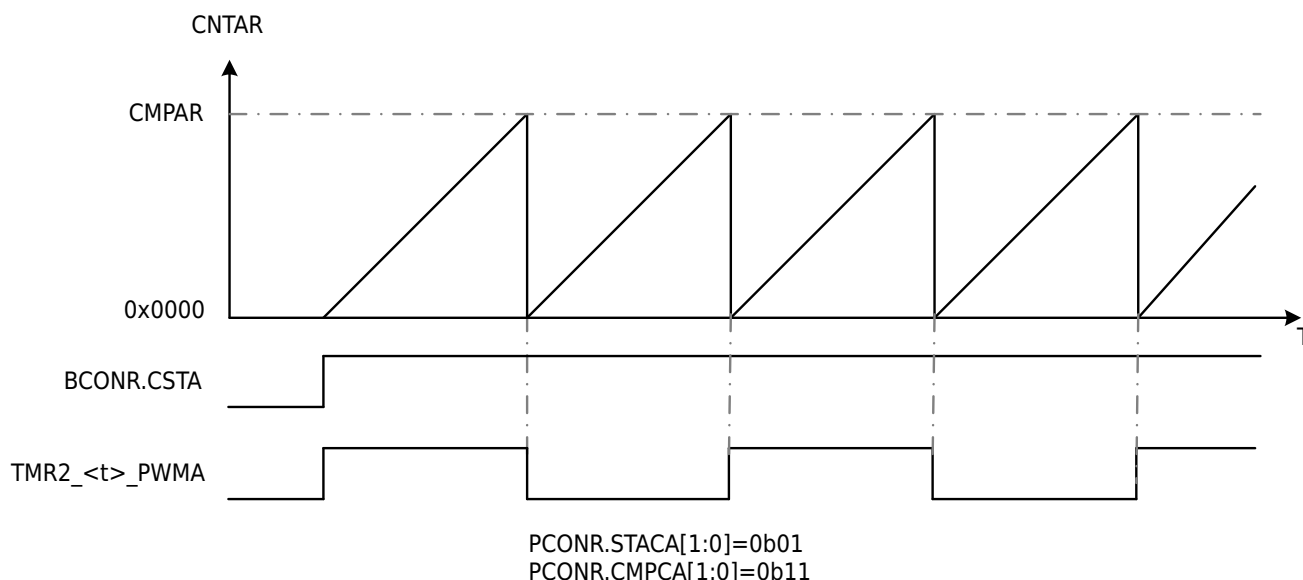


Figure 20-2 Compare Output Action

20.3.4 Hardware Triggering

20.3.4.1 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_TRGSELx) (x=0-1). 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 register (CMU_FCG).

20.3.4.2 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.

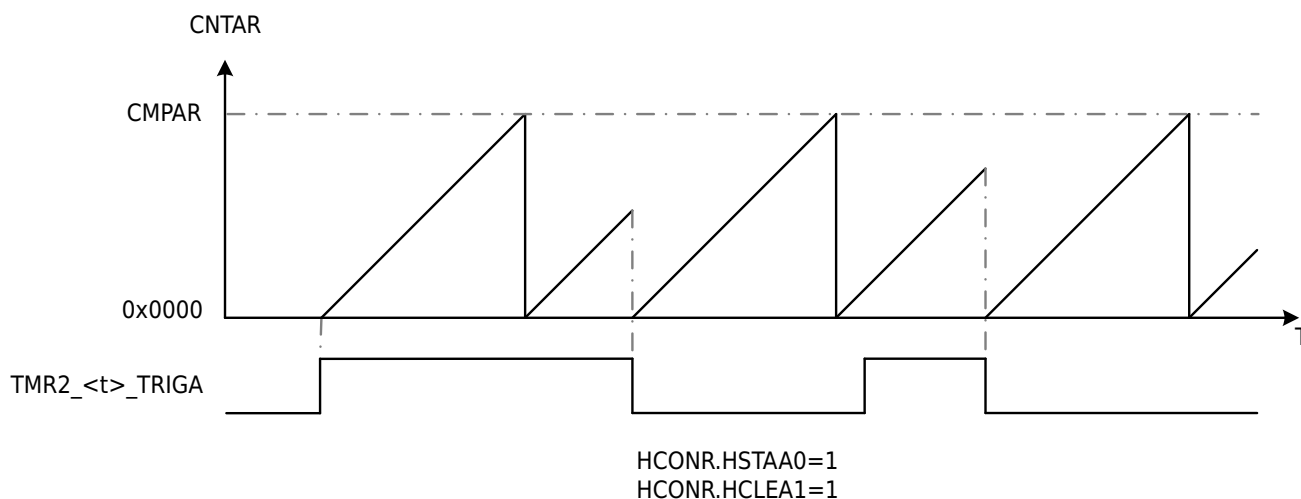


Figure 20-3 Hardware Start and Clear Action

20.3.4.3 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.

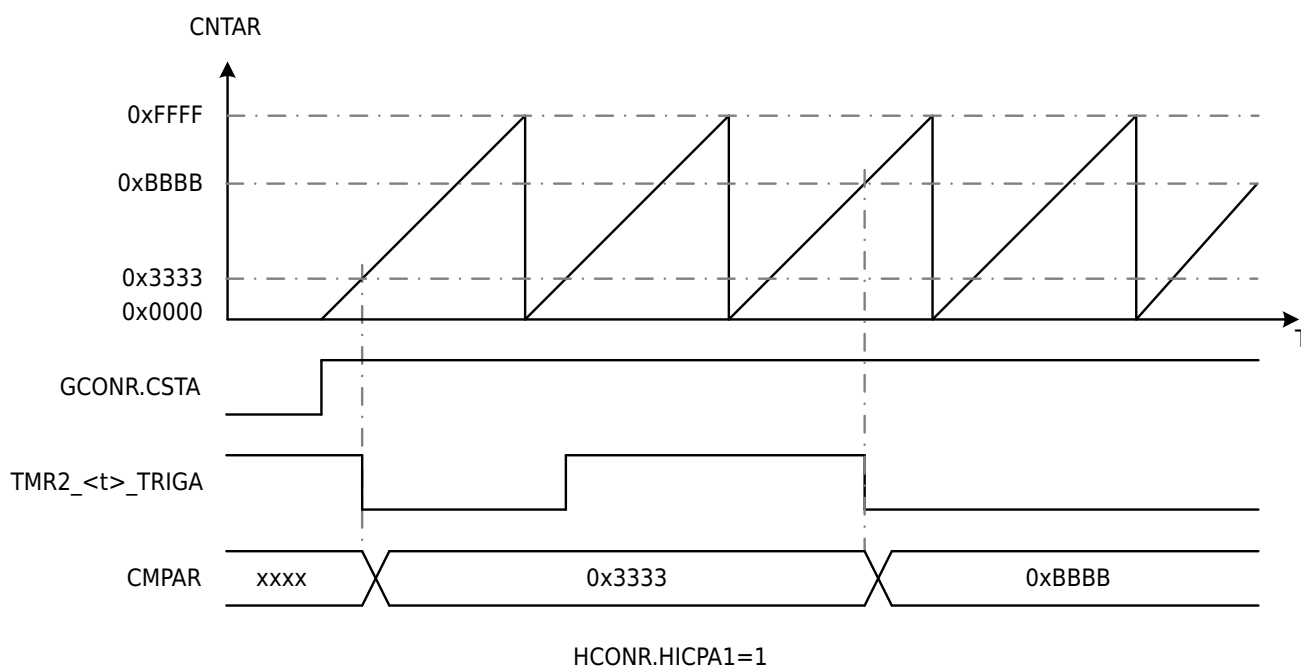


Figure 20-4 Capture Input Action

20.3.4.4 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.

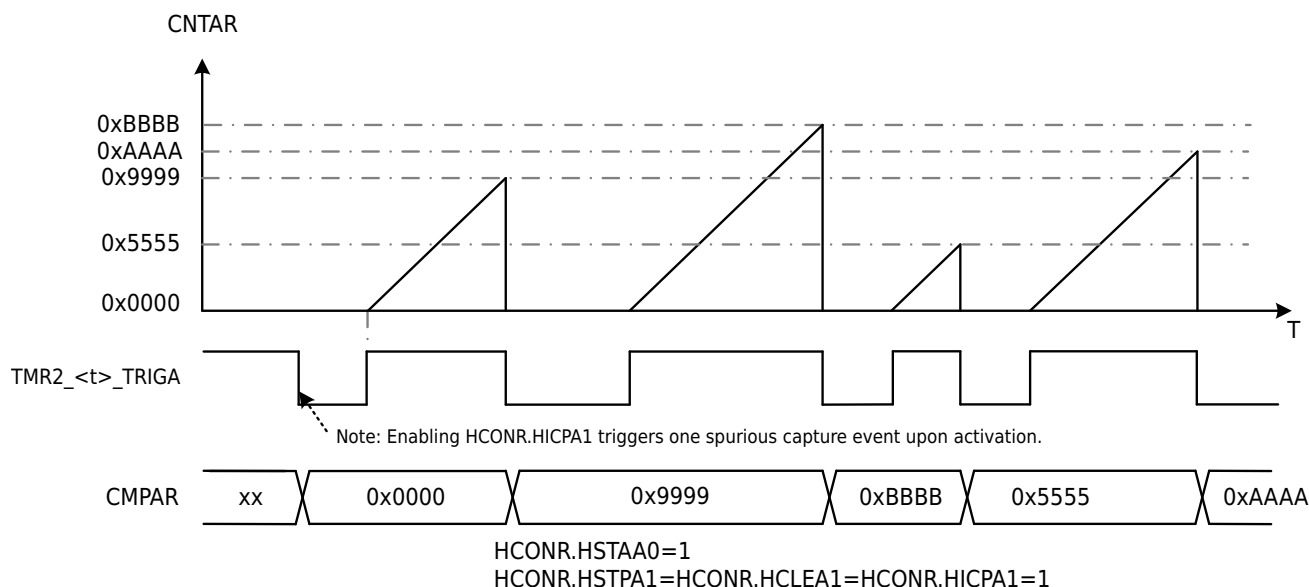


Figure 20-5 Pulse Width Measurement

20.3.4.5 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.

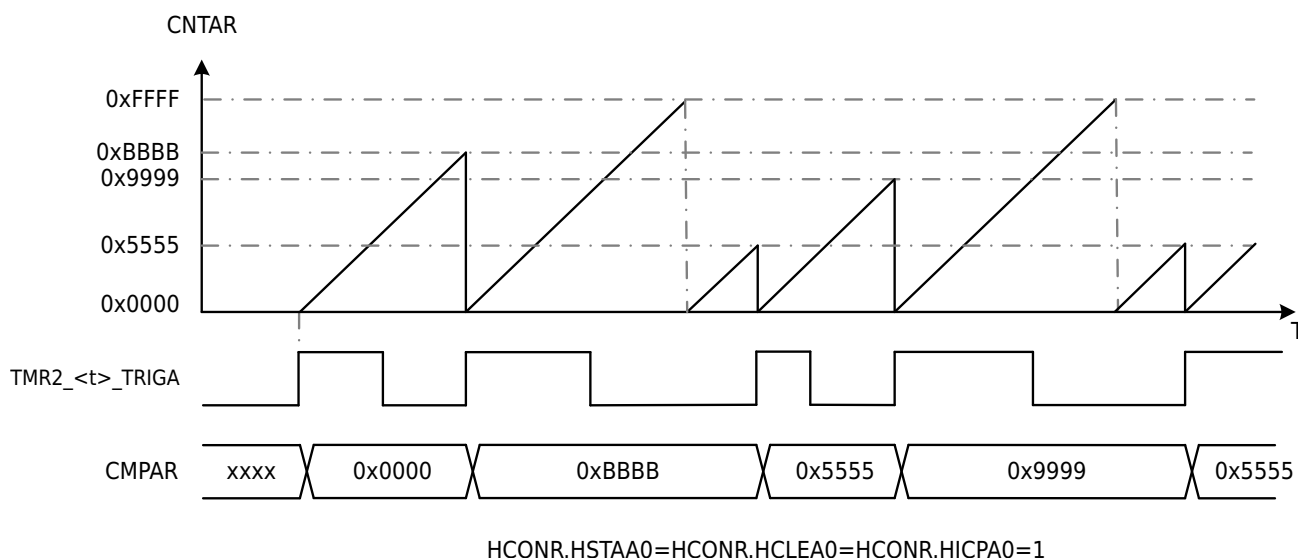


Figure 20-6 Period Measurement

20.3.5 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 register. The reference clock for filtering is also set by the port control register (PCONR.NOFICKA[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.

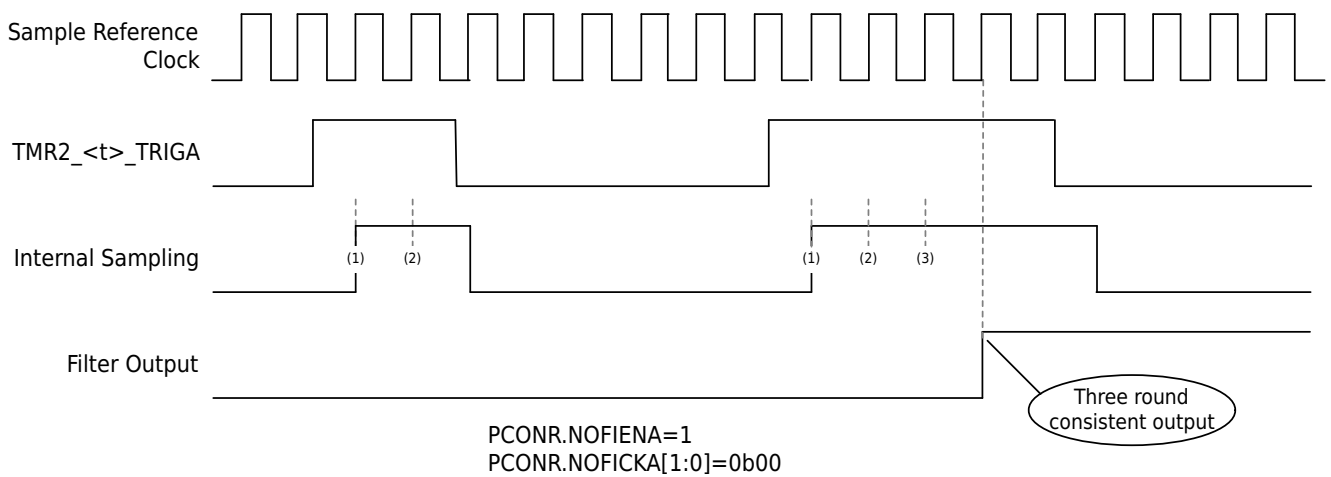


Figure 20-7 Digital Filtering of TRIG Input Port

20.3.6 Interrupt and Events

20.3.6.1 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 will be set to 1. At this time, if the ICONR.CMENA 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 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/0xFFFFFFFF, a count overflow event occurs. At this time, the STFLR.OVFA will be set to 1. If the ICONR.OVENA 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 and overflow 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.

20.3.6.2 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.

20.4 Register Description

20.4.1 Register List

- Timer2_1 base address: 0x40005C00
- Timer2_2 base address: 0x40006000



Note

Timer2 is the name of the timer, and Timer2_<t> (<t>:1-2) is the timer unit.

Table 20-3 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

20.4.2 Count Value Register m (TMR2_CNTmR) (m=A/B)

Unit 1:

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  Note Writing to this register is prohibited while counting is in progress. This register must be initialized before the timer can be started again after counting stops.	RW

Unit 2:

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	CNT																																					
Access	RW																																					

Bit	Name	Function	Description	Access
31:0	CNT	Count Value	Current timer count value  Note Writing to this register is prohibited while counting is in progress. This register must be initialized before the timer can be started again after counting stops.	RW

20.4.3 Reference Value Register m (TMR2_CMPmR) (m=A/B)

Unit 1:

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																																

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

Unit 2:

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	0xFFFFFFFF																															
Name	CMP																															
Access	RW																															

Bit	Name	Function	Description	Access
31:0	CMP	Reference Value	Set the counting reference value to generate a compare match event	RW

20.4.4 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	CMPBSTP	Reserved	SYNCLKBT	ASYNCLKB	SYNCLKB		CKDIVB	SYNSB	Reserved	CAPMDB	CSTB	CMFASTP	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		RW		RW		RW		RW		RW

Bit	Name	Function	Description	Access
31	CMPBSTP	Count Compare Stop Control B	0b0: The counter continues counting after a count match occurs 0b1: The counter stops counting after a count match occurs  Note At this time, the TMR2_CNTBR value is 0 and the CSTB bit is 0.	RW
30:29	Reserved	Reserved	-	-
28	SYNCLKBT	Synchronous Counting and Clock Source Selection BT	Condition: Count compare match occurs in channel A of Timer2 unit n 0b0: When the condition matches, the synchronous counting in channel B of Timer2 unit m is disabled 0b1: When the condition matches, channel B of Timer2 unit m performs one synchronous count (When m=1, 2, then n=2, 1)	RW
27:26	ASYNCLKB	Asynchronous Counting and Clock Source Selection B	0b00: LRC 0b01: XTAL32 0b10: TMR2_<t>_CLKB clock input 0b11: Setting prohibited	RW
25:24	SYNCLKB	Synchronous Counting and Clock Source Selection B	0b00: HCLK 0b01: TMR2_<t>_TRIGB rising edge (internal HCLK synchronization) 0b10: TMR2_<t>_TRIGB falling edge (internal HCLK synchronization) 0b11: Internal trigger event input	RW
23:20	CKDIVB	Count Clock Frequency Division Selection B	0b0000: Clock source 0b0001: Clock source/2 0b0010: Clock source/4 0b0011: Clock source/8 0b0100: Clock source/16 0b0101: Clock source/32 0b0110: Clock source/64	RW

Bit	Name	Function	Description	Access
23:20	CKDIVB	Count Clock Frequency Division Selection B	0b0111: Clock source/128 0b1000: Clock source/256 0b1001: Clock source/512 0b1010: Clock source/1024 - Others: Setting prohibited  Note The divided clock source can be various clock sources for asynchronous counting or HCLK for synchronous counting.	RW
19	SYNSB	Counting mode selection B	0b0: Synchronous counting mode 0b1: Asynchronous counting mode	RW
18	<i>Reserved</i>	<i>Reserved</i>	-	-
17	CAPMDB	Functional Mode Selection B	0b0: Compare output function 0b1: Capture input function	RW
16	CSTB	Timer Start B	0b0: Channel B timer stop 0b1: Channel B timer start  Note This bit will automatically be set to 0 when the hardware trigger stop condition is enable.	RW
15	CMPASTP	Count Compare Stop Control A	0b0: The counter continues counting after a count match occurs 0b1: The counter stops counting after a count match occurs  Note At this time, the TMR2_CNTAR value is 0, and the CSTA bit is 0.	RW
14:13	<i>Reserved</i>	<i>Reserved</i>	-	-
12	SYNCLKAT	Synchronous Counting and Clock Source Selection AT	Condition: Count compare match occurs in channel B of Timer2 unit n 0b0: When the condition matches, the synchronous counting in channel A of Timer2 unit m is disabled 0b1: When the condition matches, channel A of Timer2 unit m performs one synchronous counting (When m=1, 2, then n=2, 1)	RW
11:10	ASYNCLKA	Asynchronous Counting and Clock Source Selection A	0b00: LRC 0b01: XTAL32 0b10: TMR2_<t>_CLKA clock input 0b11: Setting prohibited	RW

Bit	Name	Function	Description	Access
9:8	SYNCLKA	Synchronous Counting and Clock Source Selection A	● 0b00: HCLK ● 0b01: TMR2_<t>TRIGA rising edge (internally synchronized with HCLK) ● 0b10: TMR2<t>_TRIGA falling edge (internally synchronized with HCLK) ● 0b11: Internal trigger event input	RW
7:4	CKDIVA	Count Clock Frequency Division Selection A	● 0b0000: Clock source ● 0b0001: Clock source/2 ● 0b0010: Clock source/4 ● 0b0011: Clock source/8 ● 0b0100: Clock source/16 ● 0b0101: Clock source/32 ● 0b0110: Clock source/64 ● 0b0111: Clock source/128 ● 0b1000: Clock source/256 ● 0b1001: Clock source/512 ● 0b1010: Clock source/1024 ● Others: Setting prohibited  Note The divided clock source can be various clock sources for asynchronous counting or HCLK for synchronous counting.	RW
3	SYNSA	Counting Mode Selection A	● 0b0: Synchronous counting mode ● 0b1: Asynchronous counting mode	RW
2	<i>Reserved</i>	<i>Reserved</i>	-	-
1	CAPMDA	Pattern Mode Selection A	● 0b0: Compare output function ● 0b1: Capture input function	RW
0	CSTA	Timer Start A	● 0b0: Channel A timer stop ● 0b1: Channel A timer start  Note This bit will automatically be set to 0 when the hardware trigger stop condition is enable.	RW

20.4.5 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/0xFFFFFFFF 0b0: Disable interrupt 0b1: Enable interrupt	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 0b0: Disable interrupt 0b1: Enable interrupt	RW
15:2	Reserved	Reserved	-	-
1	OVENA	Count Overflow Interrupt Enable A	Interrupt generation condition: when the count value (CNTAR)=0xFFFF/0xFFFFFFFF 0b0: Disable interrupt 0b1: Enable interrupt	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 0b0: Disable interrupt 0b1: Enable interrupt	RW

20.4.6 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	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			
Reset	0x00000000																																		
Name	Reserved	NOFICKB		NOFIENB		Reserved		OUTENB		Reserved		CMPCB		STPCB		STACB		Reserved		NOFICKA		NOFIENA		Reserved		OUTENA		Reserved		CMPCA		STPCA		STACA	
Access		RW		RW				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	0b00: Clock source 0b01: Clock source/4 0b10: Clock source/16 0b11: Clock source/64  Note The clock source is various clock sources for asynchronous counting and HCLK for synchronous counting.	RW
28	NOFIENB	Trigger Port Filter B	0b0: TMR2_<t>_TRIGB input port filter function is disabled 0b1: TMR2_<t>_TRIGB input port filter function is enabled	RW
27:25	Reserved	Reserved	-	-
24	OUTENB	Port Output Enable B	0b0: TMR2_<t>_PWMB port output disabled 0b1: TMR2_<t>_PWMB port output enabled	RW
23:22	Reserved	Reserved	-	-
21:20	CMPCB	Port Status Setting B when Comparison Values Match	0b00: When the count value (CNTBR) is equal to CMPBR, the TMR2_<t>_PWMB port output is set to low level 0b01: When the count value (CNTBR) is equal to CMPBR, the TMR2_<t>_PWMB port output is set to high level 0b10: When the count value (CNTBR) is equal to CMPBR, the TMR2_<t>_PWMB port output retains the previous state 0b11: When the count value (CNTBR) is equal to CMPBR, the TMR2_<t>_PWMB port output is set to inverted level	RW

Bit	Name	Function	Description	Access
19:18	STPCB	Port Status Setting B when Counting Stops	0b00: When the counter stops, the TMR2_<t>_PWMB port output is set to low level 0b01: When the counter stops, the TMR2_<t>_PWMB port output is set to high level 0b10: When the counter stops, the TMR2_<t>_PWMB port output retains the previous state 0b11: When the counter stops, the TMR2_<t>_PWMB port output retains the previous state	RW
17:16	STACB	Port Status Setting B when Counting Starts	0b00: When counting starts, TMR2_<t>_PWMB port output is set to low level 0b01: When counting starts, TMR2_<t>_PWMB port output is set to high level 0b10: When counting starts, TMR2_<t>_PWMB port output retains the previous state 0b11: When counting starts, TMR2_<t>_PWMB port output retains the previous state	RW
15	Reserved	Reserved	-	-
14:13	NOFICKA	Filtering Sampling Reference Clock Selection A	0b00: Clock source 0b01: Clock source/4 0b10: Clock source/16 0b11: Clock source/64  Note The clock source is various clock sources for asynchronous counting and HCLK for synchronous counting.	RW
12	NOFIENA	Trigger Port Filter A	0b0: TMR2_<t>_TRIGA input port filter function disabled 0b1: TMR2_<t>_TRIGA input port filter function enabled	RW
11:9	Reserved	Reserved	-	-
8	OUTENA	Port Output Enable A	0b0: TMR2_<t>_PWMA port output disabled 0b1: TMR2_<t>_PWMA port output enabled	RW
7:6	Reserved	Reserved	-	-

Bit	Name	Function	Description	Access
5:4	CMPCA	Port Status Setting A when Comparison Values Match	● 0b00: When the count value (CNTAR) is equal to CMPAR, the TMR2_<t>_PWMA port output is set to low level ● 0b01: When the count value (CNTAR) is equal to CMPAR, the TMR2_<t>_PWMA port output is set to high level ● 0b10: When the count value (CNTAR) is equal to CMPAR, the TMR2_<t>_PWMA port output retains the previous state ● 0b11: When the count value (CNTAR) is equal to CMPAR, the TMR2_<t>_PWMA port output is set to inverted level	RW
3:2	STPCA	Port Status Setting A when Counting Stops	● 0b00: When counting stops, the TMR2_<t>_PWMA port output is set to low level ● 0b01: When counting stops, the TMR2_<t>_PWMA port output is set to high level ● 0b10: When counting stops, the TMR2_<t>_PWMA port output retains the previous state ● 0b11: When counting stops, the TMR2_<t>_PWMA port output retains the previous state	RW
1:0	STACA	Port Status Setting A when Counting Starts	● 0b00: When counting starts, TMR2_<t>_PWMA port output is set to low level ● 0b01: When counting starts, TMR2_<t>_PWMA port output is set to high level ● 0b10: When counting starts, TMR2_<t>_PWMA port output retains the previous state ● 0b11: When counting starts, TMR2_<t>_PWMA port output retains the previous state	RW

20.4.7 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 Enable 0b0: Hardware capture input is disabled when the condition is met 0b1: Hardware capture input is enabled when the condition is met	RW
29	HICPB1	Hardware Capture Input Condition B1	Condition: TMR2_<t>_TRIGB port is sampled to the falling edge 0b0: Hardware capture input is disabled when the condition is met 0b1: Hardware capture input is enabled when the condition is met	RW
28	HICPB0	Hardware Capture Input Condition B0	Condition: TMR2_<t>_TRIGB port is sampled to the rising edge 0b0: Hardware capture input is disabled when the condition is met 0b1: Hardware capture input is enabled when the condition is met	RW
27	Reserved	Reserved	-	-
26	HCLEB2	Hardware Clear Condition B2	Condition: Internal Hardware Trigger Event Enable 0b0: Hardware clear is disabled when the condition is met 0b1: Hardware clear is enabled when the condition is met	RW
25	HCLEB1	Hardware Clear Condition B1	Condition: TMR2_<t>_TRIGB port is sampled to the falling edge 0b0: Hardware clear is disabled when the condition is met 0b1: Hardware clear is enabled when the condition is met	RW

Bit	Name	Function	Description	Access
24	HCLEB0	Hardware Clear Condition B0	Condition: TMR2_<t>_TRIGB port is sampled to the rising edge ● 0b0: Hardware clear is disabled when the condition is met ● 0b1: Hardware clear is enabled when the condition is met	RW
23	<i>Reserved</i>	<i>Reserved</i>	-	-
22	HSTPB2	Hardware Stop Condition B2	Condition: Internal Hardware Trigger Event Enable ● 0b0: Hardware stop is disabled when the condition is met ● 0b1: Hardware stop is enabled when the condition is met	RW
21	HSTPB1	Hardware Stop Condition B1	Condition: TMR2_<t>_TRIGB port is sampled to the falling edge ● 0b0: Hardware stop is disabled when the condition is met ● 0b1: Hardware stop is enabled when the condition is met	RW
20	HSTPB0	Hardware Stop Condition B0	Condition: TMR2_<t>_TRIGB port is sampled to the rising edge ● 0b0: Hardware stop is disabled when the condition is met ● 0b1: Hardware stop is enabled when the condition is met	RW
19	<i>Reserved</i>	<i>Reserved</i>	-	-
18	HSTAB2	Hardware Start Condition B2	Condition: Internal Hardware Trigger Event Enable ● 0b0: Hardware start is disabled when the condition is met ● 0b1: Hardware start is enabled when the condition is met	RW
17	HSTAB1	Hardware Start Condition B1	Condition: TMR2_<t>_TRIGB port is sampled to the falling edge ● 0b0: Hardware start is disabled when the condition is met ● 0b1: Hardware start is enabled when the condition is met	RW
16	HSTAB0	Hardware Start Condition B0	Condition: TMR2_<t>_TRIGB port is sampled to the rising edge ● 0b0: Hardware start is disabled when the condition is met ● 0b1: Hardware start is enabled when the condition is met	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 Enable 0b0: Hardware capture input is disabled when the condition is met 0b1: Hardware capture input is enabled when the condition is met	RW
13	HICPA1	Hardware Capture Input Condition A1	Condition: TMR2_<t>_TRIGA port is sampled to the falling edge 0b0: Hardware capture input is disabled when the condition is met 0b1: Hardware capture input is enabled when the condition is met	RW
12	HICPA0	Hardware Capture Input Condition A0	Condition: TMR2_<t>_TRIGA port is sampled to the rising edge 0b0: Hardware capture input is disabled when the condition is met 0b1: Hardware capture input is enabled when the condition is met	RW
11	<i>Reserved</i>	<i>Reserved</i>	-	-
10	HCLEA2	Hardware Clear Condition A2	Condition: Internal Hardware Trigger Event Enable 0b0: Hardware clear is disabled when the condition is met 0b1: Hardware clear is enabled when the condition is met	RW
9	HCLEA1	Hardware Clear Condition A1	Condition: TMR2_<t>_TRIGA port is sampled to the falling edge 0b0: Hardware clear is disabled when the condition is met 0b1: Hardware clear is enabled when the condition is met	RW
8	HCLEA0	Hardware Clear Condition A0	Condition: TMR2_<t>_TRIGA port is sampled to the rising edge 0b0: Hardware clear is disabled when the condition is met 0b1: Hardware clear is enabled when the condition is met	RW
7	<i>Reserved</i>	<i>Reserved</i>	-	-
6	HSTPA2	Hardware Stop Condition A2	Condition: Internal Hardware Trigger Event Enable 0b0: Hardware stop is disabled when the condition is met 0b1: Hardware stop is enabled when the condition is met	RW

Bit	Name	Function	Description	Access
5	HSTPA1	Hardware Stop Condition A1	Condition: TMR2_<t>_TRIGA port is sampled to the falling edge ● 0b0: Hardware stop is disabled when the condition is met ● 0b1: Hardware stop is enabled when the condition is met	RW
4	HSTPA0	Hardware Stop Condition A0	Condition: TMR2_<t>_TRIGA port is sampled to the rising edge ● 0b0: Hardware stop is disabled when the condition is met ● 0b1: Hardware stop is enabled when the condition is met	RW
3	<i>Reserved</i>	<i>Reserved</i>	-	-
2	HSTAA2	Hardware Start Condition A2	Condition: Internal Hardware Trigger Event Enable ● 0b0: Hardware start is disabled when the condition is met ● 0b1: Hardware start is enabled when the condition is met	RW
1	HSTAA1	Hardware Start Condition A1	Condition: TMR2_<t>_TRIGA port is sampled to the falling edge ● 0b0: Hardware start is disabled when the condition is met ● 0b1: Hardware start is enabled when the condition is met	RW
0	HSTAA0	Hardware Start Condition A0	Condition: TMR2_<t>_TRIGA port is sampled to the rising edge ● 0b0: Hardware start is disabled when the condition is met ● 0b1: Hardware start is enabled when the condition is met	RW

20.4.8 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								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved								Reserved						

Bit	Name	Function	Description	Access
31:25	Reserved	Reserved	-	-
24	SYDB	Synchronization Completion Flag B	0b0: B channel register is synchronizing during asynchronous counting 0b1: B channel register synchronization is completed during asynchronous counting	RO
23:19	Reserved	Reserved	-	-
18	ICPB	Repeat Capture Flag B	0b0: No capture input occurred after the count match flag bit CMFB was set to 1 0b1: A capture input occurred again after the count match flag bit CMFB was set to 1	RW
17	OVFB	Count Overflow Flag B	0b0: The count value (CNTBR) does not count to 0xFFFF/0xFFFFFFFF 0b1: The count value (CNTBR) counts to 0xFFFF/0xFFFFFFFF	RW
16	CMFB	Count Match Flag B	0b0: The value of the CMPBR register is not equal to the count value (CNTBR) and no capture input action occurs 0b1: The value of the CMPBR register is equal to the count value (CNTBR) or a capture input action occurs	RW
15:9	Reserved	Reserved	-	-
8	SYDA	Synchronization Completion Flag A	0b0: A channel register is synchronizing during asynchronous counting 0b1: A channel register synchronization is completed during asynchronous counting	RO
7:3	Reserved	Reserved	-	-
2	ICPA	Repeat Capture Flag A	0b0: No capture input occurred after the count match flag bit CMFA was set to 1 0b1: A capture input occurred again after the count match flag bit CMFA was set to 1	RW

Bit	Name	Function	Description	Access
1	OVFA	Count Overflow Flag A	● 0b0: The count value (CNTAR) does not count to 0xFFFF/0xFFFFFFFF ● 0b1: The count value (CNTAR) counts to 0xFFFF/0xFFFFFFFF	RW
0	CMFA	Count Match Flag A	● 0b0: The value of the CMPAR register is not equal to the count value (CNTAR) and no capture input action occurs ● 0b1: The value of the CMPAR register is equal to the count value (CNTAR) or a capture input action occurs	RW

20.5 Precautions

1. During asynchronous counting operation, you must first configure the asynchronous clock source by setting the BCONR.ASYNCLKA bit, then select the asynchronous counting mode by setting the BCONR.SYNSA bit, and finally start Timer2.
2. Under asynchronous counting mode, when CNTAR, CMPAR, BCONR.CSTA, and status bit (STFLR.CMFA, STFLR.OVFA, STFLR.ICPA) 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=0b1), when performing consecutive write actions to any of the following registers: CNTAR, CMPAR, BCONR.CSTA, or the status bits (STFLR.CMFA, STFLR.OVFA, STFLR.ICPA), you must wait until the status flag STFLR.SY-DA becomes 1 before issuing the next write to any of these registers.
4. When asynchronous counting is selected, ensure that BCONR.SYNCLKA is set to 0.
5. When writing to BCONR.CSTA, do not simultaneously overwrite BCONR.SYNSA (BCONR.SYNSA must retain its current value); otherwise, BCONR.CSTA cannot be written successfully.
6. 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.

21 General Timer (Timer0)

21.1 Introduction

General Timer 0 (Timer0) is a basic timer that can implement both synchronous counting and asynchronous counting. This timer contains one channel, which can generate compare match events and counter overflow events during counting. These events can trigger interrupts or serve as event outputs to control other modules. This product series is equipped with one unit of Timer0.

21.2 Functional Description

21.2.1 Timer0 Block Diagram

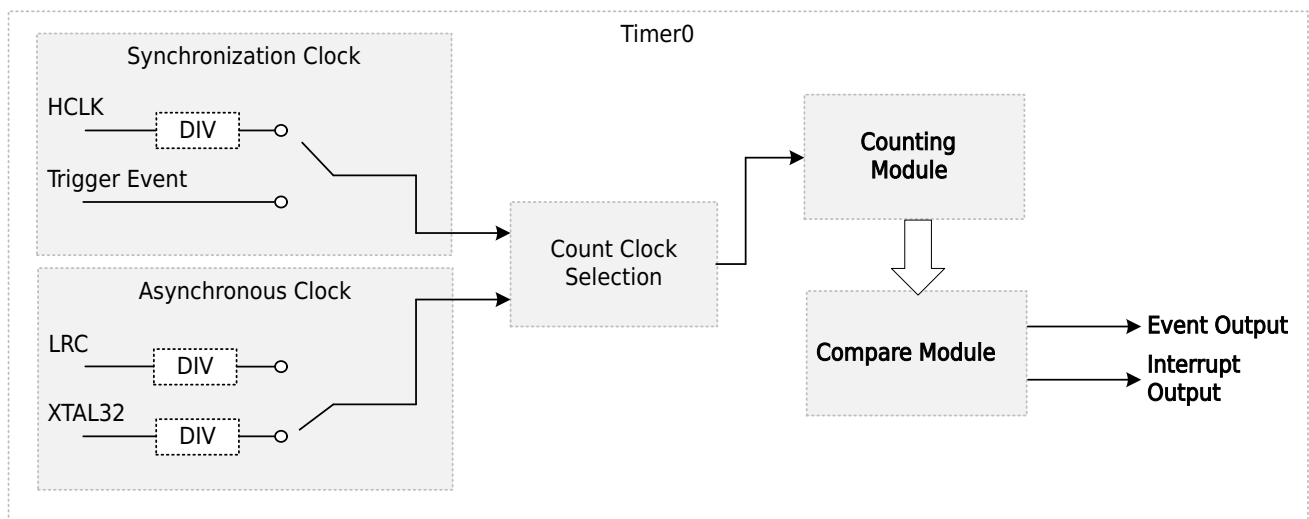


Figure 21-1 Timer0 Basic Block Diagram

21.2.2 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 asynchronous 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 (BCONR.SYNSA=0), the clock source can have the following options (BCONR.SYNCLKA setting) :

1. HCLK and HCLK divided by 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024 are used as synchronous count clocks (BCONR.SYNCLKA=0 & BCONR.CKDIVA settings);
2. An internal hardware trigger event input can serve as the synchronous count clock (BCONR.SYNCLKA=1);

Asynchronous Count Clock Source

In asynchronous counting mode ($\text{BCONR.SYNSA}=1$), the clock source can be selected from the following options (BCONR.ASYNCLKA 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 ($\text{BCONR.ASYNCLKA}=0$ & BCONR.CKDIVA 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 ($\text{BCONR.ASYNCLKA}=1$ & BCONR.CKDIVA settings).

21.2.3 Basic Counting

Timer0 can set the reference count value, and generate a count compare match event when the count value is equal to the reference value.

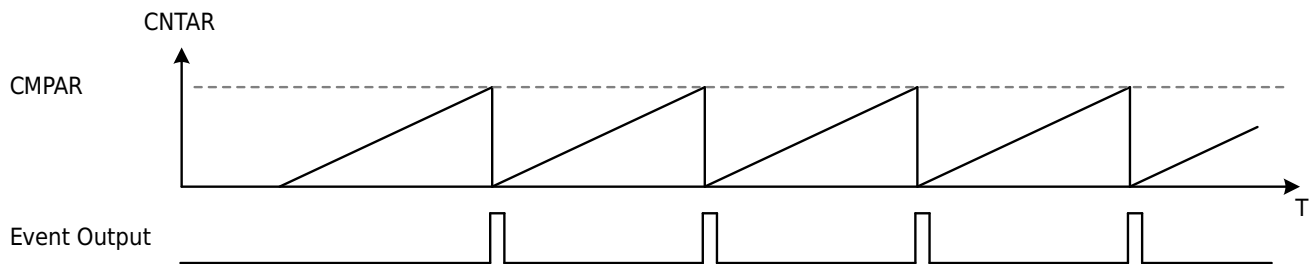


Figure 21-2 Timer0 Counting Timing Diagram

21.2.4 Hardware Triggering

Timer0 has a 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 Register (CMU_FCG) must be enabled first.

21.2.5 Interrupt and Event

21.2.5.1 Interrupt

A Timer0 contains two interrupt outputs: the compare match or capture input interrupt, and the count overflow interrupt.

The CMPAR register is compared with the CNTAR register to generate a valid compare match signal. When a count compare match occurs, the STFLR.CMFA bit is set to 1. If the BCONR.INTENA bit is set to enable the interrupt, the corresponding interrupt request (TMR0_GCMP) is also triggered.

When the internal hardware trigger input serves as the capture input condition, a corresponding capture input action can be generated. If the BCONR.INTENA bit is set to enable the interrupt, the corresponding interrupt request (TMR0_GCMP) is triggered.

When the CNTAR register counts to 0xFFFF, a count overflow event occurs. At this time, the STFLR.OVFA bit is set to 1. If the BCONR.OVENA bit is set to enable the interrupt, the corresponding interrupt request (TMR0_OVF) is also triggered.

When the asynchronous counting mode is selected, the compare match interrupt generated by the CMPAR register can be used to wake up the system in low power consumption mode. For details, refer to the description in the "Interrupt Controller (INTC)" chapter.

21.2.5.2 Event

During the counting process, when a count compare match or count overflow occurs, corresponding event request signals (TMR0_GCMP, TMR0_OVF) are generated. These signals can be used to trigger other modules.

21.3 Register Description

21.3.1 Register List

Base address: 0x40005800

Table 21-1 General Timer (Timer0) Register Offset

Offset	Register	Description	Width
0x0000	TMRO_CNTAR	Counter Value Register	32
0x0008	TMRO_CMPAR	Compare Value Register	32
0x0010	TMRO_BCONR	Basic Control Register	32
0x0014	TMRO_STFLR	Status Flag Register	32

21.3.2 Counter Value Register (TMR0_CNTAR)

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																CNTA															
Access																	RW															

Bit	Name	Function	Description	Access
31:16	Reserved	Reserved	-	-
15:0	CNTA	Counter Value A	Specifies the current timer counter value.  Note Do not write to this register while counting is in progress.	RW

21.3.3 Compare Value Register (TMR0_CMPAR)

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	0x0000FFFF																															
Name	Reserved																CMPA															
Access																	RW															

Bit	Name	Function	Description	Access
31:16	Reserved	Reserved	-	-
15:0	CMPA	Reference Value A	Specifies the counting reference value A, generating a compare match event	RW

21.3.4 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	Reserved																	HICPA		HCLEA		HSTPA		HSTAA		Reserved		ASYNCLKA		SYNCLKA		SYNSA		CKDIVA					OVENA		CMENA		CAPMDA		CSTA																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																		
Access																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																															

Bit	Name	Function	Description	Access
31:16	Reserved	Reserved	-	-
15	HICPA	Hardware Trigger Capture Input A	Condition: Internal Hardware Trigger Event Active 0b0: Capture input is disabled when the condition matches 0b1: Capture input is enabled when the condition matches	RW
14	HCLEA	Hardware Trigger Clear A	Condition: Internal Hardware Trigger Event Active 0b0: Timer clear operation is disabled when the condition matches 0b1: Timer clear operation is enabled when the condition matches	RW
13	HSTPA	Hardware Trigger Stop A	Condition: Internal Hardware Trigger Event Active 0b0: Timer stop is disabled when the condition matches 0b1: Timer stop is enabled when the condition matches	RW
12	HSTAA	Hardware Trigger Start A	Condition: Internal Hardware Trigger Event Active 0b0: Timer start is disabled when the condition matches 0b1: Timer start is enabled when the condition matches	RW
11	Reserved	Reserved	-	-
10	ASYNCLKA	Channel A Asynchronous Count Clock Source Selection	0b0: LRC 0b1: XTAL32	RW
9	SYNCLKA	Channel A Synchronous Counting Clock Source Selection	0b0: HCLK 0b1: Internal hardware triggering event	RW

Bit	Name	Function	Description	Access
8	SYNSA	Channel A Counting Mode Selection	● 0b0: Synchronous counting mode ● 0b1: Asynchronous counting mode	RW
7:4	CKDIVA	Channel A Count Clock Frequency Division Selection	● 0b0000: Clock source ● 0b0001: Clock source/2 ● 0b0010: Clock source/4 ● 0b0011: Clock source/8 ● 0b0100: Clock source/16 ● 0b0101: Clock source/32 ● 0b0110: Clock source/64 ● 0b0111: Clock source/128 ● 0b1000: Clock source/256 ● 0b1001: Clock source/512 ● 0b1010: Clock source/1024 ● Others: Setting prohibited  Note The divided clock source can be various clock sources for asynchronous counting or HCLK for synchronous counting.	RW
3	OVENA	Count Overflow Interrupt Enable A	Interrupt condition: count overflow, that is, count value (CNTAR)=0xFFFF ● 0b0: Disable interrupt ● 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 ● 0b0: Disable interrupt ● 0b1: Enable interrupt	RW
1	CAPMDA	Pattern Selection A	● 0b0: Compare output function ● 0b1: Capture input function	RW
0	CSTA	Channel A Timer ON	● 0b0: Channel A timer OFF ● 0b1: Channel A timer ON  Note This bit will automatically be set to 0 when the hardware trigger stop condition is enable.	RW

21.3.5 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	0x00000100																																
Name	Reserved																							SYDA		Reserved			ICPA		OVFA		CMFA
Access																								RO				RW	RW	RW			

Bit	Name	Function	Description	Access
31:9	Reserved	Reserved	-	-
8	SYDA	Synchronization Completion Flag A	0b0: A channel register is synchronizing during asynchronous counting 0b1: A channel register synchronization is completed during asynchronous counting	RO
7:3	Reserved	Reserved	-	-
2	ICPA	Repeat Capture Flag A	0b0: No capture input occurred after the count match flag bit CMFA was set to 1 0b1: A capture input occurred again after the count match flag bit CMFA was set to 1	RW
1	OVFA	Count Overflow Flag A	0b0: The count value (CNTAR) does not count to 0xFFFF 0b1: The count value (CNTAR) counts to 0xFFFF	RW
0	CMFA	Count Match Flag A	0b0: The value of the CMPAR register is not equal to the count value (CNTAR) and no capture input action occurs 0b1: The value of the CMPAR register is equal to the count value (CNTAR) or a capture input action occurs	RW

21.4 Precautions

1. During asynchronous counting operation, first configure the asynchronous clock source by setting the BCONR.ASYNCLKA bit, then select the asynchronous counting mode by setting the BCONR.SYNSA bit, and then start Timer0.
2. When selecting asynchronous counting, when the count value (CNTAR), reference value (CMPAR), start bit (BCONR.CSTA), and status bits (STFLR.CMFA, STFLR.OVFA, STFLR.ICPA) are modified, Timer0 will write the modified values to the corresponding registers after 3 asynchronous counting clock following the write action.
3. When selecting asynchronous counting, during consecutive write operations to the count value (CNTAR), reference value (CMPAR), start bit (BCONR.CSTA), and status bits (STFLR.CMFA, STFLR.OVFA, STFLR.ICPA), it is necessary to wait until the status bit (STFLR.SYDA) is 1 before performing the next write operation to any of the aforementioned registers.
4. When selecting asynchronous counting, ensure that BCONR.SYNCLKA is set to 0.
5. When writing BCONR.CSTA, please do not rewrite BCONR.SYNSA at the same time (BCONR.SYNSA need to keep the current value), otherwise BCONR.CSTA cannot be written.
6. When the Timer0 event is used to trigger other peripheral actions, the reference value register for the channel corresponding to Timer0 must be set to a value greater than 5.

22 Real Time Clock (RTC)

22.1 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.

22.2 Features

Table 22-1 RTC Main Features

Count clock source	External low-speed oscillator (32.768kHz) External pin 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
Interrupt	Periodic interrupt
	Alarm interrupt

22.3 Functional Description

22.3.1 RTC Block Diagram

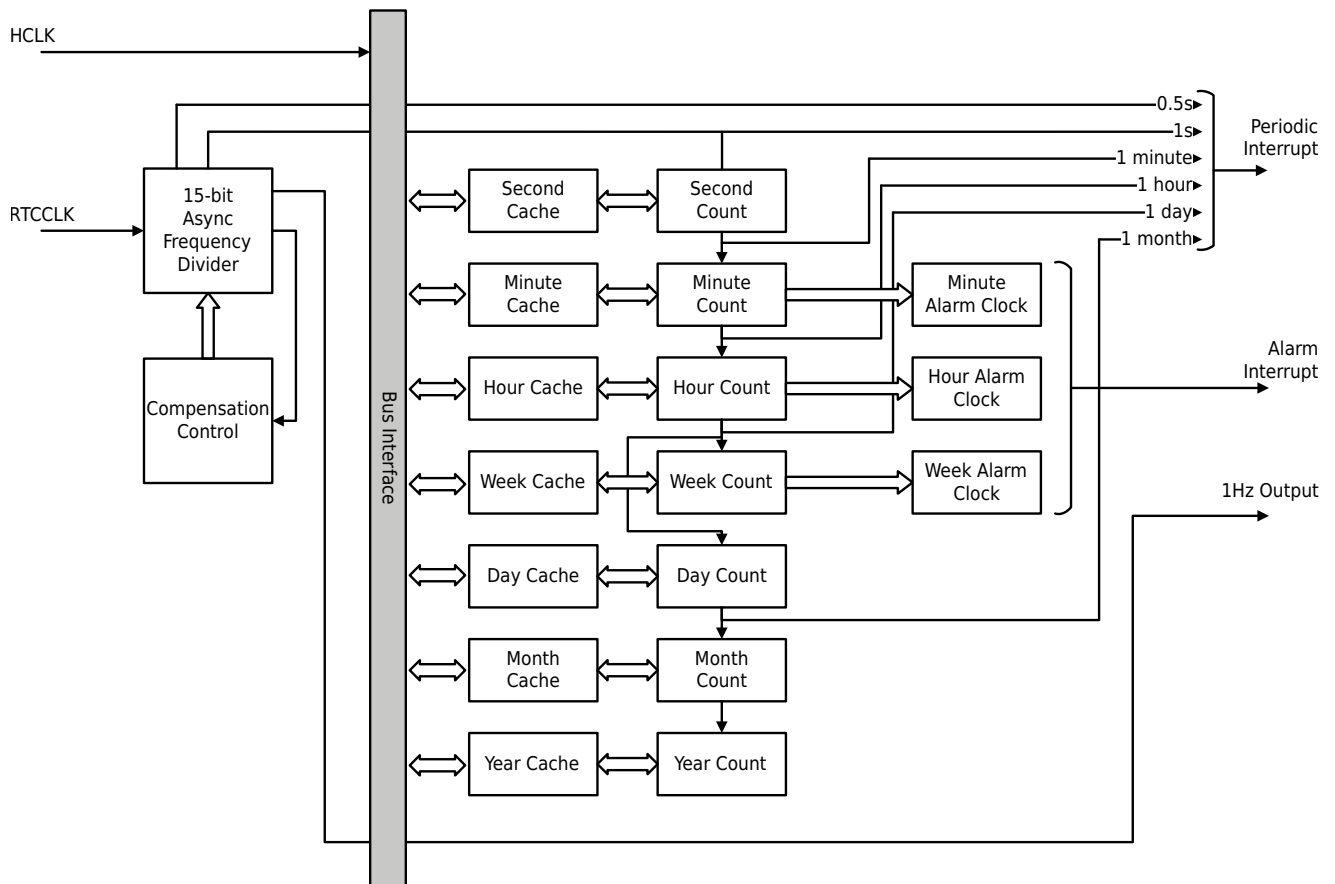


Figure 22-1 RTC Basic Block Diagram

22.3.2 Power-on Setting

The RTC register can be reset by power-on reset or setting `RTC_CR0.RESET`. After setting the control register, calendar initial value, alarm clock setting, etc., start the 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.

22.3.3 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.

22.3.4 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 \(RTC_ERRCRH and RTC_ERRCRL\)](#)".

22.3.5 Interrupt

RTC supports 2 interrupt types. Timing alarm interrupt, periodic 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 Register (RTC_ALMMIN), the Hour Alarm Register (RTC_ALMHOUR), and the Week Alarm Register (RTC_ALMWEEK) equal, trigger the alarm interrupt.

Periodic Interrupt

Periodic interrupt RTC_PRD, when RTC_CR2.PRDI=1, after the selected period occurs, a periodic wake-up interrupt is triggered.

22.4 Operation Example

22.4.1 RTC Count Start Setting

- Step1.** After power-on, 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, and enable interrupts;
- Step8.** Set RTC_CR1.START=1, counting starts.

22.4.2 Read Count Register

- Step1.** After RTC_CR1.START=1 is set, after more than 2 RTC count clocks, set RTC_CR2.RWREQ=1 to stop calendar register counting and enter read/write mode;
- Step2.** Query until RTC_CR2.RWEN=1;
- Step3.** Read the seconds, minutes, hours, weeks, days, months, and years counter register values;
- Step4.** Set RTC_CR2.RWREQ=0, counter counting;
- Step5.** Query until RTC_CR2.RWEN=0.

22.4.3 Write Count Register

- Step1.** After setting RTC_CR1.START=1, after more than 2 RTC counting clocks, set RTC_CR2.RWREQ=1 to stop calendar register counting and enter read/write mode;
- Step2.** Query until RTC_CR2.RWEN=1;
- Step3.** Write the seconds, minutes, hours, weeks, days, months, and years counter register values;
- Step4.** Set RTC_CR2.RWREQ=0, the counter starts counting again. Note that all write operations must be completed within 1 second;
- Step5.** Query until RTC_CR2.RWEN=0.

22.4.4 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.

22.4.5 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 N seconds (When RTC_ERRCRH.COMPRES=0b00, N is 32; when RTC_ERRCRH.COMPRES=0b01, N is 128; when RTC_ERRCRH.COMPRES=0b10, N is 256; when RTC_ERRCRH.COMPRES=0b11, N is 512);
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.** RTC_ERRCRH.COMPRES compensation resolution setting;
- Step7.** Clock error compensation register RTC_ERRCRL.COMP[7:0] and RTC_ERRCRH.COMP[12:8] compensation number setting;
- Step8.** Clock error compensation register RTC_ERRCRH.COMPEN=1, error compensation is enabled;
- Step9.** Set RTC_CR1.START=1, start counting;
- Step10.** Wait for more than 2 count periods;
- Step11.** 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.** RTC_ERRCRH.COMPRES compensation resolution setting, only supports being set to 0b00;
- Step7.** Clock error compensation register RTC_ERRCRL.COMP[7:0] and RTC_ERRCRH.COMP[12:8] compensation number setting;
- Step8.** Clock error compensation register RTC_ERRCRH.COMPEN=1, accuracy compensation is enabled;
- Step9.** Set RTC_CR1.START=1, start counting;
- Step10.** Wait for more than 2 count periods;
- Step11.** 1Hz output starts.

22.5 Register Description

22.5.1 Register List

Base address: 0x4000D400

Table 22-2 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
0x0060	RTC_FCLR	Flag Clear Register	8

22.5.2 Control Register 0 (RTC_CR0)

Offset	Bit Position							
0x0000	7	6	5	4	3	2	1	0
Reset	0x00							
Name	Reserved							RESET
Access								RW

Bit	Name	Function	Description	Access
7:1	Reserved	Reserved	-	-
0	RESET	RTC Calendar Counter Reset	Write 1 to initialize all RTC registers; write 0 has no effect. Read: 0b0: Normal counting status or RTC software reset ends 0b1: RTC is in reset state  Note Before writing "1", confirm that the bit is "0"; otherwise, initialization will fail.	RW

22.5.3 Control Register 1 (RTC_CR1)

Offset	Bit Position							
0x0004	7	6	5	4	3	2	1	0
Reset	0x00							
Name	START	ONEHZSEL	ONEHZOE	Reserved	AMPM	PRDS		
Access	RW	RW	RW		RW	RW		

Bit	Name	Function	Description	Access
7	START	RTC Counting Start	0b0: RTC counting stop 0b1: RTC counting start  Note When using an RTC event as the trigger source, if you need to stop the RTC, please reset the entire RTC through software reset (write 1 to RTC_CR0.RESET).	RW
6	ONEHZSEL	1Hz Output Selection	0b0: Distributed compensation 1Hz output 0b1: Uniform compensation 1Hz output  Note When RTC_ERRCRH.COMPEN=1, this bit setting is enable.	RW
5	ONEHZOE	1Hz Output Permission	0b0: 1Hz output disabled 0b1: 1Hz output permission	RW
4	<i>Reserved</i>	<i>Reserved</i>	-	-
3	AMPM	Time Format Selection	0b0: 12-hour time format 0b1: 24-hour time format	RW
2:0	PRDS	Period Interrupt Selection	Period selection settings: 0b000: No periodic interruption occurs 0b001: 0.5 seconds 0b010: 1 second 0b011: 1 minute 0b100: 1 hour 0b101: 1 day (Every day at 00:00 00) 0b110/0b111: 1 month (Every 1st of each month at 00:00 00)  Notice 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.	RW

22.5.4 Control Register 2 (RTC_CR2)

Offset	Bit Position							
0x0008	7	6	5	4	3	2	1	0
Reset	0x00							
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	0b0: Alarm function disabled 0b1: Alarm function enabled	RW
6	ALMIE	Alarm Interrupt Enabled	0b0: Alarm interrupt disabled 0b1: Alarm interrupt enabled	RW
5	PRDIE	Periodic Interrupt Enabled	0b0: Periodic interrupt disabled 0b1: Periodic interrupt enabled	RW
4	<i>Reserved</i>	<i>Reserved</i>	-	-
3	ALMF	Alarm Clock Flag	0b0: Alarm mismatch 0b1: Alarm match	RW
2	PRDF	Periodic Flag	0b0: Periodic does not occur 0b1: Periodic occur	RW
1	RWEN	Read/Write Enabled	0b0: Read/write disabled 0b1: Read/write enabled  Note 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.	RW
0	RWREQ	Read/Write Request	0b0: Normal counting mode 0b1: Read/write request  Note 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.	RW

22.5.5 Control Register 3 (RTC_CR3)

Offset	Bit Position							
0x000C	7	6	5	4	3	2	1	0
Reset	0x00							
Name	RCKSEL		Reserved					
Access	RW							

Bit	Name	Function	Description	Access
7:6	RCKSEL	RTC Count Clock Selection	● 0b00: External low-speed oscillator (XTAL32) ● 0b01: Setting prohibited ● 0b10: External pin clock ● 0b11: Setting prohibited	RW
5:0	Reserved	Reserved	-	-

22.5.6 Second Count Register (RTC_SEC)

Offset	Bit Position							
0x0010	7	6	5	4	3	2	1	0
Reset	0x00							
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.

22.5.7 Minute Count Register (RTC_MIN)

Offset	Bit Position							
0x0014	7	6	5	4	3	2	1	0
Reset	0x00							
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.

22.5.8 Hour Count Register (RTC_HOUR)

Offset	Bit Position							
0x0018	7	6	5	4	3	2	1	0
Reset	0x00							
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

22.5.9 Day Count Register (RTC_DAY)

Offset	Bit Position							
0x0020	7	6	5	4	3	2	1	0
Reset	0x00							
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

22.5.10 Week Count Register (RTC_WEEK)

Offset	Bit Position							
0x001C	7	6	5	4	3	2	1	0
Reset	0x00							
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

22.5.11 Month Count Register (RTC_MON)

Offset	Bit Position							
0x0024	7	6	5	4	3	2	1	0
Reset	0x00							
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.

22.5.12 Year Count Register (RTC_YEAR)

Offset	Bit Position							
0x0028	7	6	5	4	3	2	1	0
Reset	0x00							
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.

22.5.13 Minute Alarm Register (RTC_ALMMIN)

Offset	Bit Position							
0x002C	7	6	5	4	3	2	1	0
Reset	0x00							
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. If an out-of-range value is written, the alarm match will not occur.

22.5.14 Hour Alarm Register (RTC_ALMHOUR)

Offset	Bit Position							
0x0030	7	6	5	4	3	2	1	0
Reset	0x00							
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.

22.5.15 Week Alarm Register (RTC_ALMWEEK)

Offset	Bit Position							
0x0034	7	6	5	4	3	2	1	0
Reset	0x00							
Name	Reserved	ALMWEEK						
Access		RW						

Bit	Name	Function	Description	Access
7	Reserved	Reserved	-	-
6:0	ALMWEEK	Week Alarm	Week Alarm Match Value 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. For example: Setting Bit0=1 and Bit5=1 means that the alarm settings on Sunday and Friday are enabled.	RW

Please write a value less than or equal to 0x7F. Writing a value outside this range will not trigger an alarm match.

22.5.16 Clock Error Compensation Register (RTC_ERRCRH, RTC_ERRCRL)

RTC_ERRCRH

Offset	Bit Position							
0x0038	7	6	5	4	3	2	1	0
Reset	0x00							
Name	COMPEN	COMPRES		COMP[12:8]				
Access	RW	RW		RW				

Bit	Name	Function	Description	Access
7	COMPEN	Compensation enable	0b0: Clock error compensation disabled 0b1: Clock error compensation enabled	RW
6:5	COMPRES	Compensation Resolution Selection	0b00: Compensation accuracy 0.96ppm 0b01: Compensation accuracy 0.238ppm 0b10: Compensation accuracy 0.119ppm 0b11: Compensation accuracy 0.06ppm	RW
4:0	COMP[12:8]	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	0x20							
Name	COMP[7:0]							
Access	RW							

Bit	Name	Function	Description	Access
7:0	COMP[7:0]	Compensation Value	Through compensation value setting, accuracy compensation of $\pm 0.96\text{ppm}$, $\pm 0.238\text{ppm}$, $\pm 0.119\text{ppm}$, or $\pm 0.06\text{ppm}$ per second can be achieved. The compensation resolution is set by RTC_ERRCRH.COMPRES. - When RTC_ERRCRH.COMPRES=0b00, the compensation value is a 9-bit 2's complement number with a decimal point, set via COMP[8:0], with the last 5 bits as the fractional part. The compensation range is -274.65ppm to +212.67ppm. The minimum resolution is 0.96ppm. - When RTC_ERRCRH.COMPRES=0b01, the compensation value is an 11-bit 2's complement number with a decimal point, set via COMP[10:0], with the last 7 bits as the fractional part. The compensation range is -274.65ppm to +213.38ppm. The minimum resolution is 0.238ppm. - When RTC_ERRCRH.COMPRES=0b10, the compensation value is a 12-bit 2's complement number with a decimal point, set via COMP[11:0], with the last 8 bits as the fractional part. The compensation range is -274.65ppm to +213.50ppm. The minimum resolution is 0.119ppm. - When RTC_ERRCRH.COMPRES=0b11, the compensation value is a 13-bit 2's complement number with a decimal point, set via COMP[12:0], with the last 9 bits as the fractional part. The compensation range is -274.65ppm to +213.56ppm. The minimum resolution is 0.06ppm.	RW

Please refer to the following table for specific compensation accuracy. Taking RTC_ERRCRH.COMPRES=0b00 as an example,:

Compensation Value Setting										Compensation Value
COMP-EN	COMP[8:0]									
1	1	0	0	0	0	0	0	0	0	-274.65ppm
	1	0	0	0	0	0	0	0	1	-273.7ppm
	...									...
	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	+211.71ppm
	0	1	1	1	1	1	1	1	1	+212.67ppm
0	X	X	X	X	X	X	X	X	X	Uncompensated

The formula for calculating the register value under different compensation accuracies:

$$\text{COMP}[12:0] = \begin{cases} \left(\frac{\text{Compensation Target Value[ppm]} \times 2^{15}}{10^6} \right)_{\text{Take 2's complement}} + 0b0001.00000 & (\text{COMPRES}=0b00) \\ \left(\frac{\text{Compensation Target Value[ppm]} \times 2^{15}}{10^6} \right)_{\text{Take 2's complement}} + 0b0001.0000000 & (\text{COMPRES}=0b01) \\ \left(\frac{\text{Compensation Target Value[ppm]} \times 2^{15}}{10^6} \right)_{\text{Take 2's complement}} + 0b0001.00000000 & (\text{COMPRES}=0b10) \\ \left(\frac{\text{Compensation Target Value[ppm]} \times 2^{15}}{10^6} \right)_{\text{Take 2's complement}} + 0b0001.000000000 & (\text{COMPRES}=0b11) \end{cases}$$

Compensation Calculation Instructions (Take RTC_ERRCRH.COMPRES=0b00 for example)

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:

$$\begin{aligned}
 \text{Actual Vibration Frequency} &= 32768 \times 0.9999888 \approx 32767.63 \\
 \text{Compensation Target Value} &= (\text{Actual Vibration Frequency} - \text{Target Frequency}) / \text{Target Frequency} \times 10^6 \\
 &= (32767.63 - 32768) / 32768 \times 10^6 \\
 &= -11.29\text{ppm}
 \end{aligned}$$

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:

```
COMP[8:0]=(20.3 × 215/106) takes 2's complement + 0b0001.00000  
=(0.6651904) 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:

```
COMP[8:0]=(-20.3 × 215/106) takes 2's complement + 0b0001.00000  
=(-0.6651904) takes 2's complement + 0b0001.00000  
=0b1111.01011 + 0b0001.00000  
=0b0000.01011
```

22.5.17 Flag Clear Register (RTC_FCLR)

Offset	Bit Position							
0x0060	7	6	5	4	3	2	1	0
Reset	0x00							
Name	Reserved				ALMFCLR	PRDFCLR	Reserved	
Access					RW	RW		

Bit	Name	Function	Description	Access
7:4	Reserved	Reserved	-	-
3	ALMFCLR	Alarm Flag Clear Bit	Write 1 to clear the flag, write 0 has no effect; read always returns 0	RW
2	PRDFCLR	Periodic Flag Clear Bit	Write 1 to clear the flag, write 0 has no effect; read always returns 0	RW
1:0	Reserved	Reserved	-	-

22.6 Precautions

1. When the HCLK clock uses the same clock source as the RTC count clock, the HCLK frequency is greater than or equal to the RTC count clock frequency.
2. When the HCLK clock uses a different clock source as the RTC count clock, the HCLK frequency is greater than or equal to 2 times the RTC count clock frequency.

23 Watchdog Timer (WDT/SWDT)

23.1 Introduction

The watchdog counter is a dedicated watchdog counter (SWDT) with an internal low-speed RC (32.768kHz) as its count clock source. The dedicated watchdog is a 16-bit down counter used to monitor software faults caused by external interference or unforeseen logic conditions that cause the application to deviate from normal operation. The watchdog counter supports a window function. A window interval can be preset before counting starts. When the count value is within the window interval, the counter can be refreshed and counting restarts.

23.2 Features

Table 23-1 Basic Features of Watchdog Timer

Count Clock	1/16/32/64/128/256/2048 frequency division of SWDTCLK
Maximum Over-flow Time	1.14 hours
Counting Mode	Decrement count
Window Function	Can set window interval, define the allowable interval of refresh action
Startup Mode	1. Hardware startup 2. Software startup
Stop Condition	1. Resetting 2. Underflow, Refresh error occurs and then restart: ● In hardware startup mode, it starts automatically after a reset or interrupt request is output ● In software startup mode, set refresh register again
Interrupt/Reset Conditions	1. Count underflow 2. Refresh error

23.3 Functional Description

23.3.1 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 (ICGA0x 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.

23.3.2 Hardware Startup Mode

When the ICGA0x.SWDTAUTS is set to 0, it is the hardware startup mode. SWDT_CR settings are ignored in hardware startup mode.

In hardware start mode, the SWDT-related settings (counting clock, window setting value, counting period, etc.) in the ICGA0x register are loaded into the 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.

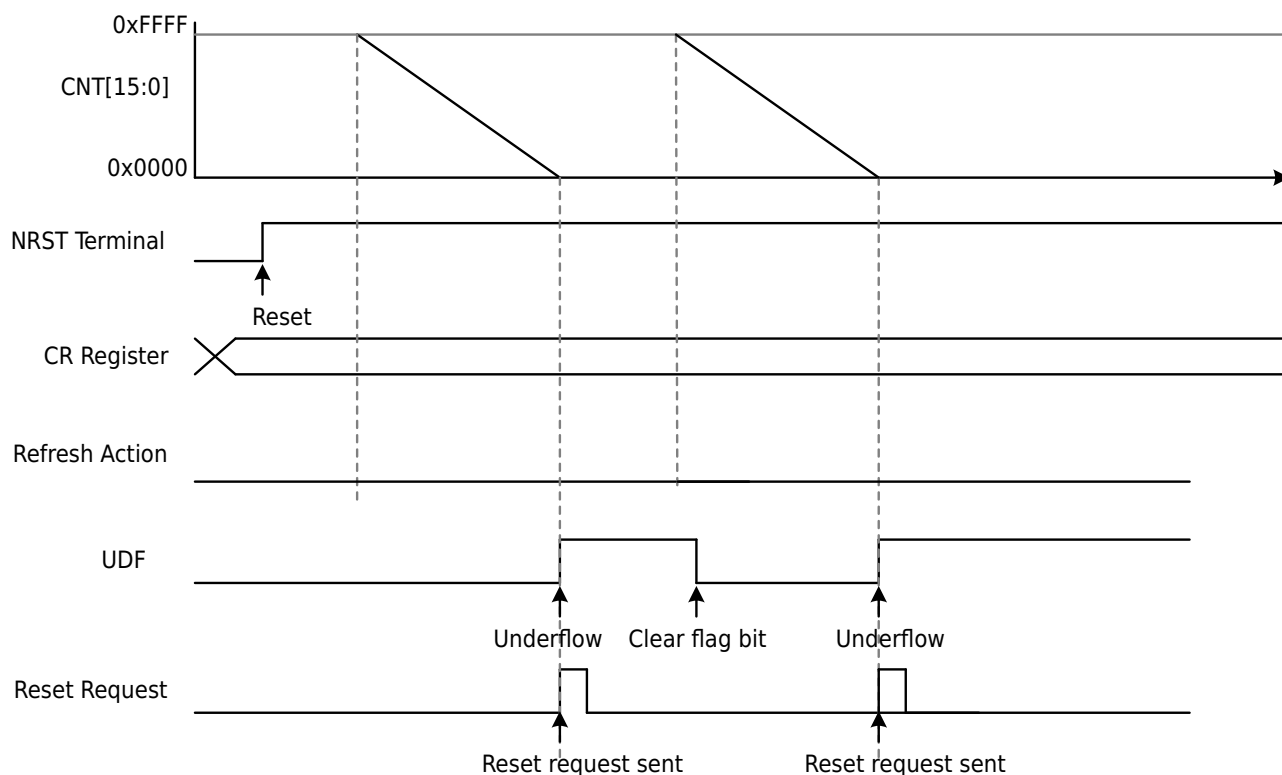


Figure 23-1 Hardware Startup Mode Example

23.3.3 Software Startup Mode

When the ICGA0x.SWDTAUTS is set to 1, SWDT is started via refresh register (software start mode). After reset, set the counting clock, window setting value, counting period, etc. in the SWDT_CR register, and then perform the refresh action, and the counter starts to count. SWDT_CR can be configured only once; subsequent writes are inactive. The figure below shows a software start mode example.

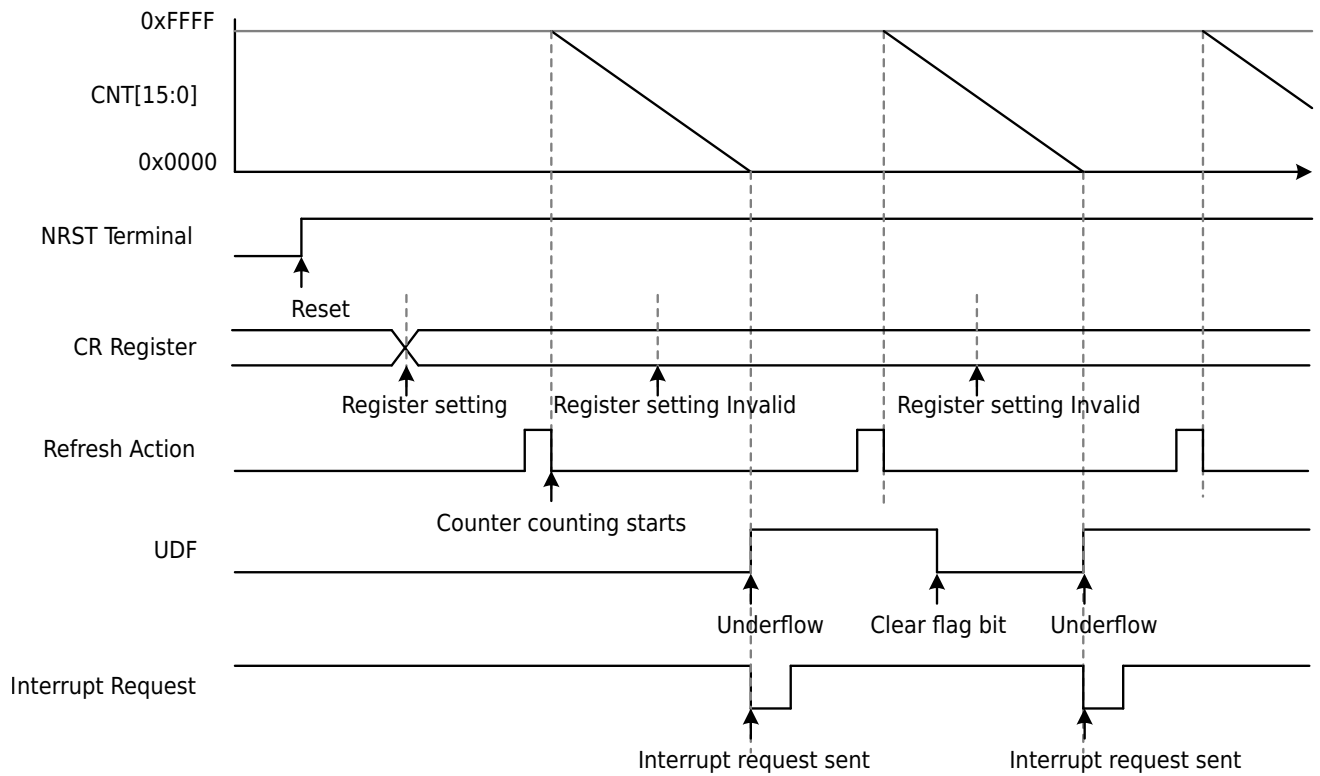


Figure 23-2 Software Startup Mode Example

23.3.4 Refresh Action

A refresh sequence (write 0x0123 followed by 0x3210 to SWDT_RR) starts (software start) or restarts the counter.

Writing between 0x0123 and 0x3210 to SWDT_RR register is unaffected by other register accesses or reads during the refresh sequence.

The figure below shows an operation example.

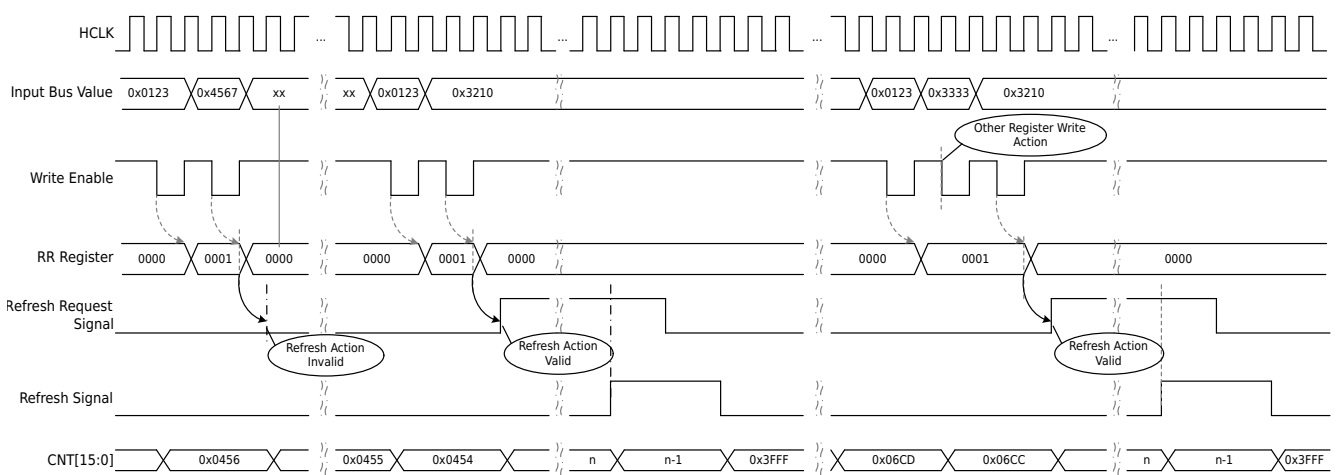


Figure 23-3 Examples of Various Refresh Action Sequences (action confirmation, refresh required signal falling edge, etc.)

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.

23.3.5 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 SWDTCLK.

When refreshing error flag bit and underflow flag bit, the counting of the hardware watchdog does not stop. When writing "0" to the flag bit, the SWDT needs at most 3 SWDTCLK and 2 HCLK times before the register bit can 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 SWDT_SR to check the flag bit.
- Step2.** Write "0" to the flag bit set to "1".
- Step3.** Read SWDT_SR to confirm if the flag bit is cleared. Repeat Step 2 if the flag bit remain 1.

23.3.6 Interrupt/Reset

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 ICGA0x.SWDTITS bit. In software startup mode, set the SWDT_CR.ITS bit to select interrupt or reset request.

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.

23.3.7 Count Underflow

As shown in the following figure, underflow occurs when the counter decrements to 0.

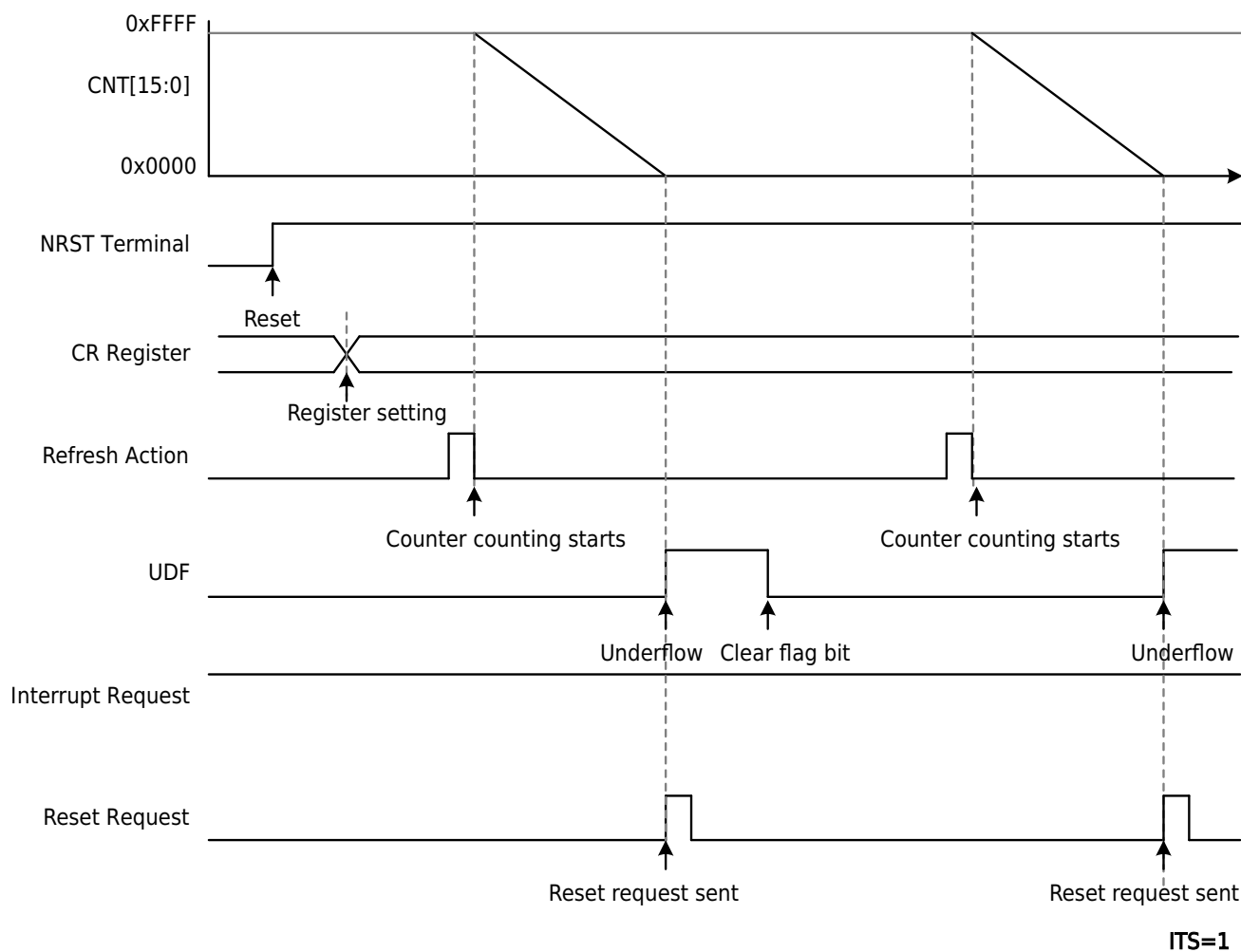


Figure 23-4 Counter Underflow Operation Example

23.3.8 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.

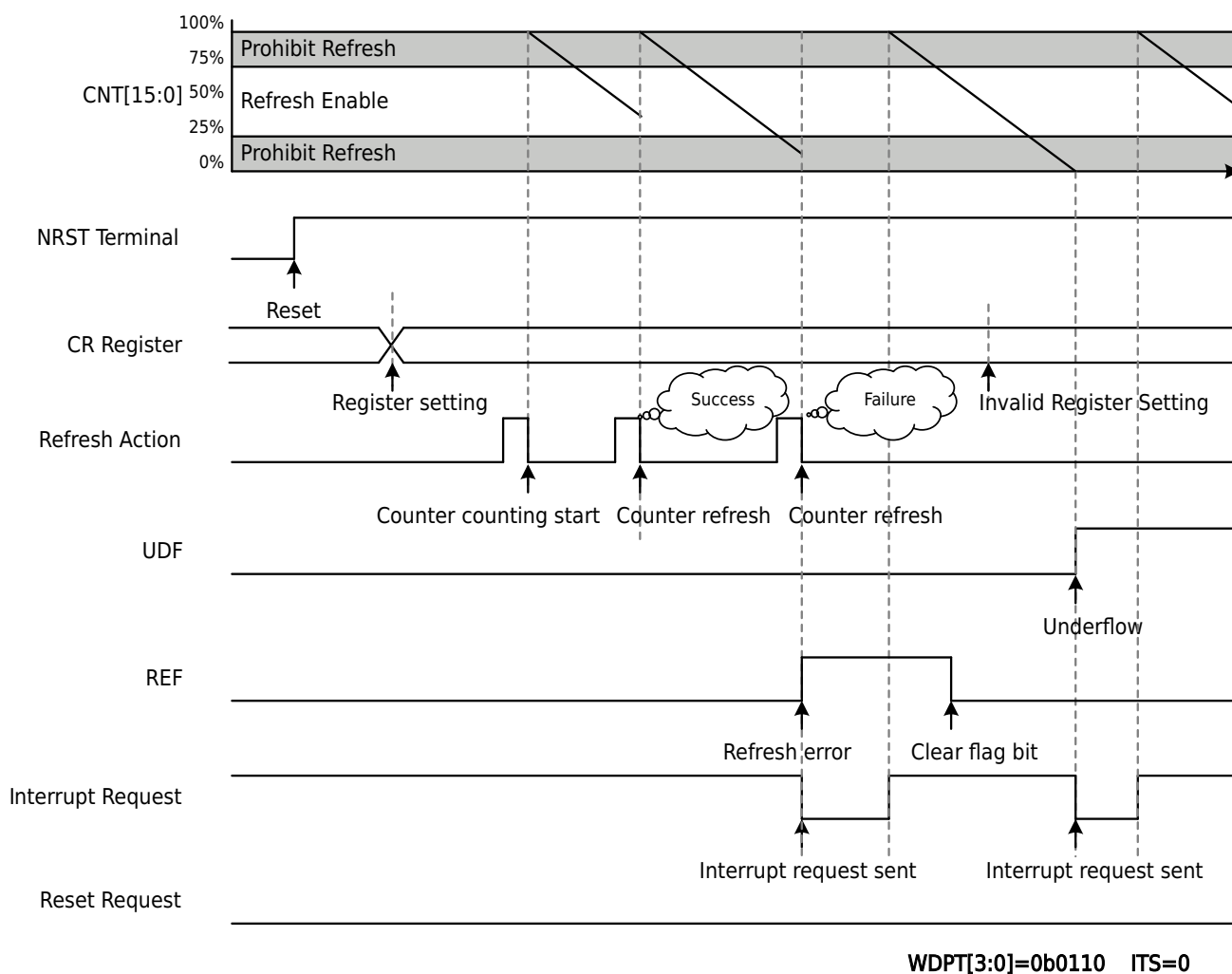


Figure 23-5 Counter Refresh Operation Example

23.4 Register Description

23.4.1 Register List

SWDT base address: 0x4000CC00

Table 23-2 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

23.4.2 Control Register (SWDT_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					RW	

Bit	Name	Function	Description	Access
31	ITS	Refresh Error/ Overflow Inter- rupt/Reset Se- lection	0b0: Interrupt request 0b1: Reset request	RW
30:17	Reserved	Reserved	-	-
16	SLPOFF	SWDT Counting is Disabled in Low-power Modes	0b0: Enable SWDT counting in Sleep/Stop mode 0b1: Disable SWDT counting in Sleep/Stop mode	RW
15:12	Reserved	Reserved	-	-
11:8	WDPT	Refresh the Percentage of the Allowed Area Count Val- ue	0b0000: 0%-100% 0b0001: 0%-25% 0b0010: 25%-50% 0b0011: 0%-50% 0b0100: 50%-75% 0b0101: 0%-25%, 50%-75% 0b0110: 25%-75% 0b0111: 0%-75% 0b1000: 75%-100% 0b1001: 0%-25%, 75%-100% 0b1010: 25%-50%, 75%-100% 0b1011: 0%-50%, 75%-100% 0b1100: 50%-100% 0b1101: 0%-25%, 50%-100% 0b1110: 25%-100% 0b1111: 0%-100%	RW
7:4	CKS	Count Clock	0b0000: SWDTCLK 0b0100: SWDTCLK/16 0b0101: SWDTCLK/32 0b0110: SWDTCLK/64 0b0111: SWDTCLK/128 0b1000: SWDTCLK/256 0b1011: SWDTCLK/2048 - Others: Setting prohibited	RW

Bit	Name	Function	Description	Access
3:2	Reserved	Reserved	-	-
1:0	PERI	Counting Period	0b00: 256 cycle 0b01: 4096 cycle 0b10: 16384 cycle 0b11: 65536 cycle	RW

23.4.3 Status Register (SWDT_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 CNT																

Bit	Name	Function	Description	Access
31:18	Reserved	Reserved	-	-
17	REF	Refresh Error Flag	0b0: No refresh error 0b1: Refresh error occurred After reading a value of 1 from this bit, writing 0 to it will clear this bit.	RW
16	UDF	Counting Underflow Mark	0b0: No counter underflow 0b1: Counter underflow occurred 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

23.4.4 Refresh Register (SWDT_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 sequentially to complete the refresh. After writing 0x0123, the register reads as 0x00000001; in all other cases, it reads as 0x00000000.	RW

23.5 Precautions

During SWDT operation, the frequency of the clock HCLK must equal to or greater than 4 timers of counting clock frequency (Either ICGA0x.SWDTCKS or SWDT_CR.CKS sets the clock after frequency division for SWDTCLK). That is, HCLK frequency $\geq$ counting clock frequency $\times 4$.

24 Universal Synchronous Asynchronous Receiver/Transmitter (USART)

24.1 Introduction

This product is equipped with 4 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, receive timeout function. USART3/USART4 supports the STOP mode wake-up function via the RX line.

24.2 Features

Function	Support Module			
	USART1	USART2	USART3	USART4
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 synchronization 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 3, 4 supports stop mode wake-up function
- Unit 3, 4 supports 10/11-bit synchronization Break Field (BF) detection and 10/11/13/14-bit Break Field (BF) transmission
- 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 Break Field (BF) detection
- Supports synchronization segment measurement, register records measurement value
- Supports 10/11/13/14-bit Break Field (BF) transmission
- Supports bus conflict detection
- Supports loop mode

24.3 Functional Description

24.3.1 System Block Diagram

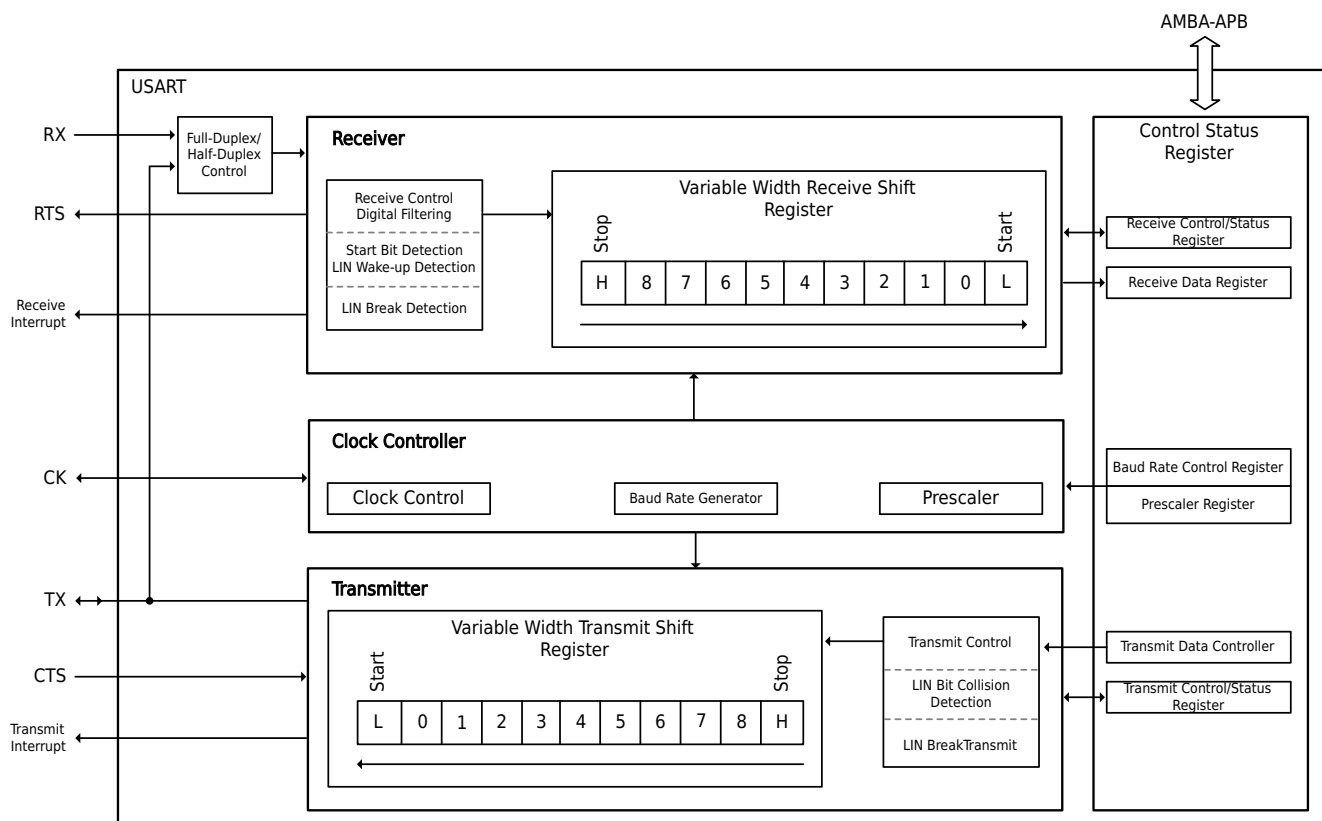


Figure 24-1 USART System Block Diagram

Table 24-2 USART pin description

Pin Name	Direction	Description
USARTn_CK	Input/Output	Clock
USARTn_TX	Input/Output	Transmit data pin Also used as receive data pin in half-duplex mode
USARTn_RX	Input	Receive data pin
USARTn_CTS	Input	Modem operation pin Clear send pin
USARTn_RTS/DE	Output	Modem operation pin Request send pin/Driver Enable pin



Note

The value of n is 1-4.

24.3.2 UART Function

24.3.2.1 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 HCLK, HCLK/4, HCLK/16, HCLK/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 HCLK/128, HCLK/256, HCLK/512, HCLK/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 (HCLK, HCLK/4, HCLK/16, HCLK/64), unit: MHz
When USARTn_PR.ULBREN is 1, the clock set by USARTn_PR.PSC bit (HCLK/128, HCLK/256, HCLK/512, HCLK/1024), unit: MHz
- OVER8: USARTn_CR1.OVER8 setting value
- DIV_Integer: USARTn_BRR.DIV_Integer setting value

Maximum Baud Rate is HCLK/8 (Mbps).

When using an External Clock Source, the maximum frequency requirement for the external input UART Clock is HCLK/4 (MHz). Therefore, when the Clock Source is an external input clock:

- If USARTn_CR1.OVER8=0, the Maximum Baud Rate is HCLK/64 (Mbps).
- If USARTn_CR1.OVER8=1, the Maximum Baud Rate is HCLK/32 (Mbps).

Please note that in addition to the HCLK-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.

24.3.2.2 Data Format

A frame of data in UART mode consists of start bit, data bit, check bit, and stop bit.

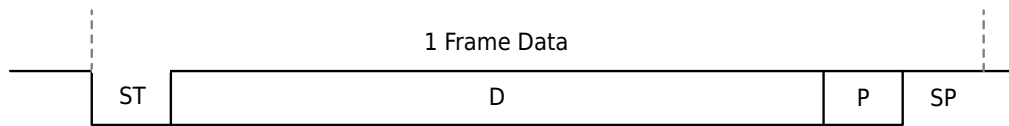


Figure 24-2 UART Data Format

- **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.

24.3.2.3 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.

24.3.2.4 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

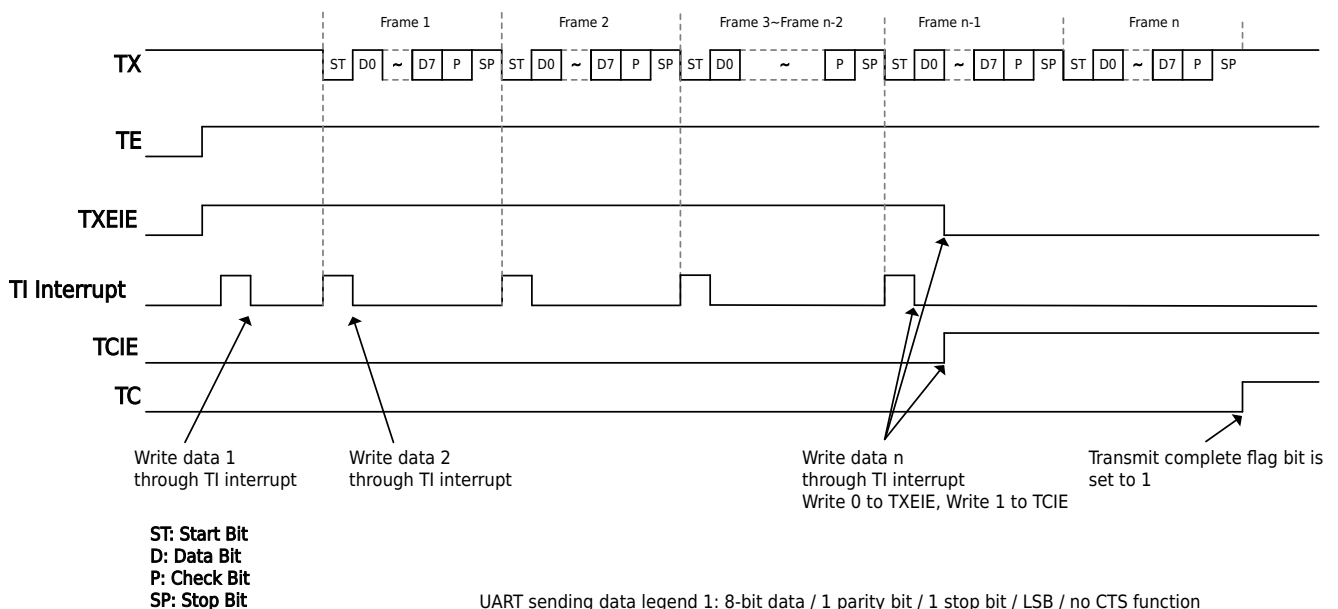
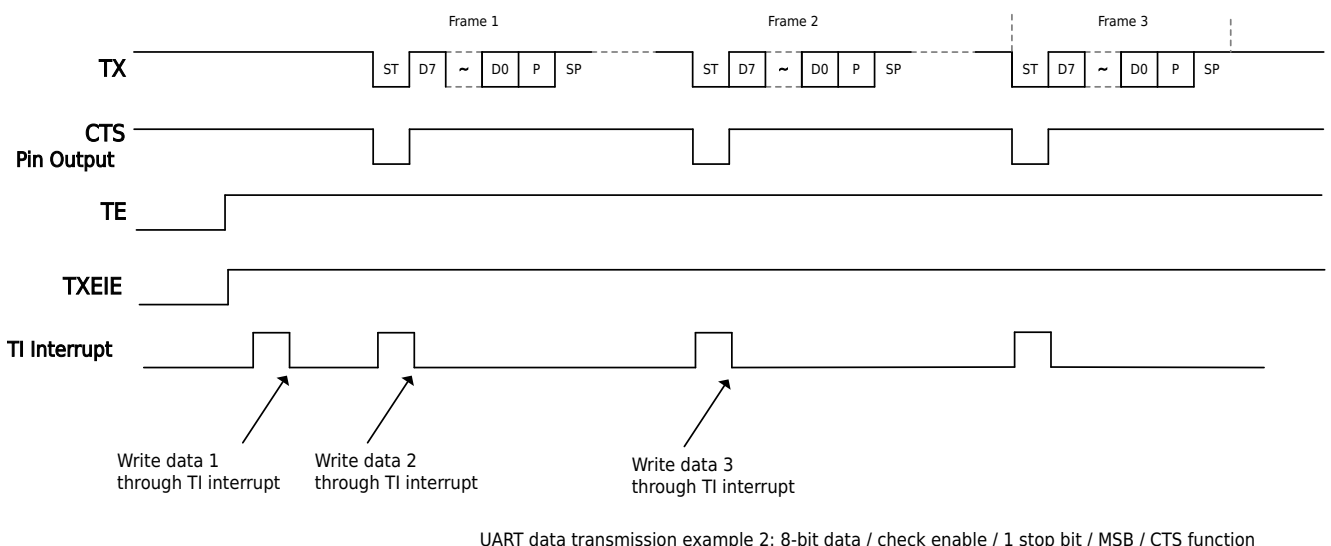
- Step1.** Set the USARTn_CR1 register to its reset value;
- Step2.** Configure the pins required for UART operation;
- Step3.** Select the clock source via the USARTn_CR2.CLKC bit;
- Step4.** Configure the USARTn_CR1, USARTn_CR2, and USARTn_CR3 Registers;

- 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 24-3 UART Transmit Data Diagram 1****Figure 24-4 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.

24.3.2.5 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.

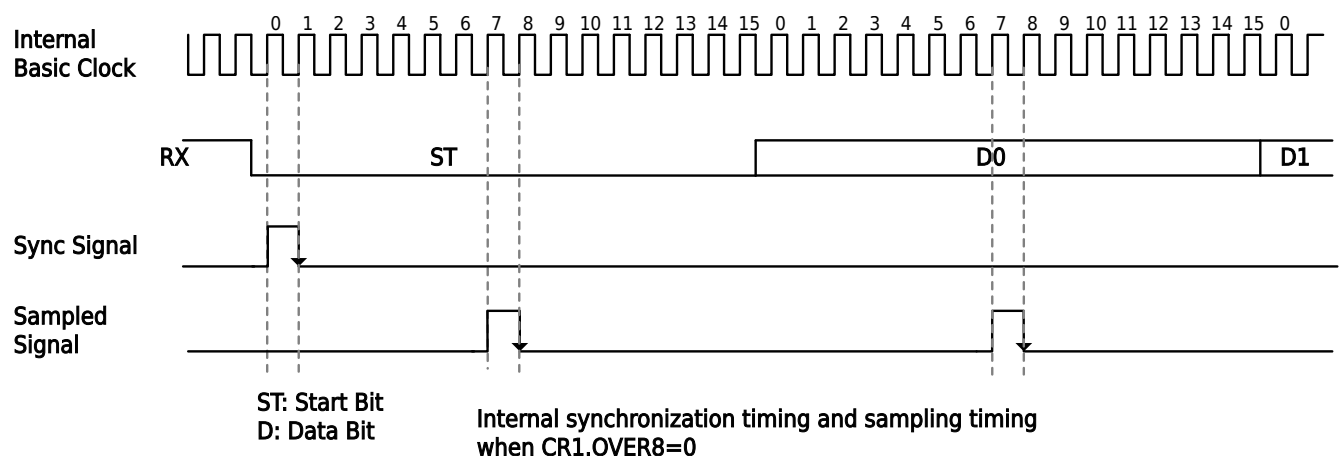


Figure 24-5 USART Internal Synchronization and Sample Timing

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:

- Frame Length (FL). FL is determined by the 8 or 9 data bits defined by the USARTn_CR1.M bit and the parity enable bit defined by the PCE bit.
- The oversampling ratio (8x or 16x) is defined by the USARTn_CR1.OVER8 bit.
- The USARTn_CR1.FBME bit defines whether to use fractional baud rate.

Table 24-3 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.

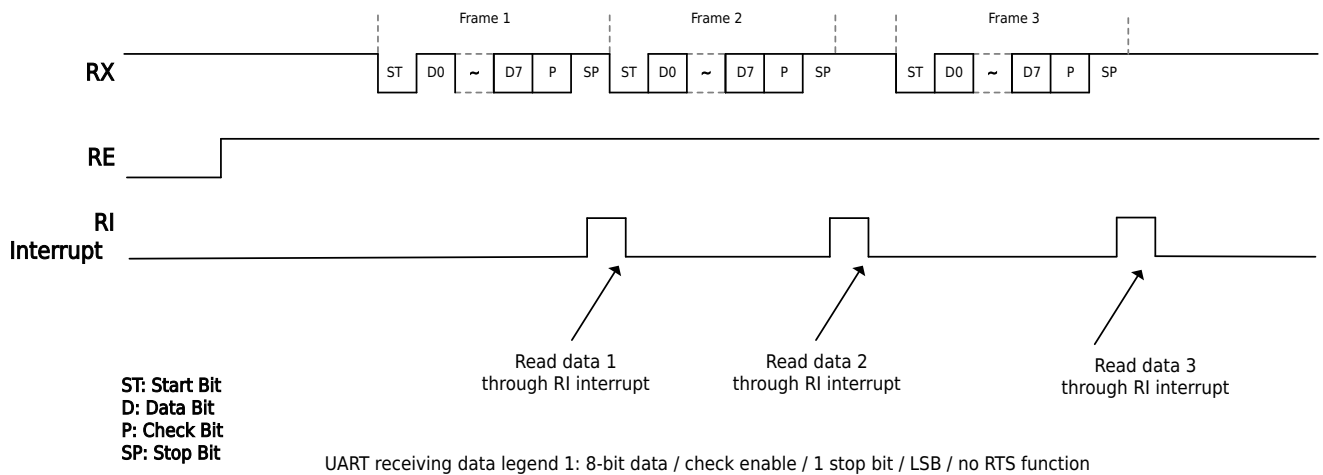


Figure 24-6 UART Receive Data Diagram 1

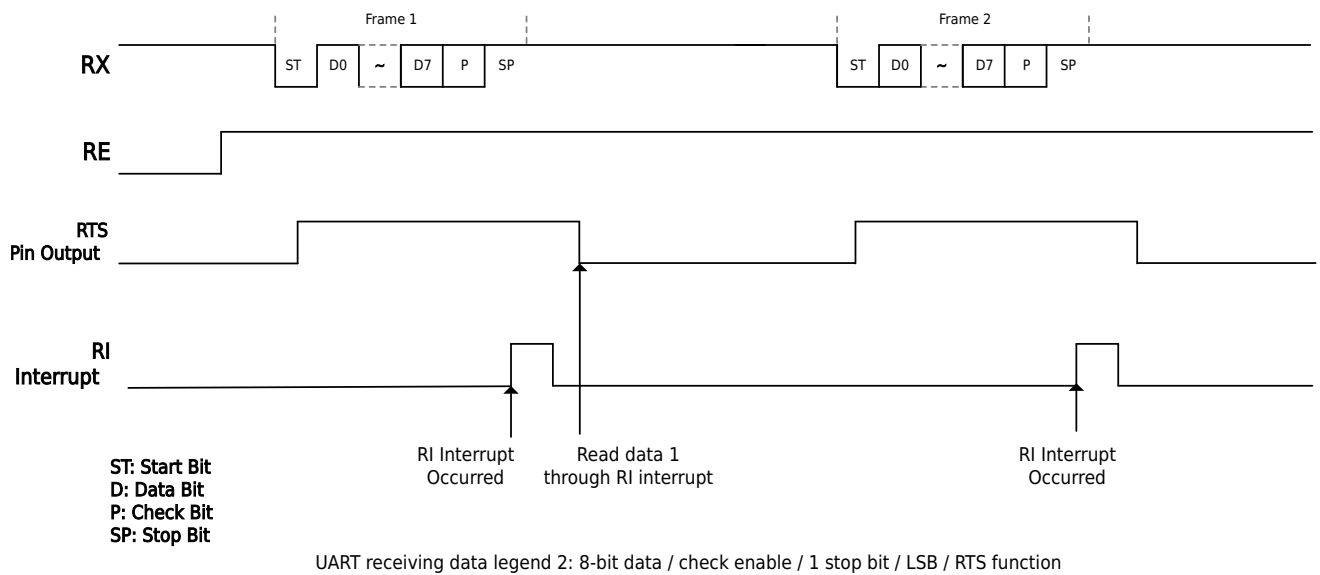


Figure 24-7 UART Receive Data Diagram 2

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 UART Mode receiver supports two types of interrupts: receive data register full interrupt (RI) and error interrupt (EI).

- USARTn_CR1.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_CR1.RIE=1, when an overflow error, parity error, or frame error occurs during reception, an EI interrupt is generated.

24.3.2.6 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 USARTn_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.

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 USARTn_PR.RTOB to determine the TIMEOUT period;
- Step8.** Enable the receiver (USARTn_CR1.RE=1). If receive interrupt is required, set USARTn_CR1.RIE=1.
- Step9.** Upon detecting TIMEOUT, clear the USARTn_SR.RTOF bit by writing USARTn_CR1.CRTOF.

24.3.2.7 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 3, 4 can wake up the system from Stop mode via the RX line. The specific steps are as follows, taking USART3 as an example:

- Step1.** When UART communication is idle, set the USART_3_WUPI interrupt vector and enable the UART receive signal line wake-up stop mode function by setting the INTC_WKEN.USART3WKEN 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_3_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 specific filtering capabilities, please refer to the relevant descriptions in the Datasheet.

24.3.2.8 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.

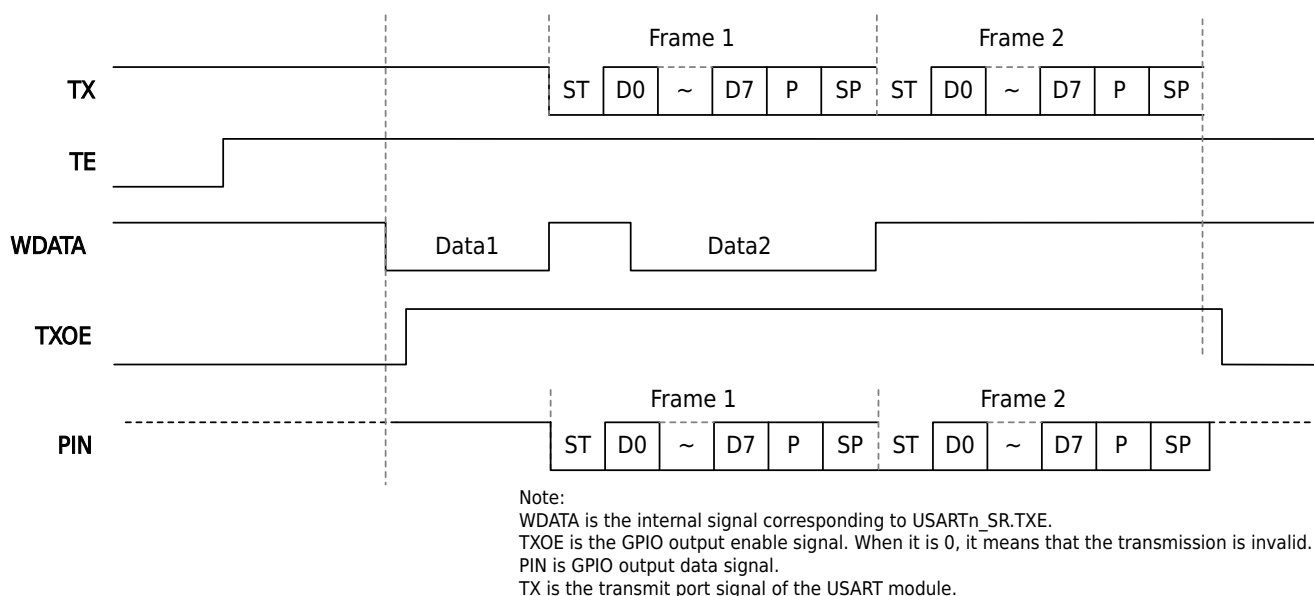


Figure 24-8 Half-Duplex Operation Timing

24.3.2.9 RS485 Driver Enable Function

UART mode supports Driver Enable mode, sharing the output signal with RTS. First configure USARTn_CR3.DEM to 1, then set USARTn_CR3.RTSE to 1.

By setting the USARTn_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 USARTn_CR3.DEA is the time between DE signal activation and the start of the START bit.

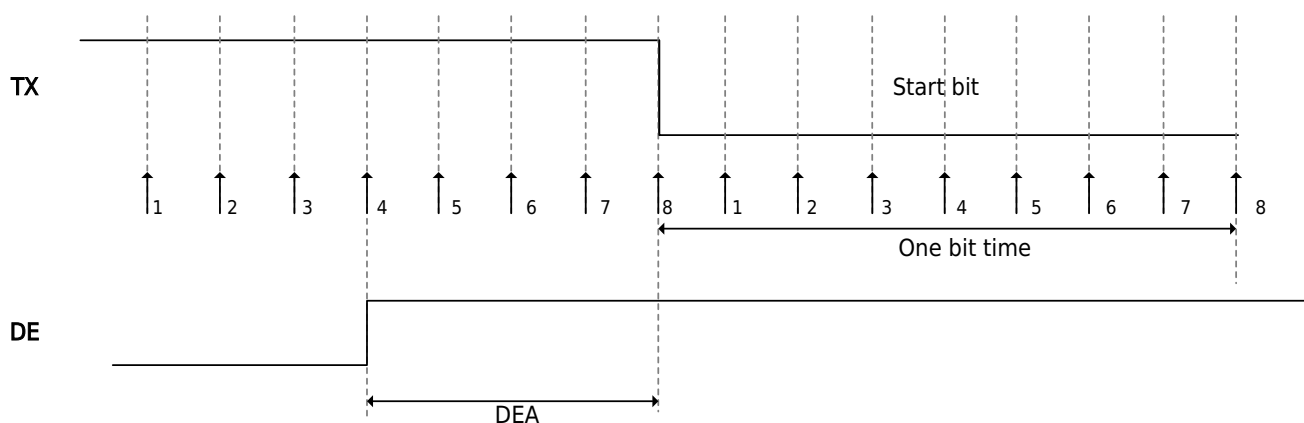


Figure 24-9 OVER8=1 DE Asserted

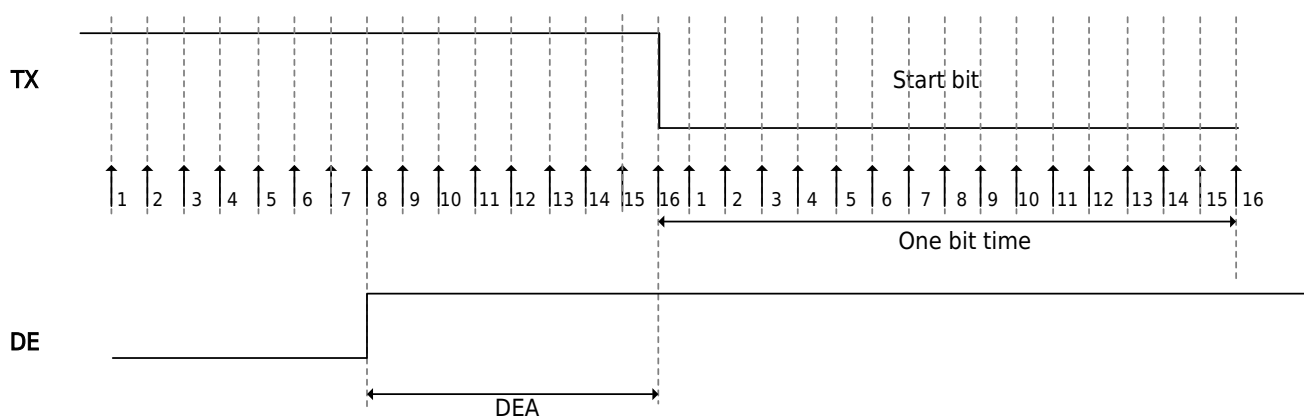


Figure 24-10 OVER8=0 DE Asserted

The de-assertion time programmed by the USARTn_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 USARTn_CR3.DEP.

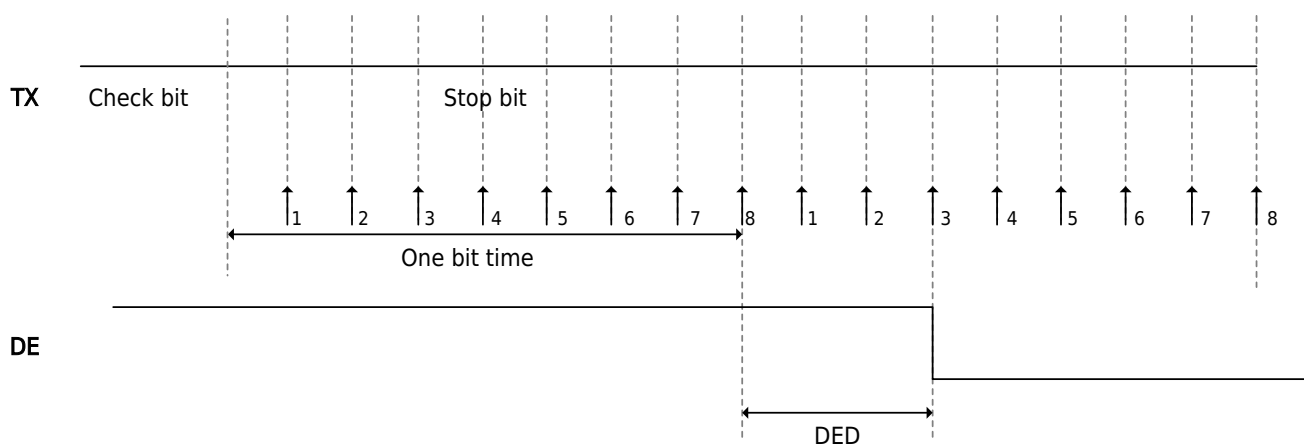


Figure 24-11 OVER8=1 DE De-assertion

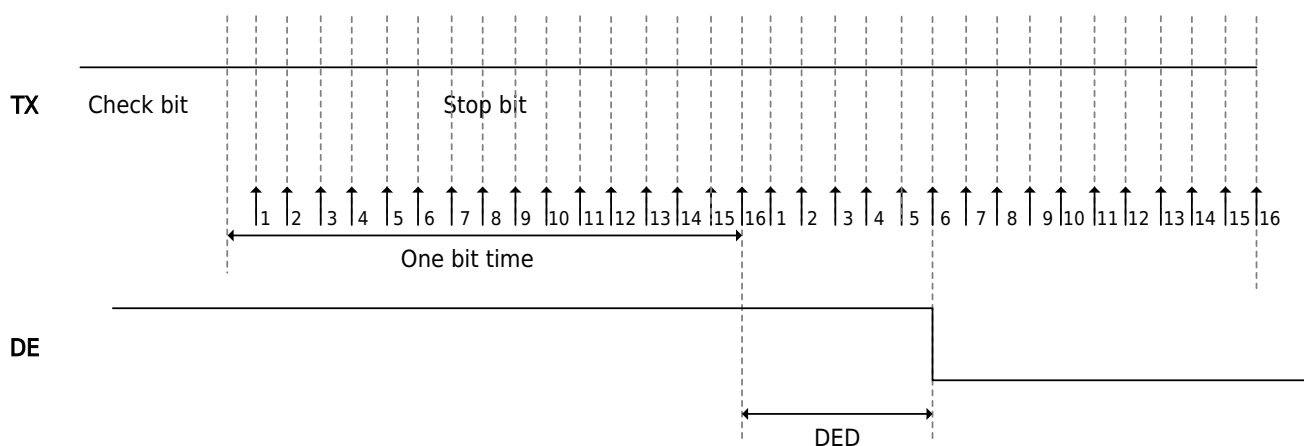


Figure 24-12 OVER8=0 DE De-assertion

24.3.2.10 Interrupt and Events

Table 24-4 UART Interrupt/Event Table

Function Name	Enable Bit (Interrupt only)	Flag Bit	Used as Event Source
Receive Error Interrupt	RIE	ORE, FE, PE	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
TIMEOUT Interrupt	RTOIE	RTOF	Enable
RX Line Wake-up Stop Mode Interrupt	INTC_WKEN.USARTnWKEN	INTC_WKEN.USARTnWKF	Disable

24.3.3 Multiprocessor Communication

24.3.3.1 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.

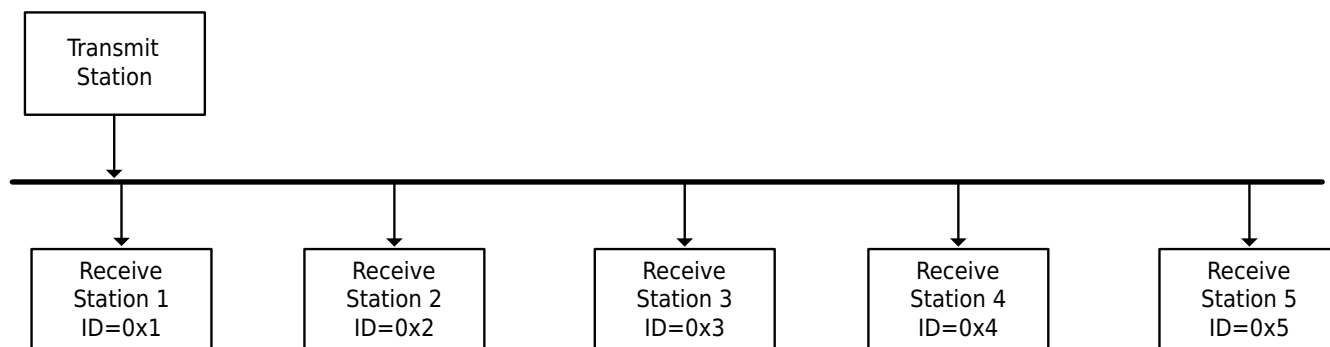


Figure 24-13 Multiprocessor Communication Legend

24.3.3.2 Data Format

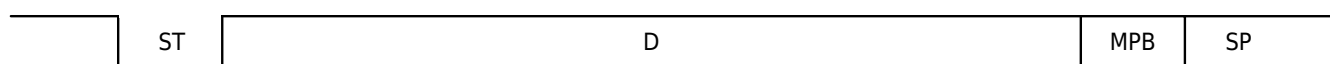


Figure 24-14 Multiprocessor Mode Data Format

- **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.

24.3.3.3 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.

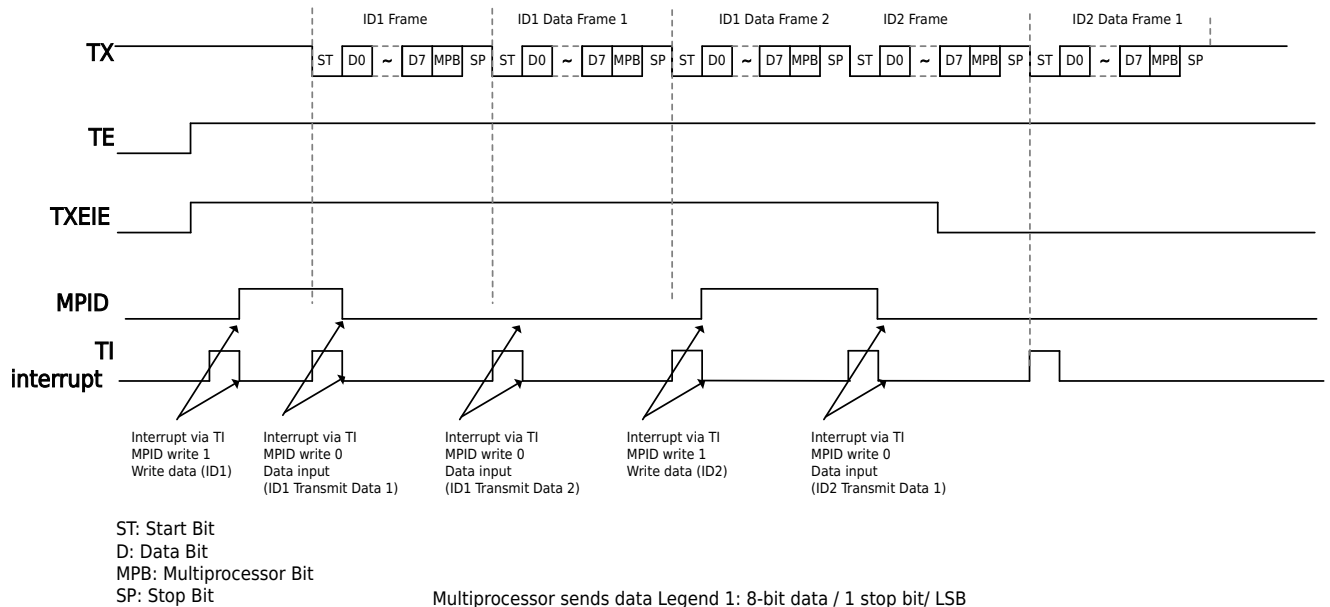


Figure 24-15 Multiprocessor Mode Transmit Data Diagram

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.

- If the ID matches, receive data normally, an interrupt occurs, and error detection is performed, which is the same as UART receive data.
- If the ID does not match, the software writes the USARTn_CR1.SLME bit to 1 again and repeats step 8.

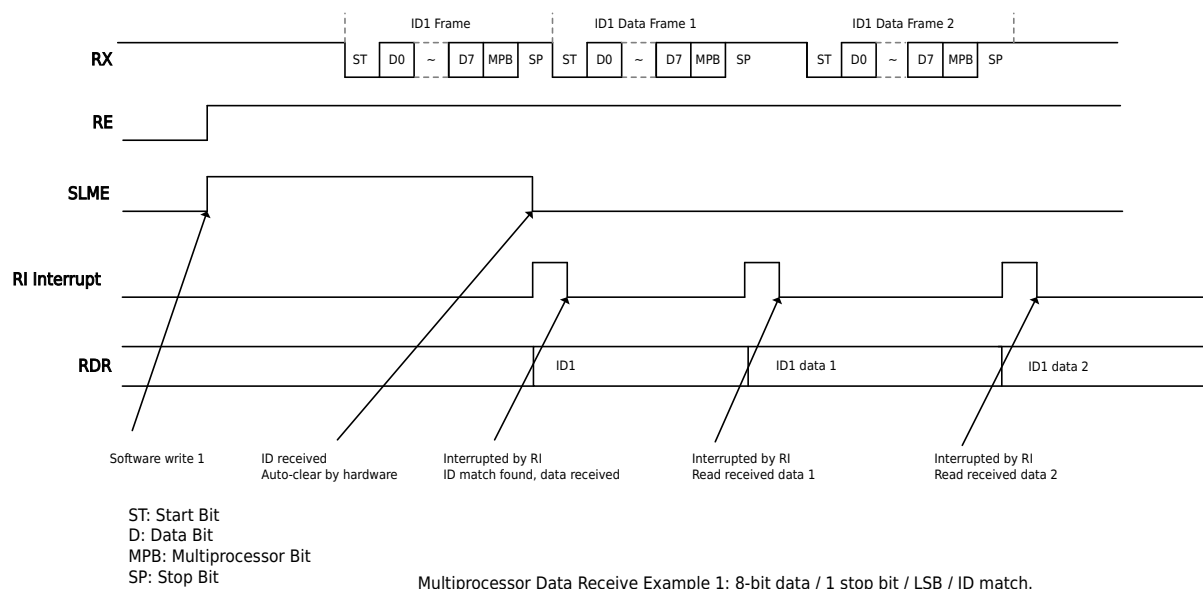


Figure 24-16 Multiprocessor Mode Receive Data Diagram 1

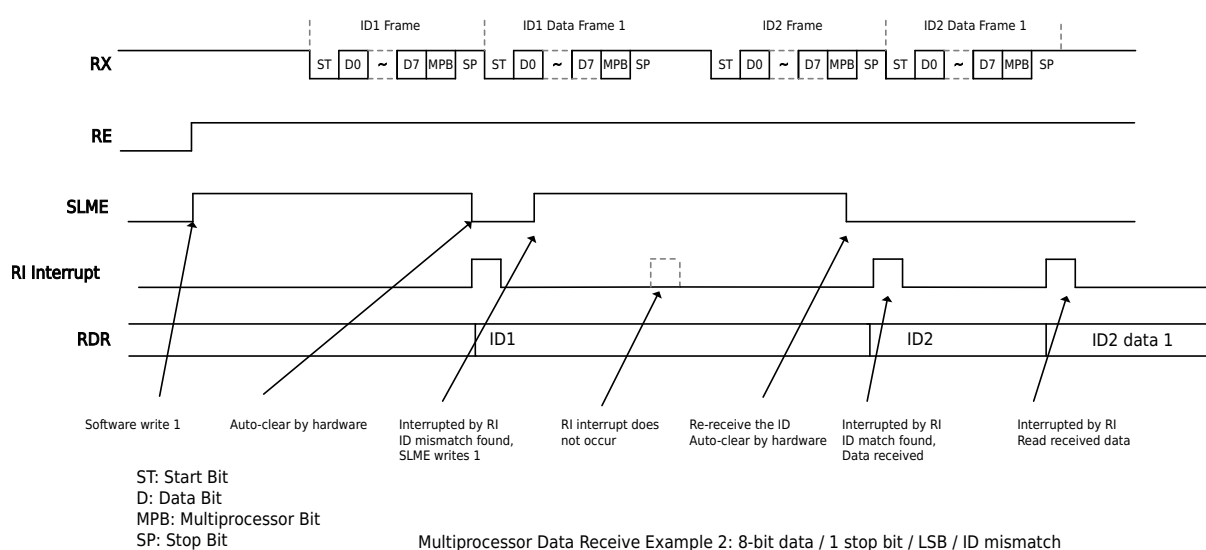


Figure 24-17 Multiprocessor Mode Receive Data Diagram 2

24.3.3.4 Interrupt and Events

In multiprocessor mode, the interrupt handling is identical to UART mode except for the absence of parity errors.

Table 24-5 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

Function Name	Enable Bit (Interrupt only)	Flag	Used as Event Source
Transmit Data Register Empty Interrupt	TXEIE	TXE	Enable
Transmit Completion Interrupt	TCIE	TC	Enable
Transmit Completion Interrupt	TENDIE	TEND	Enable

24.3.4 UART-LIN

24.3.4.1 Function Introduction

LIN stands for Local Interconnect Network, a low-speed (1-20kbps) serial communication protocol designed to reduce automotive network costs.

24.3.4.2 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.

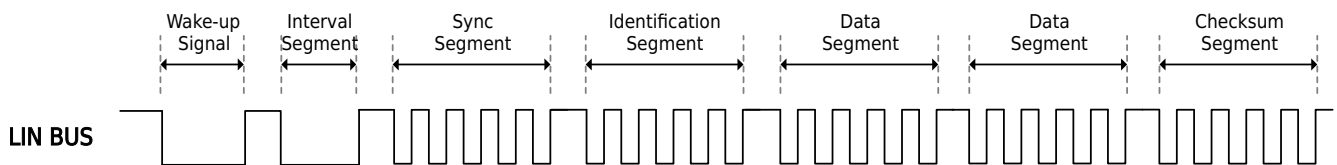


Figure 24-18 LIN Bus Data Behavior

24.3.4.3 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.

After the transmission of the wake-up frame is completed, the transmission of the gap interval segment will proceed.

- 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.

24.3.4.4 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.

After the USARTn_LBMC register update is completed, the USARTn_SR.LBM flag bit will be set. When USARTn_CR2.LBMIE=1, the corresponding interrupt is generated.

Note that during synchronization segment detection, the USARTn_SR.FE flag may be set. When clearing the USARTn_SR.LBM flag, the USARTn_SR.FE flag also needs to be cleared; reading the USARTn_LBMC register must be performed after the synchronization segment reception is completed.

Do not enable the digital filter function during synchronization segment detection.

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. After completing the synchronization segment reception, read the USARTn_LBMC register to calculate the baud rate;
- b. Clear the RE and TE bits in USARTn_CR1, and configure the correct baud rate via the USARTn_BRR and USARTn_PR registers;
- c. Set USARTn_CR1.RE bit again;
- d. Clear the USARTn_SR.FE/LBM flag and read the RDR register to enable normal data reception later;
- e. If receive interrupts are required for later data reception, set USARTn_CR1.RIE=1.

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).

24.3.4.5 Interrupt and Events

Table 24-6 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/Synchronization Segment Detection Interrupt	WKUPIE	WKUP	Enable
	LBDIE	LBD	Enable
	LBMIE	LBM	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

24.3.5 Smart Card

24.3.5.1 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.

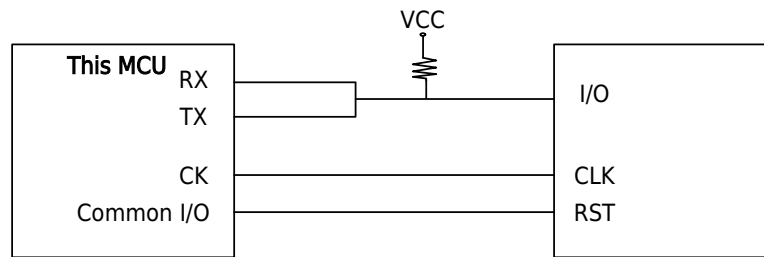


Figure 24-19 Smart Card Connection Schematic Diagram

24.3.5.2 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 (HCLK, HCLK/4, HCLK/16, HCLK/64), unit: MHz
When USARTn_PR.ULBREN is 1, the clock set by USARTn_PR.PSC bit (HCLK/128, HCLK/256, HCLK/512, HCLK/1024), unit: MHz
- DIV_Integer: USARTn_BRR.DIV_Integer setting value
- BCN: USARTn_CR3.BCN register setting value

When C is HCLK, DIV_Integer=0, and BCN=0, the baud rate reaches the maximum baud rate of HCLK/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.

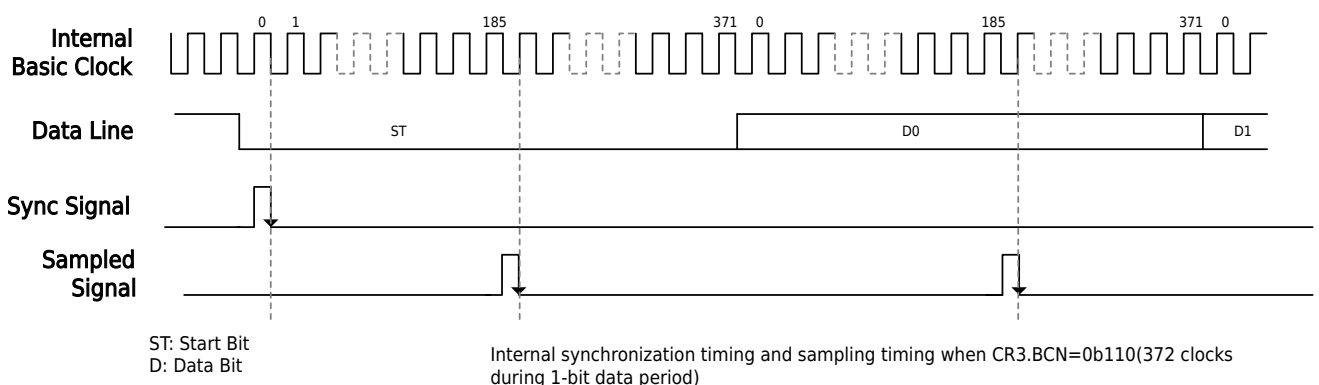


Figure 24-20 Smart Card Mode Synchronization Timing and Sample Timing Diagram

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

24.3.5.3 Data Format

In smart card mode, a frame of data consists of start bit, data bit and check bit.

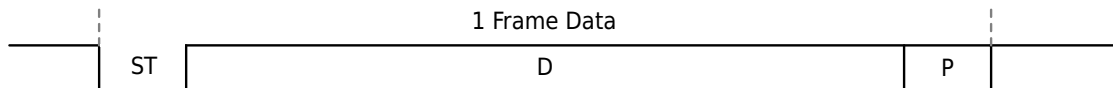


Figure 24-21 Smart Card Mode Data Format

- **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.

24.3.5.4 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.

24.3.5.5 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.

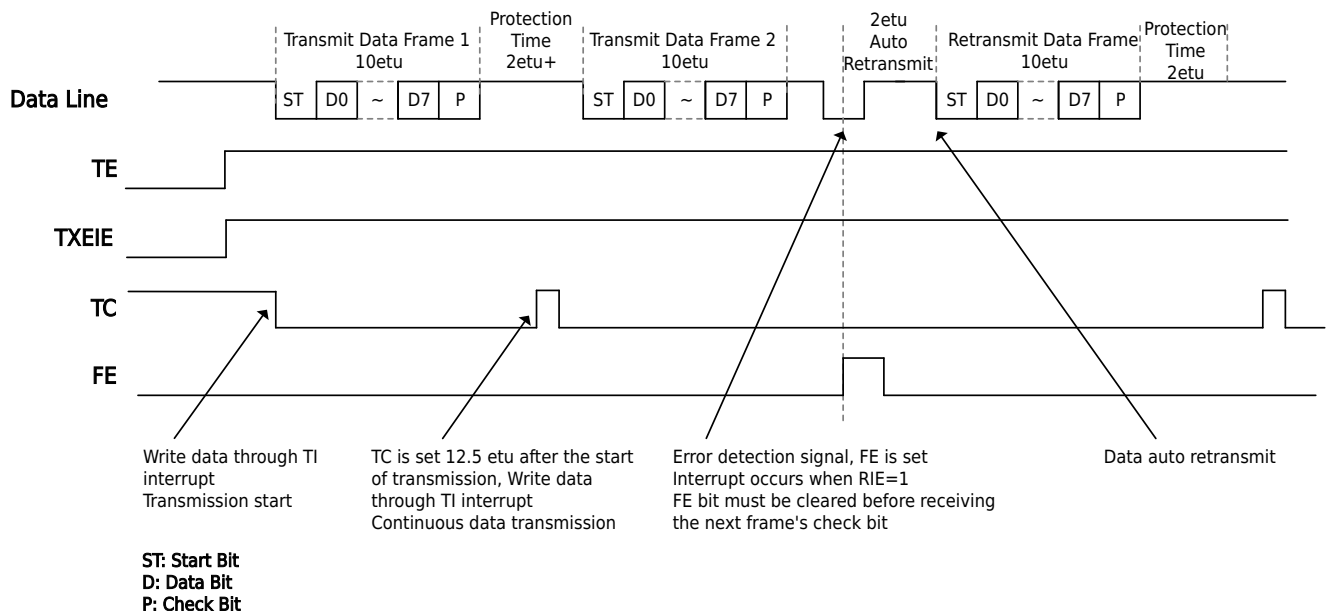


Figure 24-22 Smart Card Mode Transmit Data Diagram

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.

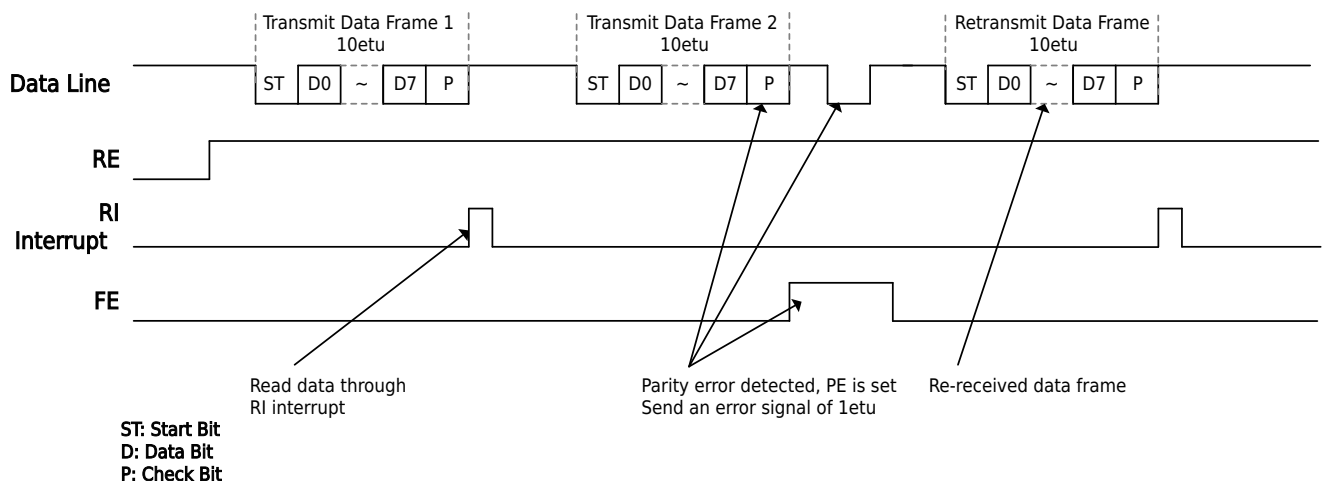


Figure 24-23 Smart Card Mode Receive Data Diagram

24.3.5.6 Interrupt and Events

Table 24-7 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

24.3.6 Clock Synchronization Mode

24.3.6.1 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 (HCLK, HCLK/4, HCLK/16, HCLK/64), unit: MHz
When USARTn_PR.ULBREN is 1, the clock set by USARTn_PR.PSC bit (HCLK/128, HCLK/256, HCLK/512, HCLK/1024), unit: MHz
- DIV_Integer: USARTn_BRR.DIV_Integer setting value

When the internal clock source is selected and C is HCLK with DIV_Integer=1, the maximum baud rate is HCLK/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 HCLK/6 (MHz), so the maximum baud rate is HCLK/6 (Mbps).

Please note that in synchronous mode, the maximum communication baud rate is not only determined by the HCLK-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.

24.3.6.2 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.

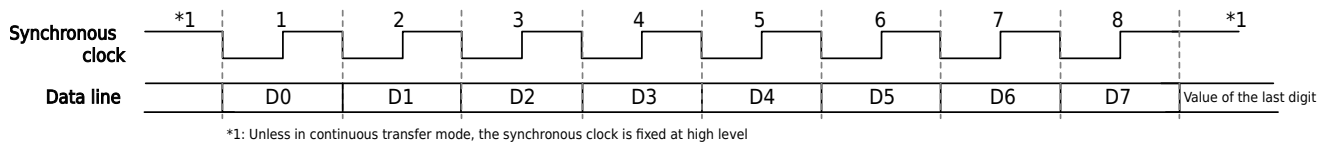


Figure 24-24 Clock Synchronization Mode Data Format

24.3.6.3 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.

24.3.6.4 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.

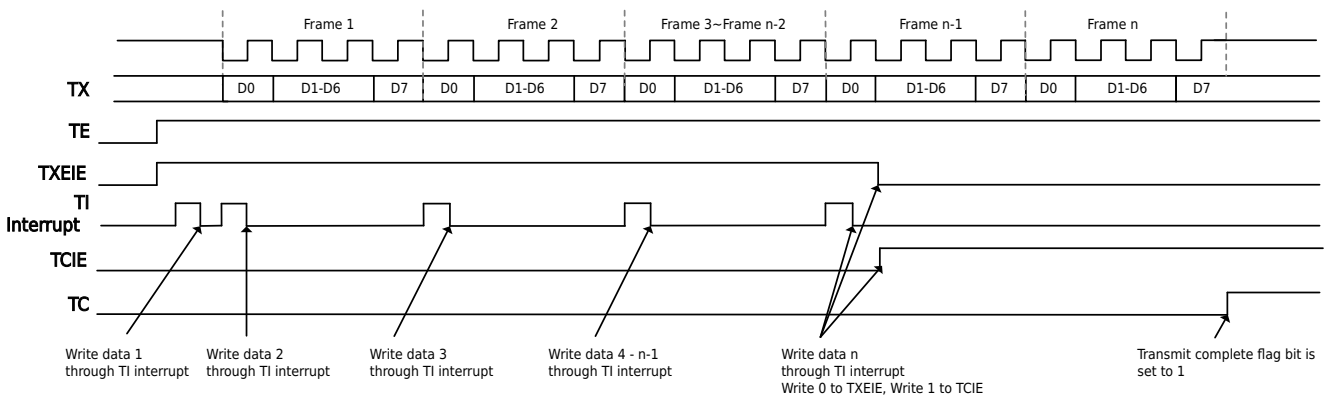


Figure 24-25 Data Transmission in Clock Synchronization Mode Diagram 1 (LSB/without CTS function)

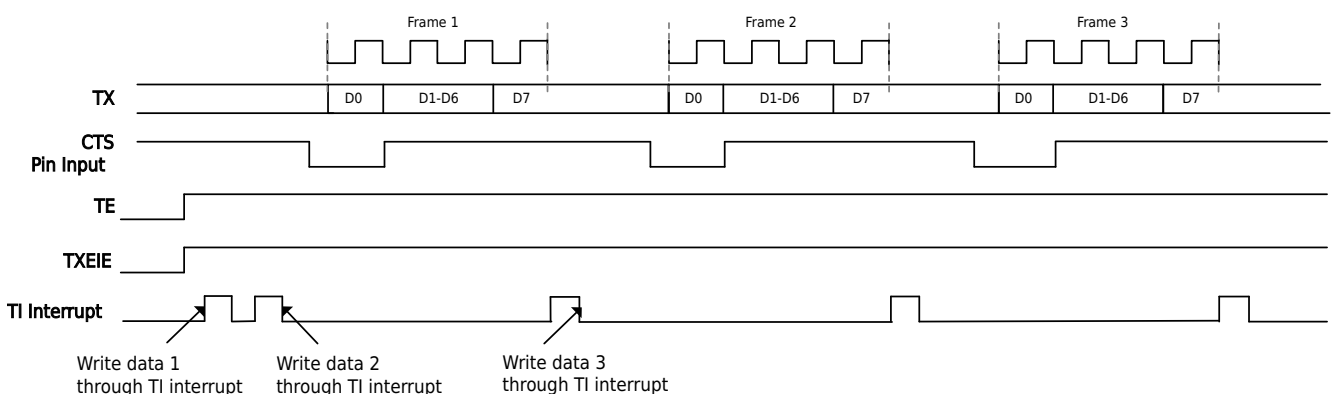


Figure 24-26 Data Transmission in Clock Synchronization Mode Diagram 2 (LSB/with CTS function)

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.

24.3.6.5 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.

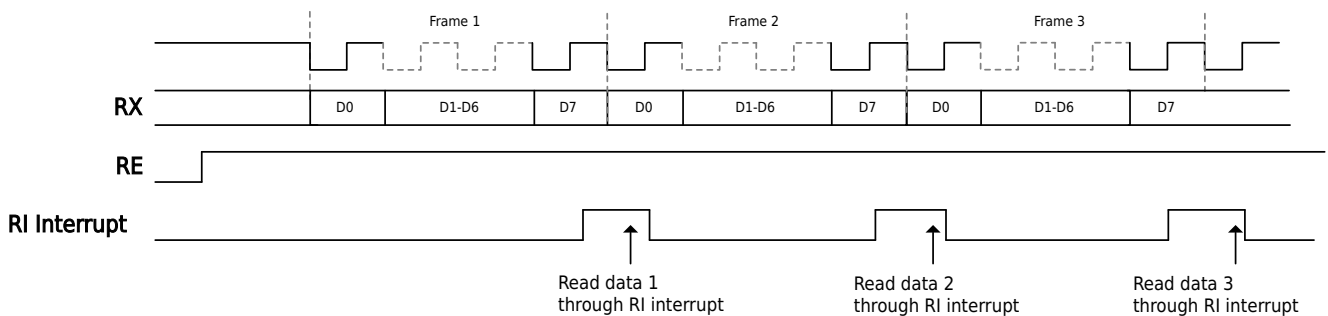


Figure 24-27 Data Reception in Clock Synchronization Mode Diagram 1 (LSB/without RTS function)

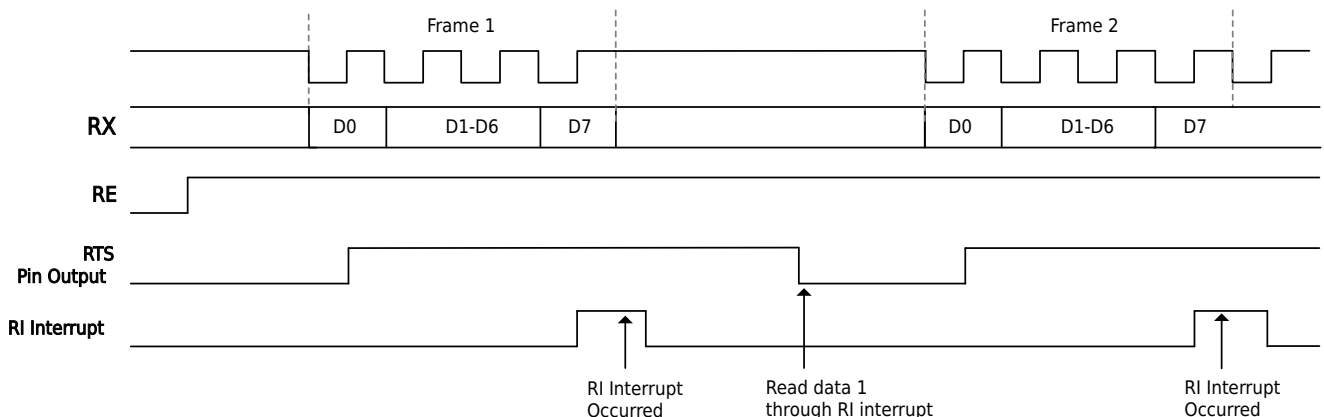


Figure 24-28 Data Reception in Clock Synchronization Mode Diagram 2 (LSB/with RTS function)

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.

24.3.6.6 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.

24.3.6.7 Interrupt and Events

Table 24-8 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

24.3.7 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.

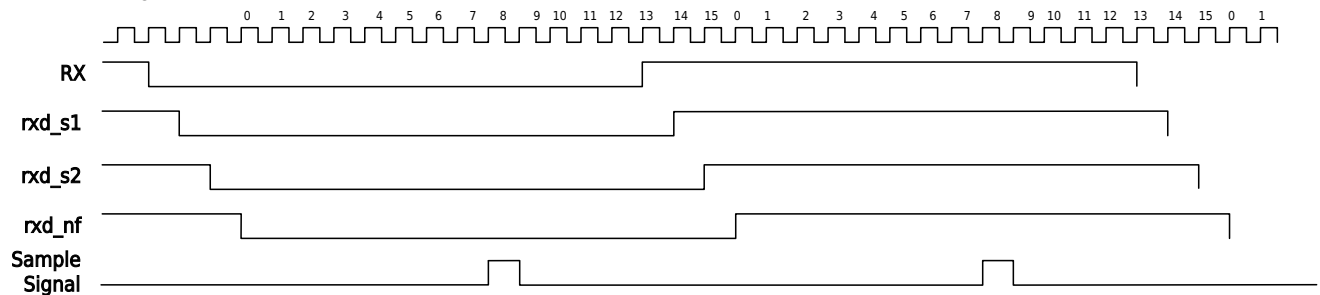
When the fractional baud rate function is enabled (USARTn_CR1.FBME=1), the actual filtering capability should account for clock frequency errors introduced by fractional division. Refer to [Table 24-20 : Baud Rate Calculation Formula \(Fractional Baud Rate is Enable where FBME=1\)](#) for error calculation methods.

When operating in LIN mode, enable the digital filtering function after the synchronization frame.

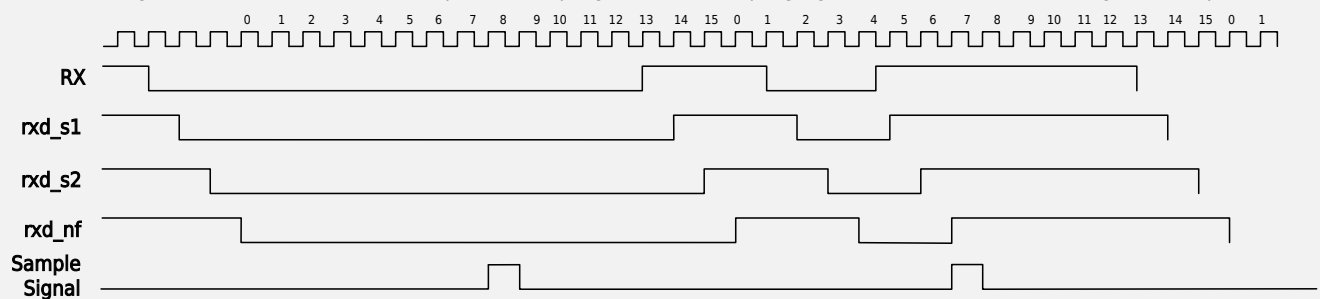
If the digital filter's clock stops and then starts again, the digital filter resumes operation from the state maintained when the clock stopped.

When USARTn_CR.TE=0 and USARTn_CR.RE=0, the Flip-Flop state inside the digital filter is reset to 1.

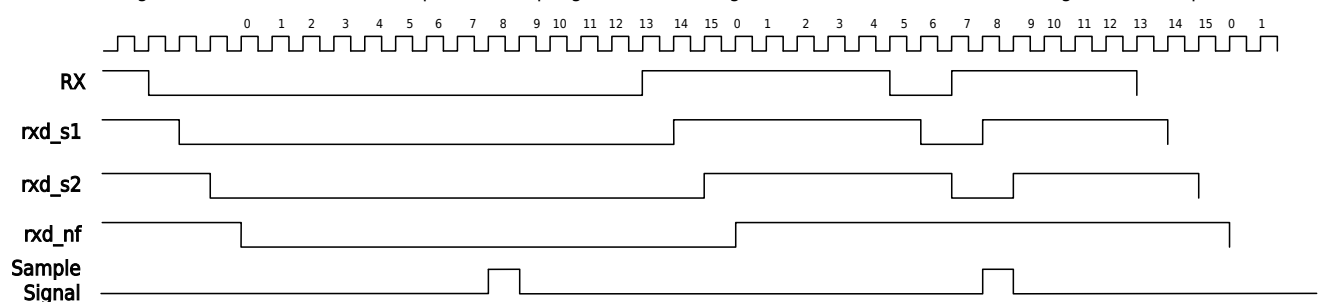
Enable Filtering, No Interference



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)

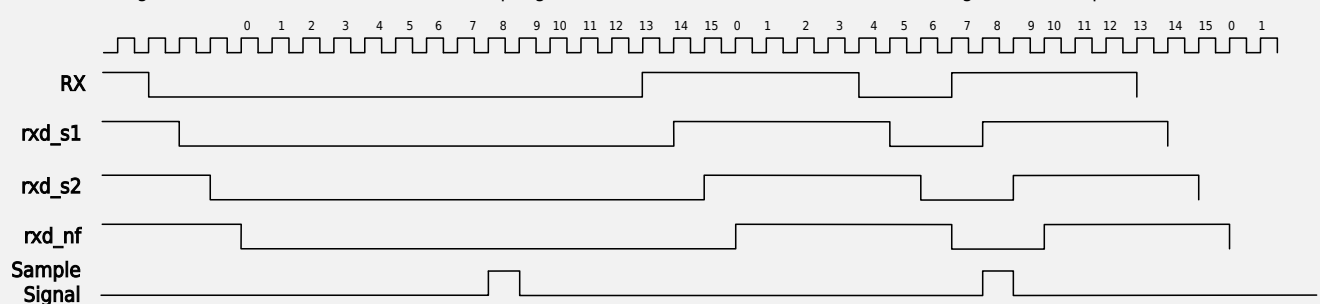


Figure 24-29 Digital Filter Function Diagram

- rxd_s1: internal signal RX delayed by 1 sampling clock
 - 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.

24.3.8 Interrupt

Table 24-9 USART Global Interrupt Summary Table

Interrupt Name	Symbol	Description
Error Interrupt	USART_n_EI n=1-4	UART/Multiprocessor/Smart Card/Clock Synchronization Mode Receive Error Interrupt LIN Error interrupt
Receive Data Register Full Interrupt	USART_n_RI n=1-4	UART/Multiprocessor/LIN/Smart Card/Clock Synchronization Mode Receive Data Register Full Interrupt LIN Error interrupt
Transmit Data Register Empty Interrupt	USART_n_TI n=1-4	UART/Multiprocessor/LIN/Smart Card/Clock Synchronization Mode Transmit Data Register Empty Interrupt
Transmit Completion Interrupt	USART_n_TCI n=1-4	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 n=1-4	UART/Multiprocessor/LIN/Smart Card/Clock Synchronization Mode Transmit Completion Interrupt (Writing 1 to USARTn_CR1.CTEND bit or writing transmit data to Transmit Data Register clears the interrupt)
Wake-up Signal/Interval Segment/Synchronization Segment Detection Interrupt	USART_n_BRKWKPI n=3/4	LIN Wake-up Signal/Interval Segment /Synchronization Segment Detection Interrupt
UART Receive TIMEOUT Interrupt	USART_n_RTO n=1-4	UART Receive TIMEOUT Interrupt
RX Line Wake-up Stop Mode Interrupt	USART_n_WUPI n=3/4	USART RX Line Wake-up Stop Mode Interrupt

24.4 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.

24.4.1 Register List

- USART1 base address: 0x40001800
- USART2 base address: 0x40001C00
- USART3 base address: 0x40002000
- USART4 base address: 0x40002400

Table 24-10 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 Baudrate Measuring Counter Register	32

24.4.2 Status Register (USART_SR)

Offset	Bit Position															
0x00	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reset	0x000000C0															
Name	Reserved												MPB	Reserved		LBM
Access													RO			RO
														TEND	LBD	WKUP
														RTOF	TXE	TC
														RXNE	BE	ORE
														Reserved	FE	PE

Bit	Name	Function	Description	Access
31:17	Reserved	Reserved	-	-
16	MPB	Multiprocessor Bit Flag	0b0: Current receive data is communication data 0b1: Current receive data is ID  Note The MPB bit is only valid in multiprocessor mode.	RO
15:13	Reserved	Reserved	-	-
12	LBM	LIN Synchronization Segment Detection Flag	Only valid in USART3/USART4: 0b0: Synchronization segment not detected 0b1: Synchronization segment detected LBM set condition: - When a synchronization segment is detected and the LBMC register update is completed. LBM clear condition: - Writing 1 to the USART_CR1.CLBM bit.  Note RE=0 does not reset the LBM bit.	RO
11	TEND	Transmit Completion Flag	0b0: Transmit data in progress 0b1: Data transmission complete UART Mode, Multiprocessor Mode, LIN Mode, Clock Synchronization Mode - TEND set condition: - When the last bit of a frame is transmitted, the value in the transmit data register is not updated. - TEND clear condition: - When TE=1, write transmit data to the transmit data register. - Write 1 to the USART_CR1.CTEND bit. Smart Card Mode - TEND set condition: - After the last byte of data is transmitted, FE becomes 0 after a specific time period, and	RO

Bit	Name	Function	Description	Access
11	TEND	Transmit Completion Flag	the value in the transmit data register is not updated. The specific timing for TEND set is: 2.5 bit durations after the check bit is transmitted. - TEND clear condition: - When TE=1, write transmit data to the transmit data register. - Write 1 to the USART_CR1.CTEND bit.	RO
10	LBD	LIN Interval Segment Detection Flag	Only valid in USART3/USART4: 0b0: No interval segment detected 0b1: Interval segment detected LBD set condition: - 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 USART_CR2.LBDL register, and a break delimiter is detected. LBD clear condition: - Write 1 to the USART_CR1.CLBD bit.  Note RE=0 does not reset the LBD bit.	RO
9	WKUP	LIN Wakeup Signal Detection Flag	Only valid in USART3/USART4: 0b0: No Wakeup signal detected 0b1: Wakeup signal detected WKUP set condition: - When USART_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. WKUP Clear condition: - Write 1 to the USART_CR1.CWKUP bit.  Note RE=0 does not reset the WKUP bit. After detecting the WKUP signal as 1, it indicates that this LIN node has exited Sleep mode. After setting register USART_CR2.WKUPE=0, perform subsequent operations such as interframe space detection.	RO

Bit	Name	Function	Description	Access
8	RTOF	UART Receive TIMEOUT Flag Bit	● 0b0: No UART receive TIMEOUT ● 0b1: UART receive TIMEOUT occurred RTOF set condition: ● From the detection of the STOP bit of the last frame data, no new Receive Data is detected after the set time has elapsed. RTOF clear condition: ● Write 1 to the USART_CR1.CRTOF bit.  Note RTOF is set to 1 by hardware, and is only set to 1 when both USART_CR1.RE=1 and USART_CR1.RTOE=1. When USART_CR1.RE=0, the TIMEOUT function is enabled, but RTOF is not set to 1.	RO
7	TXE	Transmit Data Register Empty Flag	● 0b0: Transmit data register is not empty, the hardware sets TXE to 0 ● 0b1: Transmit data register is empty, the hardware sets TXE to 1  Note The TXE bit is valid in UART/Clock synchronization mode/LIN Mode.	RO
6	TC	Transmit Completion Flag	● 0b0: Transmit data in progress ● 0b1: Data transmission complete UART Mode, Multiprocessor Mode, LIN Mode, Clock Synchronization Mode ● TC set condition: ▶ TE=0: Transmission Disabled ▶ When the last bit of a frame is transmitted, the value in the Transmit Data Register is not updated. ● TC clear condition: ▶ When TE=1, write transmit data to the transmit data register. Smart Card Mode ● TC set condition: ▶ TE=0: Transmission Disabled ▶ 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. The specific timing for TC set is: 2.5 bit durations after the check bit is transmitted. ● TC clear condition: ▶ When TE=1, write transmit data to the transmit data register.	RO

Bit	Name	Function	Description	Access
6	TC	Transmit Completion Flag	 Note When the TE bit changes from 0 to 1, TC remains at 1.	RO
5	RXNE	Receive Data Register not Empty Flag	● 0b0: No data received ● 0b1: Ready to read received data 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.	RO
4	BE	LIN Bus Error Flag	Only valid in USART3/USART4: ● 0b0: No bus error. ● 0b1: Bus error occurred BE set condition: ● 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). BE clear condition: ● Write 1 to the USART_CR1.CBE bit.  Note TE=0 does not reset the BE bit. 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.	RO
3	ORE	Receive Overflow Error Flag	● 0b0: No receive overflow error ● 0b1: Receive overflow error occurred ORE set condition: ● New data frame received while the receive data register remains unread. ORE clear condition: ● Write 1 to the USART_CR1.CORE bit.  Note RE=0 does not reset the ORE bit. Data received before ORE=1 will be retained, while data received when ORE=1 will be discarded. When ORE=1, it cannot continue to receive data, and transmit data is also prohibited in clock synchronization mode.	RO
2	<i>Reserved</i>	<i>Reserved</i>	-	-

Bit	Name	Function	Description	Access
1	FE	Receive Frame Error Flag	● 0b0: No receive frame error ● 0b1: Receive frame error occurred UART Mode, Multiprocessor Mode, LIN Mode ● FE set condition: ▶ The stop bit of the received data frame is at low level. For two stop bits, only the first stop bit is checked. ● FE clear condition: ▶ Write 1 to the USART_CR1.CFE bit.  Note RE=0 does not reset the FE bit. When FE=1, the received data is retained but the RI interrupt does not occur. After FE=1, receive data is disabled. 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. Smart Card Mode ● FE set condition: ▶ Sampled low level error signal flag. ● FE clear condition: ▶ Write 1 to the USART_CR1.CFE bit.  Note RE=0 does not reset the FE bit.	RO
0	PE	Receive Data Check Error Flag	● 0b0: No receive data parity error ● 0b1: Receive data parity error occurred PE set condition: ● A parity check error occurs when receiving data. PE clear condition: ● Write 1 to the USART_CR1.CPE bit.  Note RE=0 does not reset the PE bit. When PE=1, the received data is retained but the RI interrupt does not occur. receive data is disabled after PE=1 is set.	RO

24.4.3 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: 0b0: Transmit data 0b1: Transmit ID  Note The MPID bit is only valid in multiprocessor mode and must be set to the reset value in other modes.	RW
8:0	TDR	Transmit Data	Transmit Data Register  Note The most significant bit TDR[8] is only valid in UART Mode when the data length is set to 9 bits.	RW

24.4.4 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  Note The most significant bit RDR[8] is only valid in UART Mode when the data length is set to 9 bits.	RO

24.4.5 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  Note 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  Note 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 24-19 Baud Rate Calculation Formula (Fractional Baud Rate is disable where FBME=0)

Mode	Baud Rate Calculation Formula	Error E Calculation Formula
UART Mode/ 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 24-20 Baud Rate Calculation Formula (Fractional Baud Rate is Enable where FBME=1)

Mode	Baud Rate Calculation Formula	Calculation Formula for Error E (%)
UART Mode/ 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 USART_PR.ULBREN and USART_PR.PSC bits
- BCN: USART_CR3.BCN register setting value

24.4.6 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	CLBM	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	In UART Mode, the start bit detection setting for receive data: ● 0b0: Start bit detection method is low level on RX pin ● 0b1: Start bit detection mode is falling edge on RX pin  Note This bit must retain the reset value in non-USART Mode. This bit can only be set when TE=0 & RE=0 (transmit/receive disabled).	RW
30	NFE	Digital Filtering Enable Bit	● 0b0: Disable ● 0b1: Enable  Note This bit must retain the reset value in Clock Synchronization Mode and synchronization frame detection of LIN Mode. This bit can only be set when TE=0 & RE=0 (transmit/receive disabled).	RW
29	FBME	Fractional Baud Rate Function Enabled	● 0b0: Disable ● 0b1: Enable  Note This bit can only be set when TE=0 & RE=0 (transmit/receive disabled).	RW
28	ML	MSB/LSB Selection Bit	In UART Mode/Clock Synchronization Mode/Smart Card Mode, MSB/LSB selection bit: ● 0b0: LSB mode ● 0b1: MSB mode  Note This bit can only be set when TE=0 & RE=0 (transmit/receive disabled).	RW

Bit	Name	Function	Description	Access
27	<i>Reserved</i>	<i>Reserved</i>	-	-
26	CLBM	LBM Clear Bit	Only valid in USART3/USART4: Write 1 to clear LBM flag, write 0 has no effect; read always returns 0.	RW
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	● 0b0: UART Mode ● 0b1: Clock synchronization mode  Note This bit can only be set when TE=0 & RE=0 (transmit/receive disabled) .	RW
23	CLBD	LBD Clear Bit	Only valid in USART3/USART4: Write 1 to clear LBD flag, write 0 has no effect; read always returns 0.	RW
22	CWKUP	WKUP Clear Bit	Only valid in USART3/USART4: Write 1 to clear WKUP flag, write 0 has no effect; read always returns 0.	RW
21	CBE	BE Clear Bit	Only valid in USART3/USART4: 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: ● 0b0: 16 bits ● 0b1: 8 bits  Note This bit must retain the reset value in Smart Card Mode/Clock Synchronization Mode. This bit can only be set when TE=0 & RE=0 (transmit/receive disabled).	RW
14:13	<i>Reserved</i>	<i>Reserved</i>	-	-

Bit	Name	Function	Description	Access
12	M	Data Length Setting Bit	In UART Mode, transmit/receive data length setting bit: ● 0b0: 8 bits ● 0b1: 9 bits  Note This bit must retain the reset value in LIN Mode/Smart Card Mode/Clock Synchronization Mode. This bit can only be set when TE=0 & RE=0 (transmit/receive disabled).	RW
11	Reserved	Reserved	-	-
10	PCE	Check Enable Bit	In UART Mode, the parity check enable bit: ● 0b0: No check ● 0b1: Check  Note 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. This bit can only be set when TE=0 & RE=0 (transmit/receive disabled).	RW
9	PS	Check Bit	In UART Mode, the parity check selection bit: ● 0b0: Even parity ● 0b1: Odd parity  Note This bit is valid only when PCE=1. This bit must retain 0 in Smart Card Mode. This bit can only be set when TE=0 & RE=0 (transmit/receive disabled).	RW
8	TENDIE	Transmit Completion Interrupt Enable Bit	● 0b0: TENDI interrupt request is disabled, no TENDI interrupt occurs ● 0b1: TENDI interrupt request is enabled, TENDI interrupt occurred	RW
7	TXEIE	Transmit Data Register Empty Interrupt Enable Bit	● 0b0: TI interrupt request is disabled, no TI interrupt occurs ● 0b1: TI interrupt request is enabled, TI interrupt occurred  Note When TE=0 and TXEIE=1 is written, it is necessary to wait for the TI interrupt to occur when TE=1. When TE=1 and writing TXEIE=1, it is necessary to wait until SR.TC=1 before the write can be performed.	RW

Bit	Name	Function	Description	Access
6	TCIE	Transmit Completion Interrupt Enable Bit	0b0: TCI interrupt request is disabled, no TCI interrupt occurs 0b1: TCI interrupt request enable, TCI interrupt occurred  Note Read is valid in non-Smart Card Mode, reads as 0 in Smart Card Mode.	RW
5	RIE	Receive Interrupt, Error Interrupt Enable Bit	0b0: Receive interrupt and error Interrupt requests are disabled; no RI and EI Interrupts occur 0b1: Receive interrupt and error interrupt requests are enabled, RI and EI interrupts occurred	RW
4	SLME	Silent Mode Enable Bit	In multiprocessor operations, the silent mode enable bit: 0b0: Normal mode 0b1: Silent mode When SLME=1, communication data with MPB bit set to 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 ID data with the MPB bit set to 1 is received, SLME is automatically cleared and starts the normal data reception Action.  Note This bit is only valid in UART Multiprocessor Communication Mode. In other modes, this bit must retain the reset value.	RW
3	TE	Transmitter Enable Bit	0b0: Transmitter disable 0b1: Transmitter enable  Note In Clock Synchronization Mode, this bit can only be written as 1 when TE=0 & RE=0 (transmit/receive disabled).	RW
2	RE	Receiver Enable bit	0b0: Receiver disable 0b1: Receiver enable  Note In Clock Synchronization Mode, this bit can only be written as 1 when TE=0 & RE=0 (transmit/receive disabled).	RW
1	RTOIE	UART TIMEOUT Interrupt Enable Bit	0b0: UART TIMEOUT interrupt request is disabled, no RTOI interrupt occurs 0b1: UART TIMEOUT interrupt request is enabled, RTOI interrupt occurred	RW

Bit	Name	Function	Description	Access
0	RTOE	UART TIMEOUT Function Enable Bit	● 0b0: UART TIMEOUT function disable ● 0b1: UART TIMEOUT function enable	RW

24.4.7 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												LBMIE	Reserved	SBKM	SBK	UATSBKE	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	RW	RW	RW	RW	RW	RW	RW

Bit	Name	Function	Description	Access															
31:20	Reserved	Reserved	-	-															
19	LBMIE	LIN Synchroni- zation Segment Interrupt Ena- ble Bit	Only valid in USART3/USART4: ● 0b0: LIN synchronization segment detection in- terrupt request is invalid; no interrupt occurs. ● 0b1: LIN synchronization segment detection in- terrupt request is valid; an interrupt occurs.  Note This bit must retain the reset value in Non-UART-LIN mode.	RW															
18	Reserved	Reserved	-	-															
17	SBKM	Interval Seg- ment Transmis- sion Mode	Only valid in USART3/USART4: For specific settings, refer to the SBK bit descrip- tion.	RW															
16	SBK	Interval Seg- ment Transmis- sion Enable Bit	Only valid in USART3/USART4 <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>Write 1 to SBK when TE=1, the interval segment is automati- cally transmitted. After trans- mission completes, SBK is auto- matically cleared.</td></tr><tr><td>0b1</td><td>0b1</td><td>Write data 0x00 to USART_TDR.TDR to start send- ing the interval segment. After transmission is complete, SBK is automatically cleared.</td></tr></table>	SBK	SBKM	Interval segment transmission	0b0	0b0	Do not send interval segment	0b0	0b1	Do not send interval segment	0b1	0b0	Write 1 to SBK when TE=1, the interval segment is automati- cally transmitted. After trans- mission completes, SBK is auto- matically cleared.	0b1	0b1	Write data 0x00 to USART_TDR.TDR to start send- ing 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	Write 1 to SBK when TE=1, the interval segment is automati- cally transmitted. After trans- mission completes, SBK is auto- matically cleared.																	
0b1	0b1	Write data 0x00 to USART_TDR.TDR to start send- ing the interval segment. After transmission is complete, SBK is automatically cleared.																	

Bit	Name	Function	Description	Access
15	UATSBKE	UART Mode Break Detection Enable Bit	Only valid in USART3/USART4: ● 0b0: Disable break detection in UART mode ● 0b1: Enable break detection in UART mode  Note This bit can only be set when TE=0 & RE=0 (transmit/receive disabled).	RW
14	LINEN	LIN Function Enabled	Only valid in USART3/USART4: ● 0b0: Disable LIN Function ● 0b1: Enable LIN function  Note This bit can only be set when TE=0 & RE=0 (transmit/receive disabled).	RW
13	STOP	Stop Bit Setting Bit	In UART Mode, stop bit length setting bit: ● 0b0: 1 stop bit ● 0b1: 2 stop bits  Note This bit must retain the reset value in LIN Mode/ Smart Card Mode/Clock Synchronization Mode. This bit can only be set when TE=0 & RE=0 (transmit/receive disabled).	RW

Bit	Name	Function	Description	Access
12:11	CLKC	Clock Control Bit	UART Mode 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. 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. 0b10/0b11: Clock source is external input clock, the input clock frequency is 16 times the baud rate (USART_CR1.OVER8=0) or 8 times the baud rate (USART_CR1.OVER8=1). Clock Synchronization Mode 0b00/0b01: Clock source is the clock generated by the internal baud rate generator and output to the USARTn_CK pin. 0b10/0b11: Clock source is external input clock, the input clock frequency equals baud rate. Smart Card Mode 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. 0b01: Clock source is the clock generated by the internal baud rate generator and output to the USARTn_CK pin. 0b10/0b11: Setting prohibited  Note This bit can only be set when TE=0 & RE=0 (transmit/receive disabled). When both TE and RE are cleared to 0, the clock output on the USARTn_CK pin is disabled.	RW
10:9	Reserved	Reserved	All bits read as 1, Write as 1	-
8	WKUPE	LIN Wake-up Signal Detection Enable Bit	Only valid in USART3/USART4: 0b0: Disable wake-up signal detection 0b1: Enable wake-up signal detection  Note Set WKUPE to 1 only when the system requires wake-up signal detection. After detecting the wake-up signal, software must clear WKUPE.	RW
7:6	SBKL	LIN Interval Low Level Width Setting Bit	Only valid in USART3/USART4: 0b00: 10 bits 0b01: 11 bits 0b10: 13 bits 0b11: 14 bits	RW

Bit	Name	Function	Description	Access
7:6	SBKL	LIN Interval Low Level Width Setting Bit	 Note This bit must retain the reset value when UATSBKE=0 in non-UART-LIN Mode. This bit can only be set when TE=0 & RE=0 (transmit/receive disabled).	RW
5	LBDL	LIN Interval Low Level Width Detection Threshold Setting Bit	Only valid in USART3/USART4: 0b0: >= 10 bits 0b1: >= 11 bits  Note This bit must retain the reset value when UATSBKE=0 in non-UART-LIN Mode. This bit can only be set when TE=0 & RE=0 (transmit/receive disabled).	RW
4	LBDIE	LIN Interval Detection Interrupt Enable Bit	Only valid in USART3/USART4: 0b0: LIN interval segment detection interrupt request is disable, no interrupt occurs. 0b1: LIN interval segment detection interrupt request is enable, an interrupt occurred.  Note This bit must retain the reset value in non-UART-LIN Mode.	RW
3	BEE	LIN Bus Error Detection Function Enable Bit	Only valid in USART3/USART4: 0b0: Disable LIN bus error function detection function 0b1: Enable LIN bus error function detection function  Note This bit must retain the reset value when UATSBKE=0 in non-UART-LIN Mode. This bit can only be set when TE=0 & RE=0 (transmit/receive disabled).	RW
2	BEIE	LIN Bus Error Interrupt Enable Bit	Only valid in USART3/USART4: 0b0: LIN bus error interrupt request is disabled, no interrupt occurs. 0b1: LIN bus error interrupt request is enabled, an interrupt occurred.  Note This bit must retain the reset value in non-UART-LIN Mode.	RW

Bit	Name	Function	Description	Access
1	WKUPIE	LIN Wake-up Signal Detection Interrupt Enable Bit	Only valid in USART3/USART4: ● 0b0: LIN Wake-up signal detection interrupt request is disabled, no interrupt occurs. ● 0b1: LIN Wake-up signal detection interrupt request is enabled, an interrupt has occurred.  Note This bit must retain the reset value in non-UART-LIN Mode.	RW
0	MPE	Multiprocessor Function Enable Bit	In UART Mode, the multiprocessor function enable bit: ● 0b0: Disable ● 0b1: Enable  Note This bit must retain the reset value in non-UART Mode. This bit can only be set when TE=0 & RE=0 (transmit/receive disabled).	RW

24.4.8 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: 0b0: DE signal is high when USART is in IDLE state 0b1: DE signal is low when USART is in IDLE state 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. 0b0: DE function disable 0b1: DE function enable 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. 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. 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: 0b000: 32 0b001: 64 0b010: Setting prohibited 0b011: 128 0b100: Setting prohibited 0b101: 256	RW

Bit	Name	Function	Description	Access
23:21	BCN	Basic Clock Count	● 0b110: 372 ● 0b111: Setting prohibited  Note This bit must retain the reset value in non-Smart Card Mode. This bit can only be set when TE=0 & RE=0 (transmit/receive disabled). USART3/USART4 does not support this function. Read always returns 0.	RW
20:14	Reserved	Reserved	-	-
13:10	DED	DE Assertion Cancellation Time	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. 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. When USART TE or RE is enabled, writing to this bit field is prohibited.	RW
9	CTSE	CTS Function Enable Bit	● 0b0: CTS function disable ● 0b1: CTS function enable  Note This bit can only be set when TE=0 & RE=0 (transmit/receive disabled).	RW
8	RTSE	RTS Function Enable Bit	● 0b0: RTS function disable ● 0b1: RTS function enable  Note This bit can only be set when TE=0 & RE=0 (transmit/receive disabled).	RW
7:6	Reserved	Reserved	-	-

Bit	Name	Function	Description	Access
5	SCEN	Smart Card Mode Enable Bit	0b0: Disable Smart Card Mode 0b1: Enable Smart Card Mode  Note This bit must retain the reset value in non-Smart Card Mode. This bit can only be set when TE=0 & RE=0 (transmit/receive disabled). USART3/USART4 does not support this function. Read always returns 0.	RW
4	LOOP	LIN Loop Mode Enable Bit	Only valid in USART3/USART4: 0b0: Normal Mode 0b1: Loop Mode  Note This bit can only be set when TE=0 & RE=0 (transmit/receive disabled).	RW
3	HDSEL	UART Single-wire Half-duplex Mode Enable Bit	0b0: UART full-duplex mode 0b1: UART half-duplex mode  Note This bit can only be set when TE=0 & RE=0 (transmit/receive disabled).	RW
2:0	Reserved	Reserved	-	-

24.4.9 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																				RTOB				Reserved				ULBREN	LBMPSC		PSC	
Access																					RW								RW	RW		RW	

Bit	Name	Function	Description	Access
31:12	Reserved	Reserved	-	-
11:8	RTOB	UART TIMEOUT Length Setting Bits	Specifies the timeout length when the UART TIME-OUT function is enabled: (The unit of setting is the number of received bits) ● 0b0000-0b0100: 15 bits ● 0b0101: 31 bits ● 0b0110: 63 bits ● 0b0111: 127 bits ● 0b1000: 255 bits ● 0b1001: 511 bits ● 0b1010: 1023 bits ● 0b1011: 2047 bits ● 0b1100: 4095 bits ● 0b1101: 8191 bits ● 0b1110: 16383 bits ● 0b1111: 32767 bits  Note This bit can only be set when TE=0 & RE=0 (transmit/receive disabled).	RW
7:5	Reserved	Reserved	-	-
4	ULBREN	Ultra-low Baud Rate Enable Bit	When using the internal clock source, the ultra-low baud rate function enable bit: ● 0b0: Ultra-low baud rate function disable ● 0b1: Ultra-low baud rate function enable And the prescaler division value is determined by PSC as follows: ● 0b00: HCLK/128 ● 0b01: HCLK/256 ● 0b10: HCLK/512 ● 0b11: HCLK/1024  Note This bit can only be set when TE=0 & RE=0 (transmit/receive disabled).	RW

Bit	Name	Function	Description	Access
3:2	LBMPSC	LIN Baud Rate Measurement Counter Clock Source Selection	Only valid in USART3/USART4: ● 0b00: HCLK ● 0b01: HCLK/2 ● 0b10: HCLK/4 ● 0b11: HCLK/8  Note This bit can only be set when TE=0 & RE=0 (transmit/receive disabled).	RW
1:0	PSC	Prescaler Value	When using the internal clock source, the prescaler division ratio selection bit: ● 0b00: HCLK ● 0b01: HCLK/4 ● 0b10: HCLK/16 ● 0b11: HCLK/64  Note This bit can only be set when TE=0 & RE=0 (transmit/receive disabled).	RW

24.4.10 LIN Baudrate Measuring Counter 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	Only valid in USART3/USART4: LIN slave node uses the clock source selected by the USART_PR.LBMPSC register to measure the frequency of the synchronization segment transmitted by the master node. The measured LIN communication baud rate is obtained by dividing the count clock frequency selected by USART_PR.LBMPSC by the value of LBMC.  Note The value of LBMC is meaningful only after being read upon completion of the synchronization segment.	RO

24.5 Precautions

24.5.1 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.

24.5.2 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.

24.5.3 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.

25 Inter-Integrated Circuit (I2C)

25.1 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.

25.2 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

25.3 Functional Description

25.3.1 I2C System Block Diagram

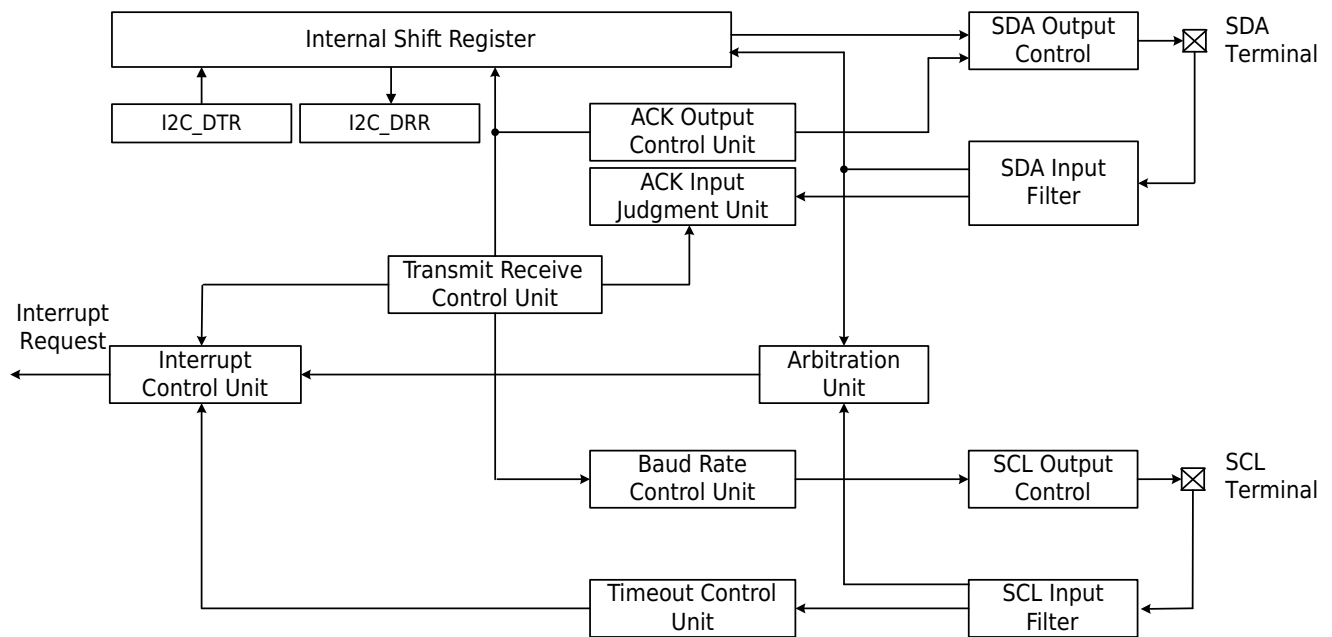


Figure 25-1 I2C System Block Diagram

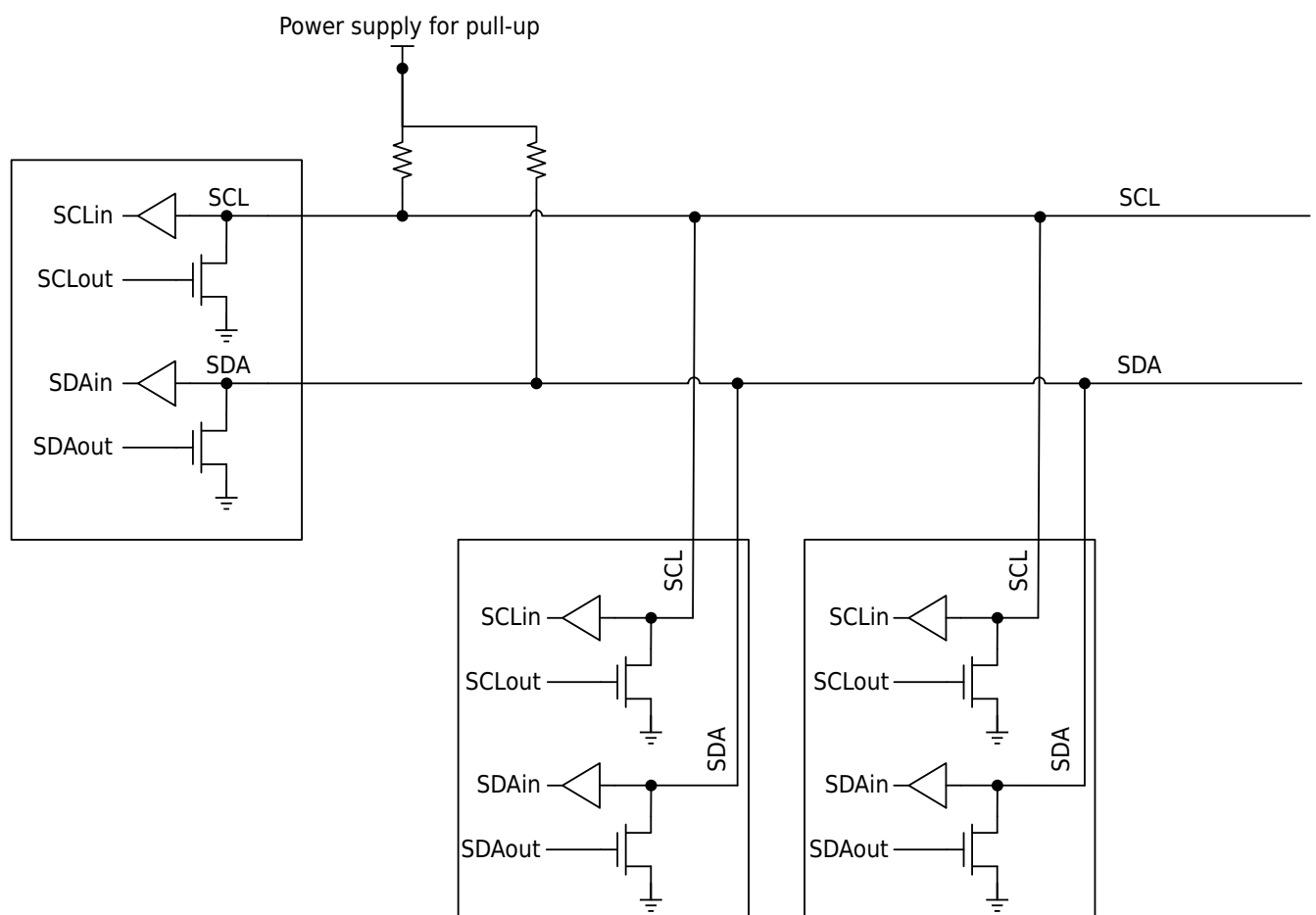


Figure 25-2 Example of I2C Bus Structure

Table 25-1 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, please set the SCL/SDA input level to the Schmitt level (compatible with the CMOS level). When the SMBus is selected, please set the SCL/SDA input level to the CMOS level (compatible with TTL level). For specific configuration details, please refer to the relevant description of the CINSEL bit in the General Control Register PCRxy in the "General-Purpose IO (GPIO) " chapter .

25.3.2 I2C Protocol

25.3.2.1 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

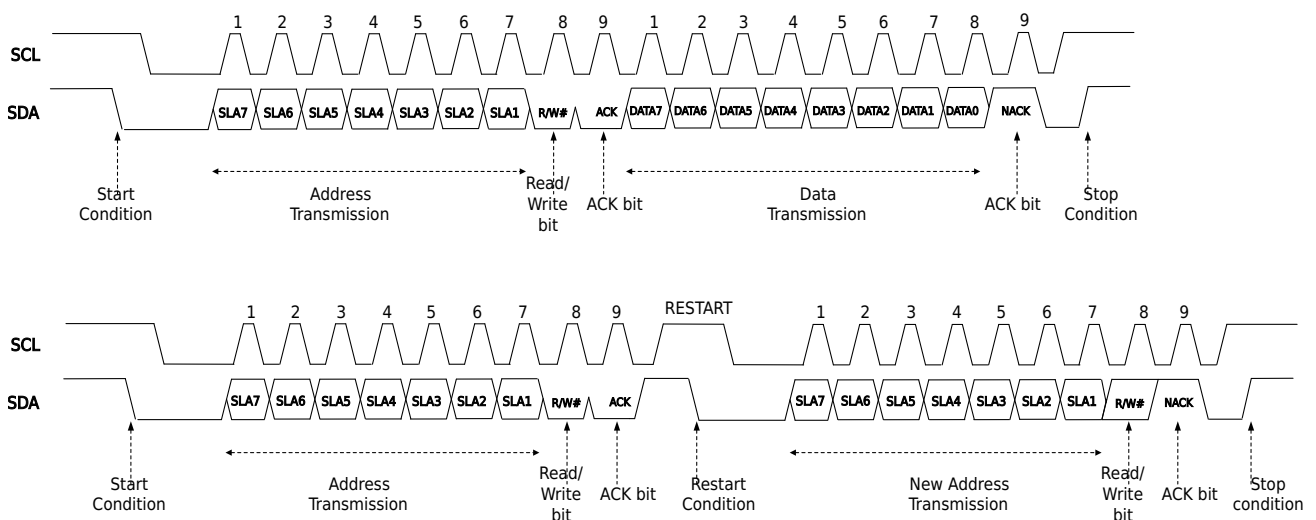


Figure 25-3 I2C Bus Timing Diagram

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/ W#	ACK/ NACK	DATA (8-bit)	ACK	DATA (8-bit)	ACK/ NACK	P

10-bit Address Format

S	0b11110+SL A (2-bit)	W#	ACK	SLA (8-bit)	ACK	DATA (8-bit)	ACK	DATA (8-bit)	ACK/ NACK	P

S	0b11110+SL A (2-bit)	W#	ACK	SLA (8-bit)	ACK	Sr	0b11110+SL A (2-bit)	R	ACK	DATA (8-bit)	ACK	DATA (8-bit)	ACK/ NACK	P

Figure 25-4 Data Format of I2C Bus

- S: Indicates the start condition
- SLA: Indicates the slave address
- R/W#: Indicates the direction of transmission and reception. Send data from slave to master when R/W# is 1; When R/W# 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.

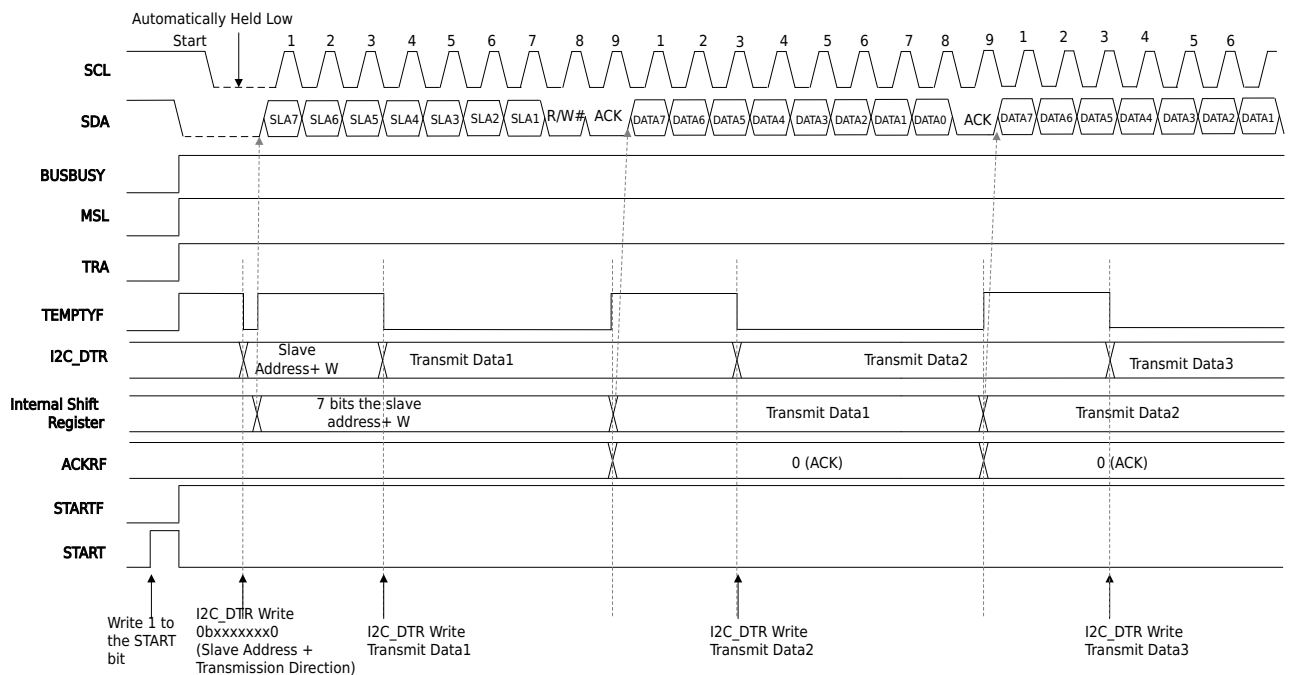


Figure 25-5 Sequence Diagram of Master Transmitting Data in 7-bit Address Format (Example)

- 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.

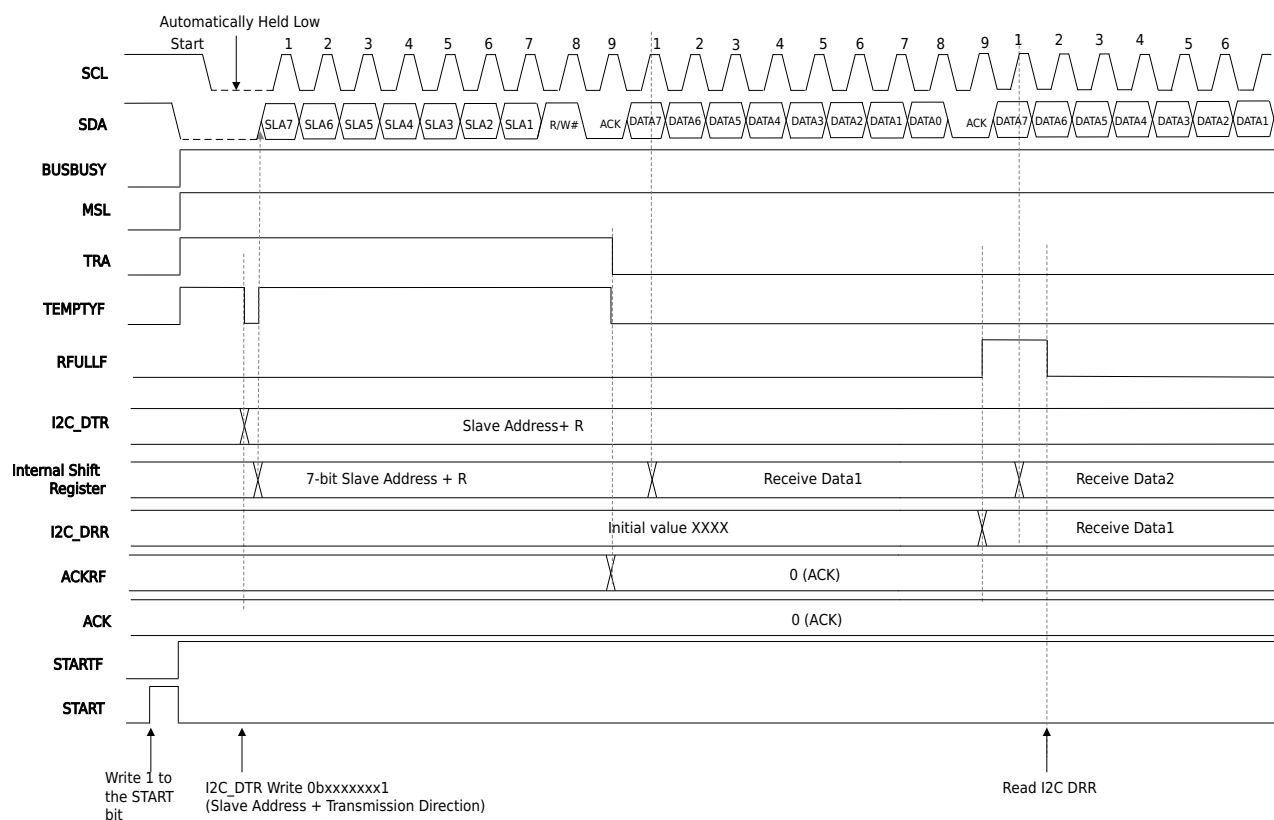


Figure 25-6 Sequence Diagram of Master Receiving Data in 7-bit Address Format (Example)

- 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.

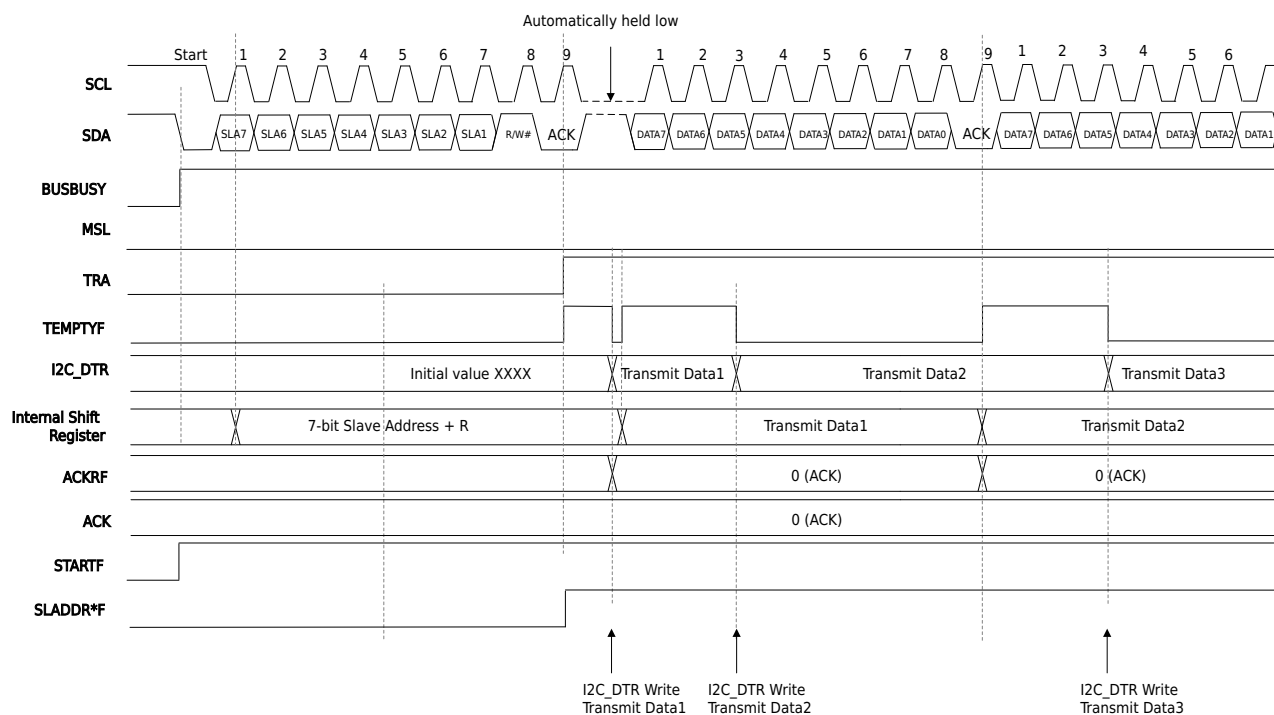


Figure 25-7 Sequence Diagram of Slave Transmitting Mode in 7-bit Address Format (Example)

● 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.

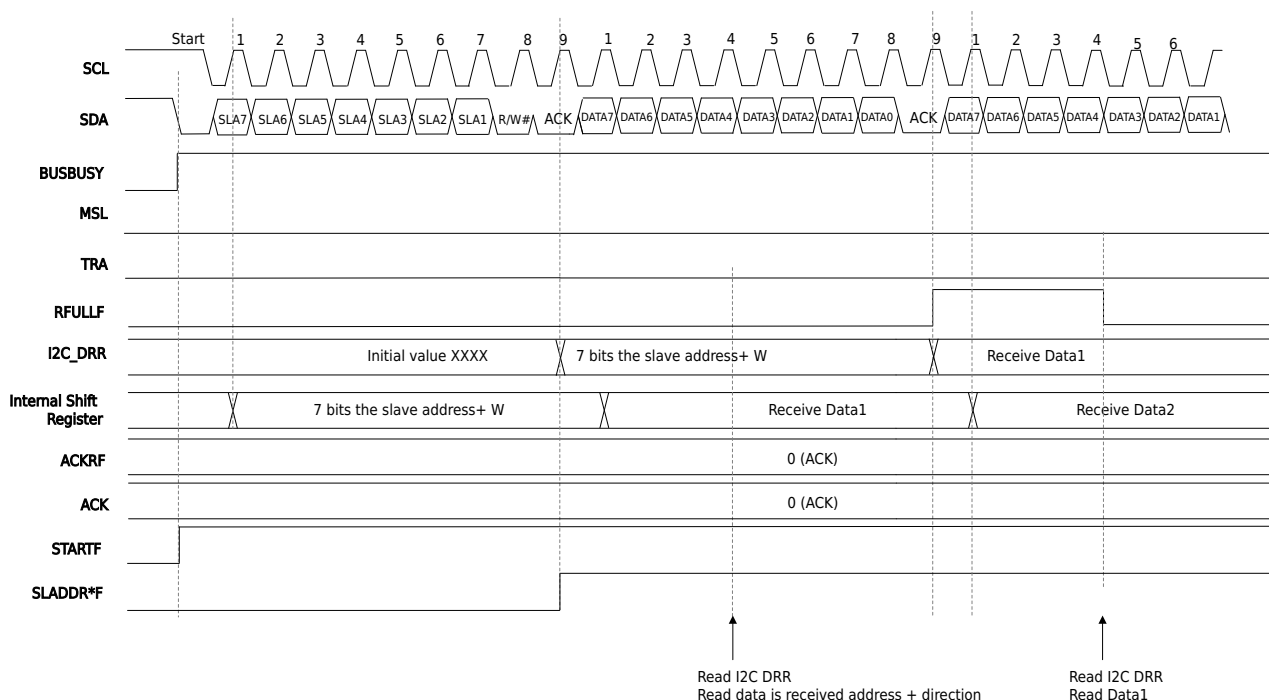


Figure 25-8 Sequence Diagram of Slave Receiving Data in 7-bit Address Format (Example)

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.

25.3.2.2 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.

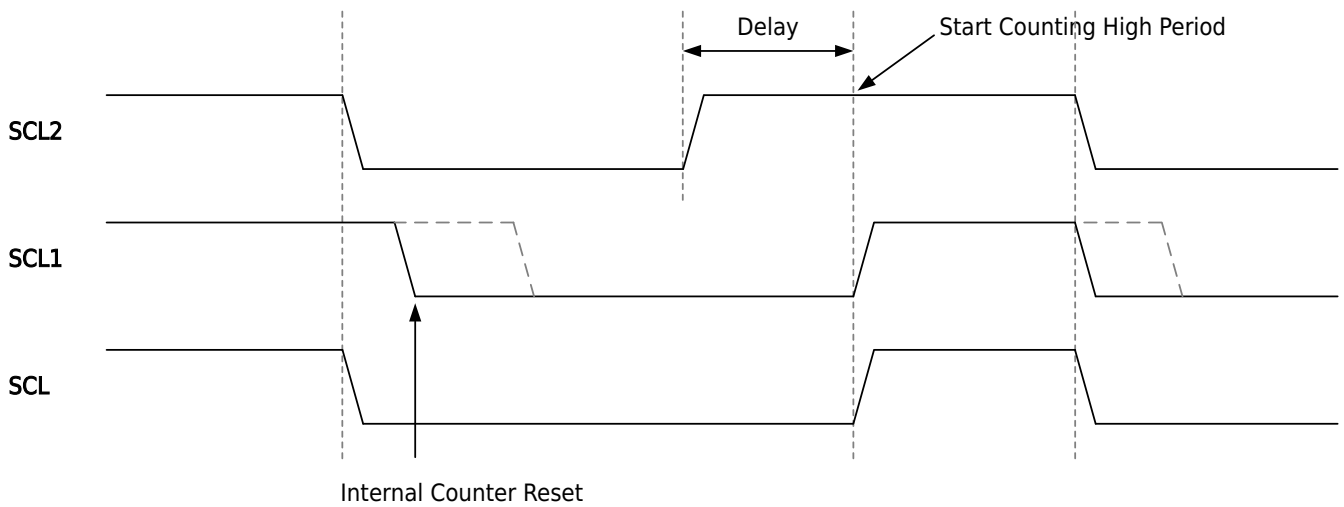


Figure 25-9 SCL Synchronous Timing

25.3.2.3 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.

25.3.2.4 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.

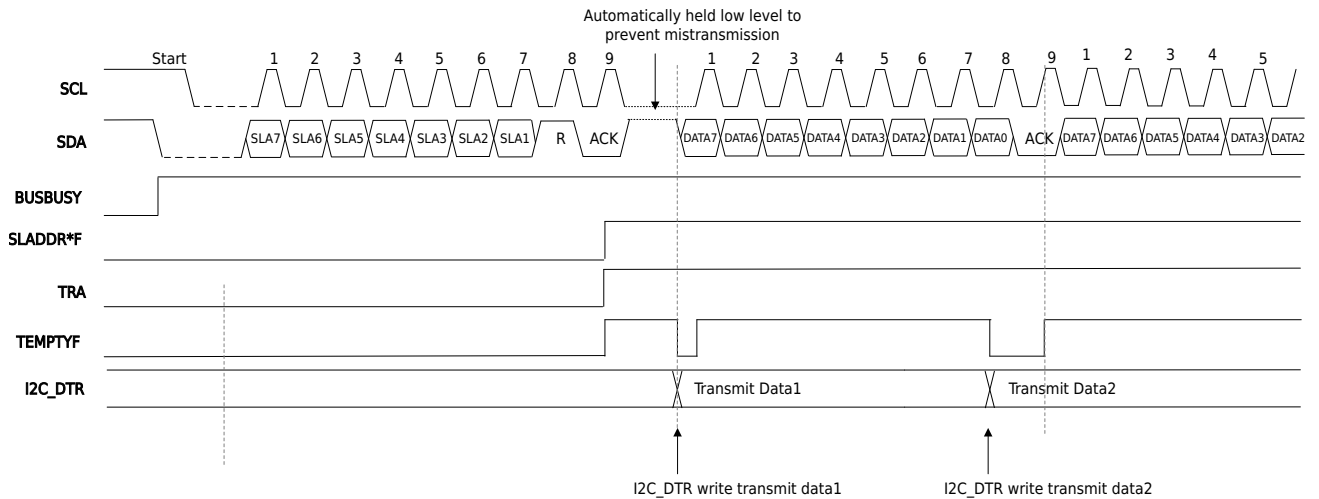


Figure 25-10 Transmitting Timing Diagram of Slave (1)

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.EMPTYF flag is 1. When the I2C_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.

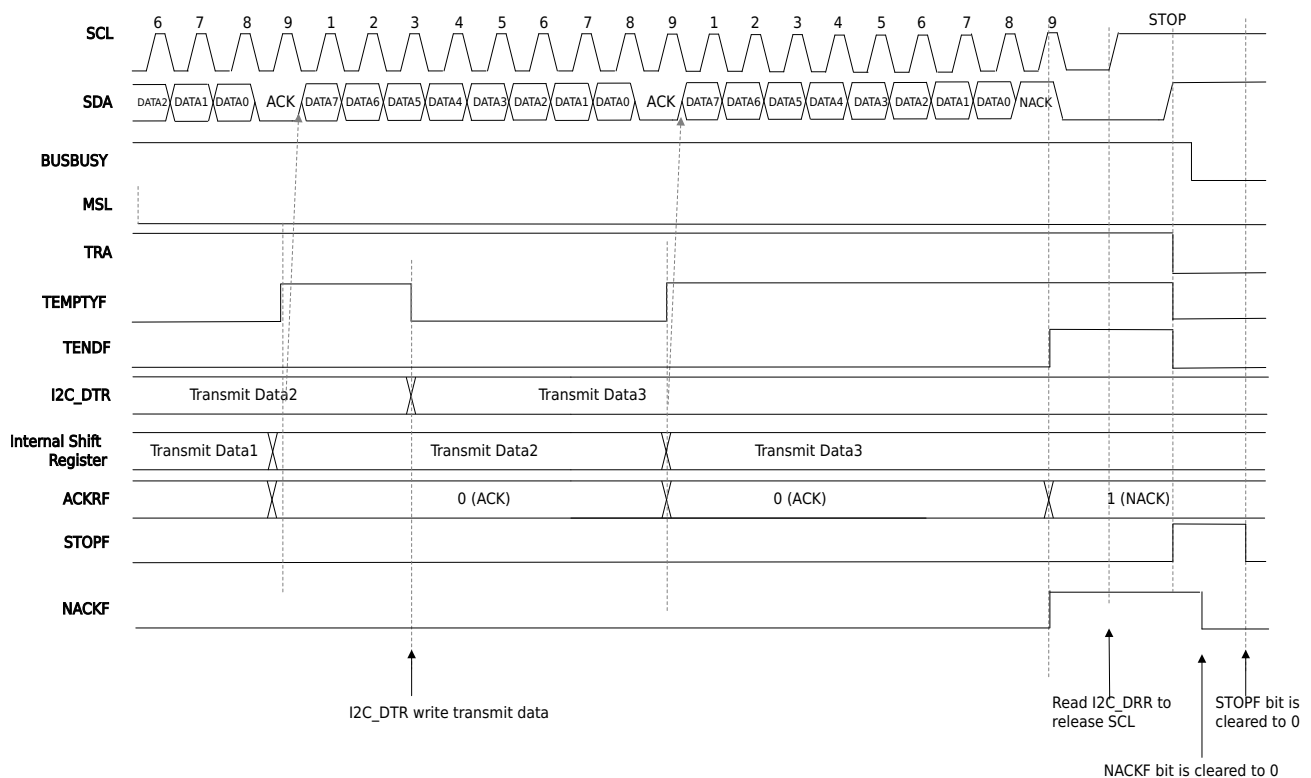


Figure 25-11 Transmitting Timing Diagram of Slave (2)

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.

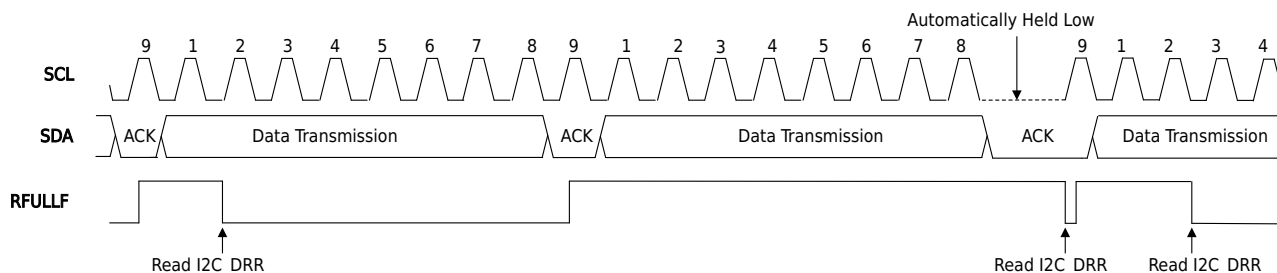


Figure 25-12 Slave Receiving Timing Diagram

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.

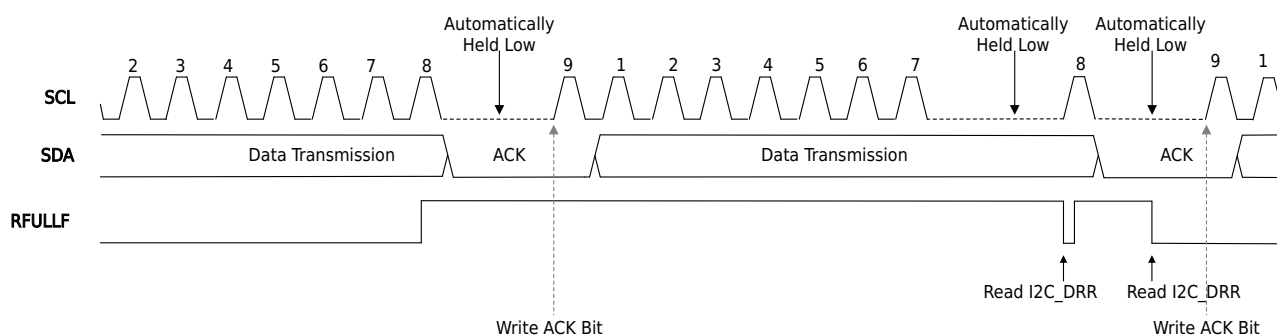


Figure 25-13 Fast ACK/NACK Timing Diagram

25.3.3 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.

25.3.3.1 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 R/W# 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.

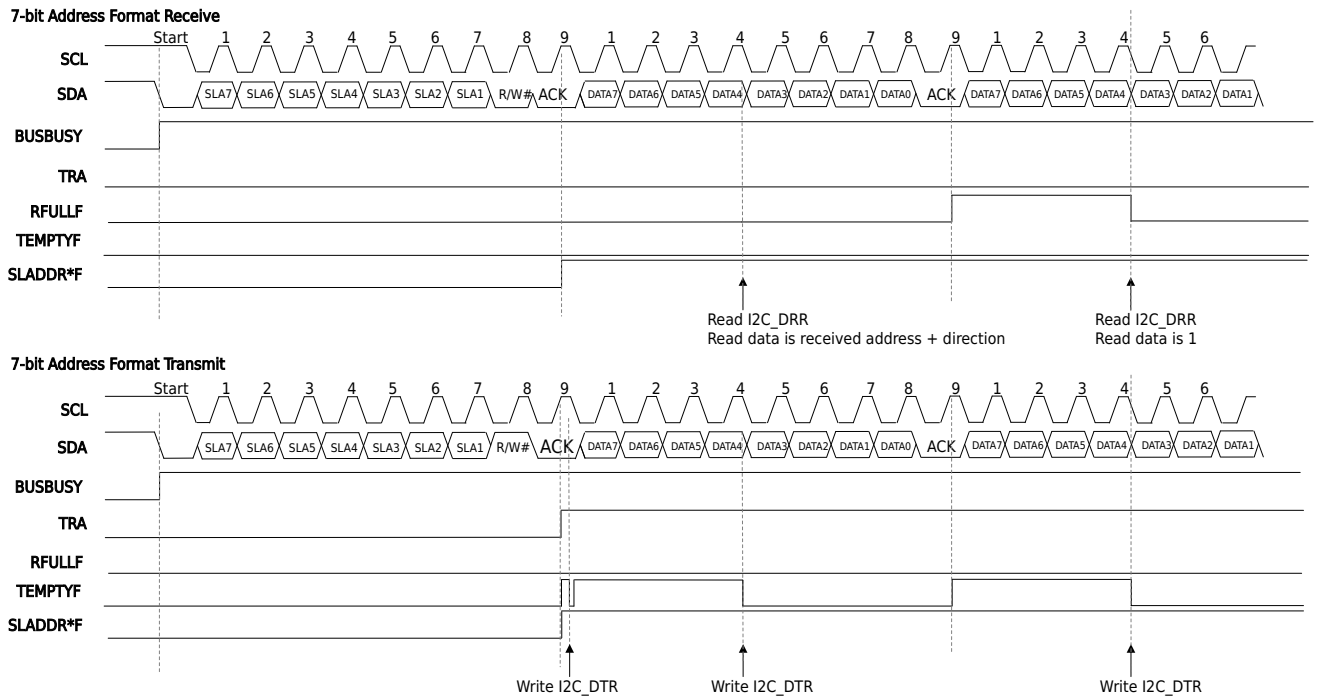


Figure 25-14 Timing when Selecting 7-bit Address Format

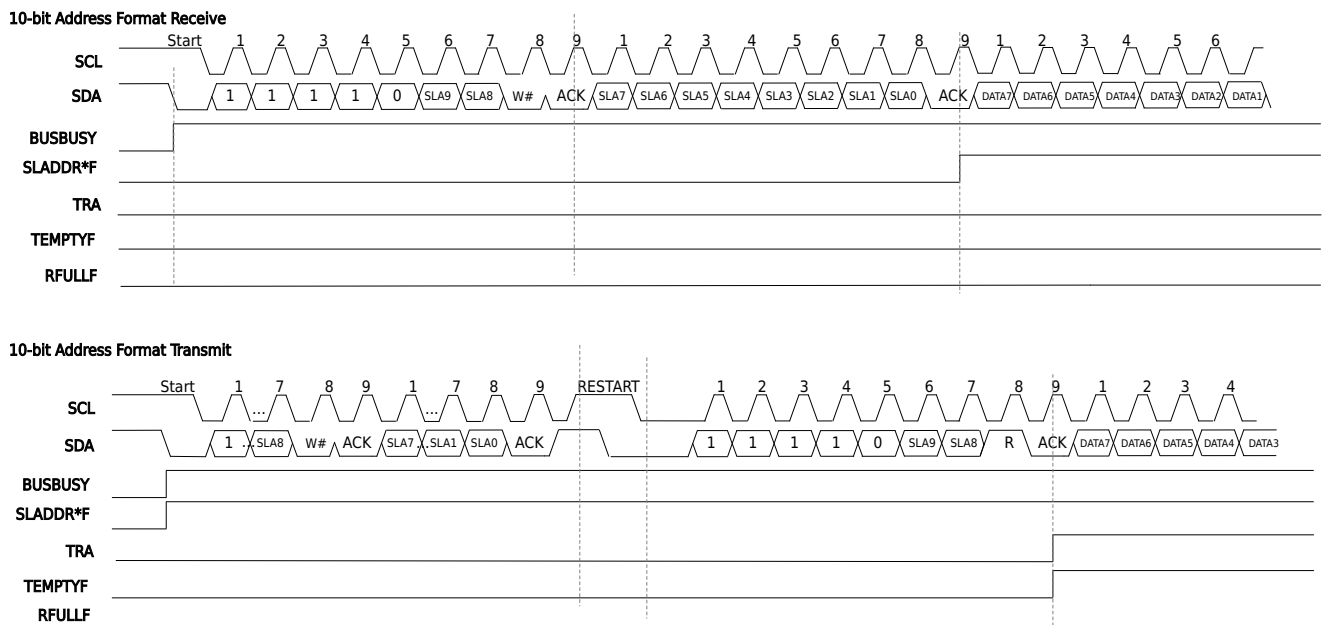


Figure 25-15 Timing when Selecting 10-bit Address Format

25.3.3.2 Broadcast Address Matching

When I2C_CR1.GCEN bit is 1, the broadcast address can be detected (0b0000000+0[W]).

But after the start condition or restart condition, the address is 0b0000000+1[R] (Start byte), which is considered the slave address of All "0" and not the broadcast address.

If the broadcast address is matched, the I2C_SR.GENCALLF flag is set to 1 at the rising edge of the 9th clock of SCL clock.

The operation after the broadcast address matches is the same as the ordinary slave receiving operation.

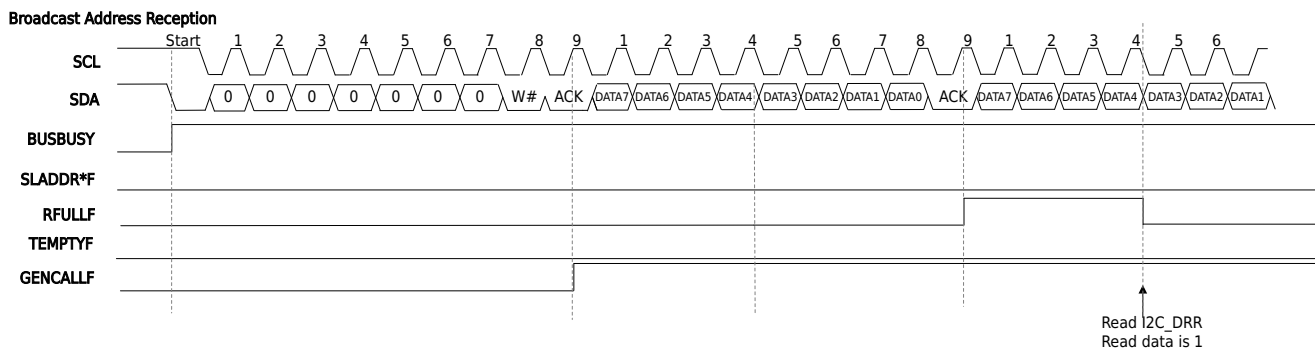


Figure 25-16 Broadcast Address Matching Timing Diagram

25.3.3.3 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=0 and I2C_SR.TRA=0).

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 (R/W# 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.

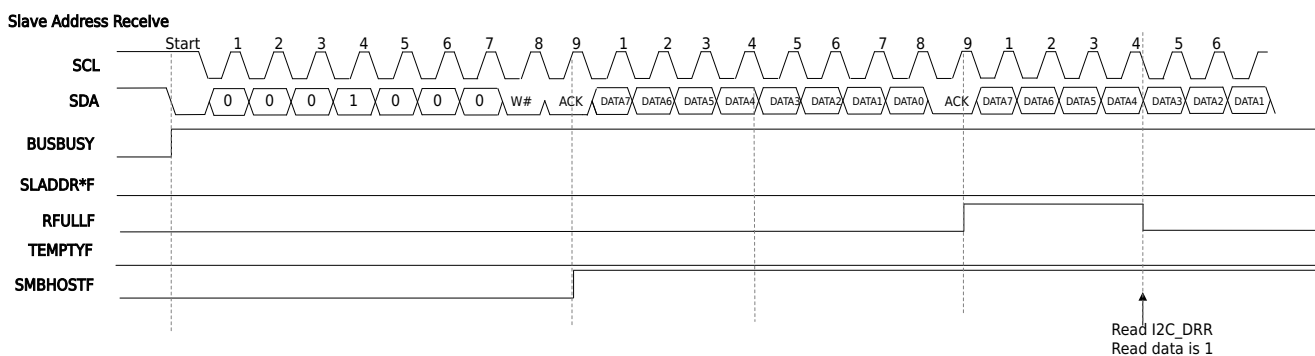


Figure 25-17 SMBus Master Address Matching Timing Diagram

25.3.3.4 SMBus Alarm Response Address Matching

This product has the function of alarm response address detection when SMBus is running. If the I2C_CR1.SMBALERTEN 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=0 and I2C_SR.TRA=0).

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.

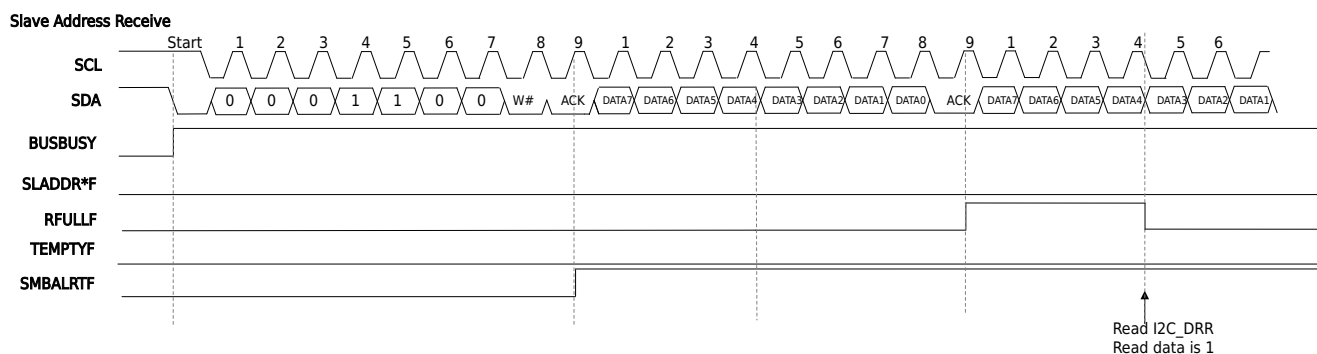


Figure 25-18 SMBus Alarm Response Address Matching Timing Diagram

25.3.3.5 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=0 and I2C_SR.TRA=0).

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.

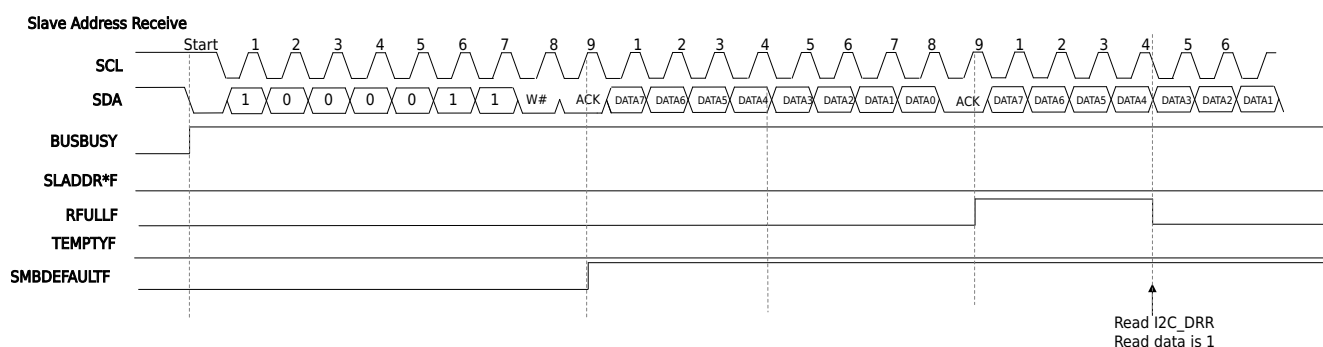


Figure 25-19 SMBus Default Address Timing Diagram

25.3.4 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.

25.3.4.1 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_{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_{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.

25.3.4.2 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.

25.3.5 Reset

It has the function of resetting the communication module. There are two types of reset, one is communication reset: all registers inside I2C 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 ($I2C\_CR1.PE=0$ and $I2C\_CR1.SWRST=1$), the bus status such as the start condition cannot be monitored.

25.3.6 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.

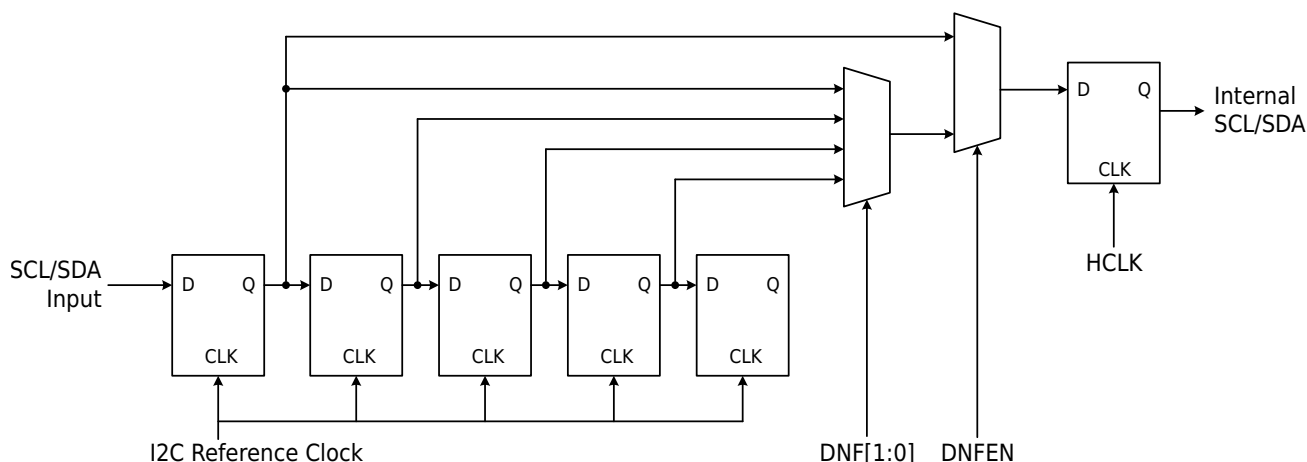


Figure 25-20 Digital Filtering Circuit Block Diagram

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.

25.3.7 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 25-2 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
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 25-3 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

25.4 Operation Example

25.4.1 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.

25.5 Register Description

25.5.1 Register List

Base address: 0x40004800

Table 25-4 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 Transmit Register	8
0x28	I2C_DRR	I2C Data Receive Register	8
0x2C	I2C_CCR	I2C Clock Control Register	32
0x30	I2C_FLTR	I2C Filter Control Register	32

25.5.2 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	● 0b0: Release reset ● 0b1: Software reset By combining with the PE bit, select internal status reset or communication reset: <table><tr><th>SWRST</th><th>PE</th><th>Reset Content</th></tr><tr><td>0b1</td><td>0b0</td><td>Communication reset: reset all registers 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>	SWRST	PE	Reset Content	0b1	0b0	Communication reset: reset all registers 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: reset all registers 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	● 0b0: The answer bit sends "0" (sends ACK) ● 0b1: The answer bit sends "1" (sends NACK)	RW									
9	STOP	Stop Condition Generation Bit	● 0b0: No stop condition is generated ● 0b1: Generate stop condition This bit can be set and cleared by software. Hardware clear condition: ● Stop condition detected ● Failure to arbitrate ● Start condition detected ● Communication reset	RW									

Bit	Name	Function	Description	Access
8	START	Start Condition Generating Bit	● 0b0: No start condition generated ● 0b1: Start condition generated This bit can be set and cleared by software. Hardware clear condition: ● Start condition detected ● Failure of arbitration ● Communication reset	RW
7	RESTART	Repeated Start Condition Generating Bit	● 0b0: No repeated start condition generated ● 0b1: Repeated start condition generated This bit can be set and cleared by software. Hardware clear condition: ● Start condition detected ● Failure of arbitration ● Communication reset	RW
6	GCEN	Broadcast Call Enabled	● 0b0: Broadcast address detection disabled ● 0b1: Broadcast address detection enabled	RW
5	<i>Reserved</i>	<i>Reserved</i>	-	-
4	SMBHOSTEN	Allow Matching of SMBUS Master Address Bit	● 0b0: SMBUS master address match disabled ● 0b1: SMBUS master address match enabled	RW
3	SMBDEFAULTEN	Allow Matching of SMBUS Default Address Bit	● 0b0: SMBUS default address match disabled ● 0b1: SMBUS default address match enabled	RW
2	SMBALRTEN	Allow Matching of SMBUS Alarm Response Address Bit	● 0b0: SMBUS alarm response address disabled ● 0b1: SMBUS alarm response address enabled	RW
1	SMBUS	SMBUS/I2C Bus Mode Selection Bit	● 0b0: I2C bus mode ● 0b1: SMBUS mode	RW
0	PE	I2C Function Enabled	By combining with the SWRST bit, select internal status reset or communication reset: ● 0b0: I2C function disabled ● 0b1: I2C function enabled	RW

25.5.3 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	SMBHOSTIE	SMBDEFAULTIE	GENCALLIE	Reserved								TMOUTIE	Reserved	NACKIE	Reserved	ARLOIE	Reserved	EMPTYIE	RFULLIE	Reserved	STOPIE	TENDIE	SLADDR1IE	SLADDR0IE	STARTIE				
Access									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	0b0: Interrupt disabled 0b1: Interrupt enabled	RW
22	SMBHOSTIE	SMBUS Master Address Matching Consistent Interrupt Enabled	0b0: Interrupt disabled 0b1: Interrupt enabled	RW
21	SMBDEFAULTIE	SMBUS Default Address Matching Consistent Interrupt Enabled	0b0: Interrupt disabled 0b1: Interrupt enabled	RW
20	GENCALLIE	Broadcast Call Address Matching Consistent Interrupt Enabled	0b0: Interrupt disabled 0b1: Interrupt enabled	RW
19:15	Reserved	Reserved	-	-
14	TMOUTIE	Timeout Interrupt Enabled	0b0: Interrupt disabled 0b1: Interrupt enabled	RW
13	Reserved	Reserved	-	-
12	NACKIE	NACK Interrupt Enabled	0b0: Interrupt disabled 0b1: Interrupt enabled	RW
11:10	Reserved	Reserved	-	-
9	ARLOIE	Arbitration Failure Interrupt Enabled	0b0: Interrupt disabled 0b1: Interrupt enabled	RW
8	Reserved	Reserved	-	-

Bit	Name	Function	Description	Access
7	EMPTYIE	Transmit Data Empty Interrupt Enabled	● 0b0: Interrupt disabled ● 0b1: Interrupt enabled	RW
6	RFULLIE	Receive Data Full Interrupt Enabled	● 0b0: Interrupt disabled ● 0b1: Interrupt enabled	RW
5	<i>Reserved</i>	<i>Reserved</i>	-	-
4	STOPIE	Stop Condition Interrupt Enabled	● 0b0: Interrupt disabled ● 0b1: Interrupt enabled	RW
3	TENDIE	Send a Frame of Data End Interrupt Enabled	● 0b0: Interrupt disabled ● 0b1: Interrupt enabled	RW
2	SLADDR1IE	Slave Address 1 Match Consistent Interrupt Enabled	● 0b0: Interrupt disabled ● 0b1: Interrupt enabled	RW
1	SLADDR0IE	Slave Address 0 Match Consistent Interrupt Enabled	● 0b0: Interrupt disabled ● 0b1: Interrupt enabled	RW
0	STARTIE	Start Condition/Repeated Start Condition Interrupt Enabled	● 0b0: Interrupt disabled ● 0b1: Interrupt enabled	RW

25.5.4 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	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) 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)  Note Remove the held low level by writing the ACK bit.	RW
6:3	Reserved	Reserved	-	-
2	HTMOUT	High Level Timeout Detection Enabled	0b0: Timeout detection disabled when SCL is high level 0b1: Timeout detection enabled when SCL is high level	RW
1	LTMOUT	Low Level Timeout Detection Enabled	0b0: Timeout detection disabled when SCL is low level 0b1: Timeout detection enabled when SCL is low level	RW
0	TMOUTEN	Timeout Function Enable Bit	0b0: SCL level timeout detection function disabled 0b1: SCL level timeout detection function enabled	RW

25.5.5 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												SDADLY		Reserved		BUSFREE_CLREN	Reserved	BUSWAIT	Reserved												
Access													RW				RW		RW													

Bit	Name	Function	Description	Access
31:22	Reserved	Reserved	-	-
21:20	Reserved	Reserved	All bits read as 1, Write as 1	-
19:16	SDADLY	SDA Output Delay Counter	0b0000: No output delay 0b0001: 1 I2C reference clock period 0b0010: 2 I2C reference clock periods 0b1111: 15 I2C reference clock periods	RW
15:13	Reserved	Reserved	-	-
12	BUSFREE_CLREN	Bus End Status Clear Bit	0b0: Disable the I2C bus STOP flag to clear the SMBALRTF, SMBHOSTF, SMBDEFAULTF, GENCALLF, MSL, SLADDR1F, SLADDR0F, and STARTF bits in the I2C_SR status register. 0b1: Enable the I2C bus STOP flag to clear the SMBALRTF, SMBHOSTF, SMBDEFAULTF, GENCALLF, MSL, TENDF, SLADDR1F, SLADDR0F, and STARTF bits in the I2C_SR status register.	RW
11	Reserved	Reserved	Reads as 1, Write as 1	-
10	BUSWAIT	Bus Wait Bit	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 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.	RW
9:8	Reserved	Reserved	All bits read as 1, Write as 1	-

Bit	Name	Function	Description	Access
7:3	<i>Reserved</i>	<i>Reserved</i>	-	-
2:0	<i>Reserved</i>	<i>Reserved</i>	<i>All bits read as 1, Write as 1</i>	-

25.5.6 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																ADDRMOD0	Reserved		SLADDR0EN	Reserved		SLADDR0									
Access																	RW			RW			RW									

Bit	Name	Function	Description	Access
31:16	Reserved	Reserved	-	-
15	ADDRMOD0	7-bit/10-bit Address Format Selection Bit	0b0: Select 7-bit address format 0b1: Select 10-bit address format	RW
14:13	Reserved	Reserved	-	-
12	SLADDR0EN	Slave Address 0 Enabled Bit	0b0: Slave address register 0 setting value disabled 0b1: Slave address register 0 setting value enabled	RW
11:10	Reserved	Reserved	-	-
9:8	SLADDR0[9:8]	High Bit of a 10-bit Slave Address.	Set the slave address: - When ADDRMOD0=0, the SLADDR0[9:8] setting is invalid; - When ADDRMOD0=1, SLADDR0[9:8] is the upper two bits of the 10-bit slave address.	RW
7:0	SLADDR0 [7:0]	Lower Bits of 7-bit Address/10-bit Address	Set the slave address: - When ADDRMOD0=0, SLADDR0[7:1] is the 7-bit slave address, and SLADDR0[0] is invalid; - When ADDRMOD0=1, SLADDR0 [7:0] is the lower 8 bits of the 10-bit slave address.	

25.5.7 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																ADDRMOD1	Reserved				SLADDR1EN	Reserved				SLADDR1							
Access																	RW					RW					RW							

Bit	Name	Function	Description	Access
31:16	Reserved	Reserved	-	-
15	ADDRMOD1	7-bit/10-bit Address Format Selection Bit	0b0: Select 7-bit address format 0b1: Select 10-bit address format	RW
14:13	Reserved	Reserved	-	-
12	SLADDR1EN	Slave Address 1 Enable Bit	0b0: Slave address register 1 setting value disabled 0b1: Slave address register 1 setting value enabled	RW
11:10	Reserved	Reserved	-	-
9:8	SLADDR1[9:8]	High Bit of a 10-bit Slave Address.	Set the slave address: - When ADDRMOD1=0, the SLADDR1[9:8] setting is invalid; - When ADDRMOD1=1, the SLADDR1[9:8] is the upper two bits of the 10-bit slave address.	RW
7:0	SLADDR1[7:0]	Low Bits of 7-bit Address/10-bit Address	Set the slave address: - When ADDRMOD1=0, SLADDR1[7:1] is the 7-bit slave address, and SLADDR1[0] is invalid; - When ADDRMOD1=1, SLADDR1[7:0] is the low 8 bits of the 10-bit slave address.	

25.5.8 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

25.5.9 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	0x00000000															
Name	Reserved								SMBALRTF	SMBHOSTF	SMBDEFAULTF	GENCALLF	Reserved	TRA	BUSY	MSL
Access									RO	RO	RO	RO		RW	RO	RW
													Reserved	TMOUTF	Reserved	NACKF
													Reserved	ACKRF	ARLOF	Reserved
													Reserved	TEMTYF	RFULLF	Reserved
													Reserved	STOPF	TENDF	SLADDR1F
																SLADDR0F
																STARTF

Bit	Name	Function	Description	Access
31:24	Reserved	Reserved	-	-
23	SMBALRTF	SMBUS Alarm Response Address Matching Consistent Flag Bit	0b0: SMBUS alarm response address not matched 0b1: Master address detected Setting conditions: - Received address matches 0b0001100 Clearing Conditions: - SMBALRTFCLR writes "1" - Stop condition detected - Communication reset - Internal state reset	RO
22	SMBHOSTF	SMBUS Master Address Matching Consistent Flag Bit	0b0: No match to SMBUS master address 0b1: Match to SMBUS master address Setting conditions: - Received address matches 0b0001000 Clearing Conditions: - SMBHOSTFCLR writes "1" - Stop condition detected - Communication reset - Internal state reset	RO
21	SMBDEFAULTF	SMBUS Default Address Matching Consistent Flag Bit	0b0: No match with the SMBUS default address. 0b1: A match with the SMBUS default address. Setting conditions: - Received address matches 0b1100001 Clearing Conditions: - SMBDEFAULTFCLR Write "1" - Stop condition detected - Communication reset - Internal state reset	RO
20	GENCALLF	Broadcast Call Address Match-	0b0: No match with the broadcast call address.	RO

Bit	Name	Function	Description	Access															
20	GENCALLF	ing Consistent Flag Bit	0b1: A match with the broadcast call address. Setting conditions: - The received slave address matches the broadcast call address (All "0") Clearing Conditions: - GENCALLFCLR write "1" - Stop condition detected - Communication reset - Internal state reset	RO															
19	Reserved	Reserved	-	-															
18	TRA	Transmit Receive Selection Bit	0b0: Receive data 0b1: Transmit data The operating mode of I2C is represented by a combination of MSL bits: <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> This bit can be set and cleared by software. Hardware set conditions: - Start condition detected - In master mode, the transmitted RW bit is 0 - In slave mode, the address matches and the received RW bit is 1 Hardware clear condition - Stop condition detected - In master mode, the transmitted RW bit is 1 - In slave mode, the address matches and the received RW bit is 0 - Communication reset - Internal state reset	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	0b0: Idle state, no communication on the bus 0b1: Occupied state, the bus is communicating Setting conditions: - Start condition detected on bus Clearing Conditions: - Bus stop condition detected - Communication reset	RO															

Bit	Name	Function	Description	Access
16	MSL	Master-slave Selection Bit	● 0b0: Slave mode ● 0b1: Master mode The operating mode of I2C is represented by a combination of TRA bits. This bit can be set and cleared by software. Hardware set conditions: ● Start condition detected when START bit is 1 Hardware clear condition ● Stop condition detected ● Failure to arbitrate ● Communication reset ● Internal state reset	RW
15	<i>Reserved</i>	<i>Reserved</i>	-	-
14	TMOUTF	Timeout Flag Bit	● 0b0: SCL level timeout not detected ● 0b1: SCL level timeout Setting conditions: ● During the period set by I2C_SLTR, SCL is not inverted Clearing conditions: ● TMOUTFCLR writes "1" ● Communication reset ● Internal state reset	RO
13	<i>Reserved</i>	<i>Reserved</i>	-	-
12	NACKF	NACK Flag Bit	● 0b0: NACK not received ● 0b1: NACK received Setting conditions: ● Receive NACK in transmit mode Clearing Conditions: ● NACKFCLR writes "1" ● Communication reset ● Internal state reset	RO
11	<i>Reserved</i>	<i>Reserved</i>	-	-
10	ACKRF	Receive Acknowledge Bit	● 0b0: The received acknowledge bit is "0" (receive ACK) ● 0b1: The received acknowledge bit is "1" (receive NACK) Setting conditions: ● Receive NACK in transmit mode Clearing conditions: ● Receive ACK in transmit mode ● Communication reset ● Internal state reset	RO

Bit	Name	Function	Description	Access
9	ARLOF	Arbitration Failure Flag Bit	● 0b0: No arbitration failure occurred ● 0b1: Arbitration failure Setting conditions: ● Failure to arbitrate Clearing conditions: ● ARLOFCLR writes "1" ● Communication reset ● Internal state reset	RO
8	<i>Reserved</i>	<i>Reserved</i>	-	-
7	EMPTYF	Transmit Data Empty Flag Bit	● 0b0: I2C_DTR register full ● 0b1: I2C_DTR register empty Setting conditions: ● I2C_DTR data is transferred to the internal shift register ● TRA bit set to 1 Clearing Conditions: ● Write I2C_DTR ● TRA bit clear 0 ● Communication reset ● Internal state reset	RO
6	RFULLF	Receive Data Full Flag Bit	● 0b0: I2C_DRR register empty ● 0b1: I2C_DRR register full Setting conditions: ● The received data is transferred from the internal shift register to I2C_DRR Clearing conditions: ● Read I2C_DRR ● RFULLFCLR writes "1" ● Communication reset ● Internal state reset	RO
5	<i>Reserved</i>	<i>Reserved</i>	-	-
4	STOPF	Stop Condition Flag Bit	● 0b0: No STOP condition has been detected on the bus. ● 0b1: A STOP condition has been detected on the bus. Setting conditions: ● Stop condition detected Clearing conditions: ● STOPFCLR writes "1" ● Communication reset ● Internal state reset	RO

Bit	Name	Function	Description	Access
3	TENDF	Transmit Data End Flag Bit	0b0: The internal shift register is still transmitting data. 0b1: The internal shift register has finished transmitting data. Setting conditions: - Under the condition of TEMPTYF=1, the 9th rising edge of SCL is "1" Clearing conditions: - Stop condition detected - Write I2C_DTR - TENDFCLR writes "1" - Communication reset - Internal state reset	RO
2	SLADDR1F	Slave Address Register 1 Matching Consistent Mark Bit	0b0: No consistent address was detected in the slave address register 1. 0b1: A consistent address is detected in the slave address register 1. Setting conditions: - When I2C_SLR1.ADDRM0D1=0, the received slave address is matched with the I2C_SLR1.SLADDR1[7:1]. - 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] Clearing conditions: - Stop condition detected - SLADDR1FCLR writes "1" - Communication reset - Internal state reset	RO
1	SLADDR0F	Slave Address Register 0 Matching Consistent Mark Bit	0b0: No consistent address was detected in the slave address register 0. 0b1: A consistent address is detected in the slave address register 0. Setting conditions: - When I2C_SLR0.ADDRM0D0=0, the received slave address is compared with the I2C_SLR0.SLADDR0[7:1]. - 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] Clearing conditions: - Stop condition detected - SLADDR0FCLR writes "1" - Communication reset - Internal state reset	RO

Bit	Name	Function	Description	Access
0	STARTF	Start Condition/ Restart Condi- tion Flag Bit	● 0b0: Bus not detected starting condition ● 0b0: Bus detected starting condition Setting conditions: ● Start condition detected Clearing conditions: ● Stop condition detected ● STARTFCLR writes "1" ● Communication reset ● Internal state reset	RO

25.5.10 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		ARLOFCLR		Reserved		RFULLFCLR		Reserved		STOPFCLR		TENDFCLR		SLADDR1FCLR		SLADDR0FCLR		STARTFCLR	
Access									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
11:10	Reserved	Reserved	-	-
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	-	-
4	STOPFCLR	Stop Condition Flag Clear	Write 1 to clear the STOPF flag bit, write 0 has no effect.	WO

Bit	Name	Function	Description	Access
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/Restart Condition Flag Clear	Write 1 to clear the STARTF flag bit, write 0 has no effect.	WO

25.5.11 I2C Data Transmit 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.

25.5.12 I2C Data Receive 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.

25.5.13 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	Reserved				SHIGHW				Reserved				SLOWW			
Access																	RW					RW								RW			

Bit	Name	Function	Description	Access
31:19	Reserved	Reserved	-	-
18:16	CKDIV	Setting Bit of I2C Reference Clock Frequency	0b000: I2C Reference Clock Frequency=HCLK 0b001: I2C Reference Clock Frequency=HCLK/2 0b010: I2C Reference Clock Frequency=HCLK/4 0b011: I2C Reference Clock Frequency=HCLK/8 0b100: I2C Reference Clock Frequency=HCLK/16 0b101: I2C Reference Clock Frequency=HCLK/32 0b110: I2C Reference Clock Frequency=HCLK/64 0b111: I2C Reference Clock Frequency=HCLK/128	RW
15:13	Reserved	Reserved	All bits read as 1, Write as 1	-
12:8	SHIGHW	Set SCL High Level Width Bit	Setting the high level width of SCL clock	RW
7:5	Reserved	Reserved	All bits read as 1, Write as 1	-
4:0	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 N	I2C_CCR.CKDIV	Baud rate
0b0	0b000	$B = \frac{1}{\frac{(SHIGHW + 3) + (SLOWW + 3)}{\Phi_{I2C}} + t_R + t_F}$
0b1	0b000	$B = \frac{1}{\frac{(SHIGHW + 3 + DNFE) + (SLOWW + 3 + DNFE)}{\Phi_{I2C}} + t_R + t_F}$
0b0	Non-0	$B = \frac{1}{\frac{(SHIGHW + 2) + (SLOWW + 2)}{\Phi_{I2C}} + t_R + t_F}$
0b1	Non-0	$B = \frac{1}{\frac{(SHIGHW + 2 + DNFE) + (SLOWW + 2 + DNFE)}{\Phi_{I2C}} + t_R + t_F}$

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.

25.5.14 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	● 0b0: Analog filter function disabled ● 0b1: Analog filter function enabled	RW
4	DNFEN	Digital Filtering Function Allowed Bit	● 0b0: Digital filter function disabled ● 0b1: Digital filter function enabled	RW
3:2	Reserved	Reserved	-	-
1:0	DNF	Selection of Filtering Capabilities for Digital Filters	● 0b00: Filtering capability 1 I2C reference clock period ● 0b01: Filtering capability 2 I2C reference clock periods ● 0b10: Filtering capability 3 I2C reference clock periods ● 0b11: Filtering capability 4 I2C reference clock periods	RW

26 Serial Peripheral Interface (SPI)

26.1 Introduction

This product is equipped with 2-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 required.

The SPI1 supports the function of waking up from STOP mode via the SS0 line.

26.2 Features

Table 26-1 SPI Main Features

Essentials	Description
Serial Communication Function	● Supporting 4-wire SPI mode and 3-wire clock synchronization mode ● Supports full-duplex synchronous communication mode and transmit-only communication mode ● Polarity and phase of adjustable communication clock SCK
Data Format	● Selectable data shift order: MSB start/LSB start ● Selectable data width: 8/16 bits ● A maximum of 1 frame width of 16-bit data can be transmitted or received at a single time
Baud Rate	● 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 HCLK. ● Maximum baud rate allowed in slave mode is 6 frequency division of HCLK
Error Monitoring	● Mode fault error monitoring ● Data overload error monitoring ● Data underload error monitoring ● Parity error monitoring
Chip Selection Signal Control	● Each channel is configured with 4 chip-selected signal line ● Polarity adjustable
Transmission Control in Master Mode	● Start transmission by writing data to the data register
Interrupt	● Receive buffer is full ● Transmit buffer is empty ● SPI error (mode/overload/underload/parity) ● SPI vacant ● Transmit complete (event source only)
Low Power Control	● Configurable module stop

Essentials	Description
Other Functions	● SPI initialization function ● Wake-up from STOP mode

26.3 Functional Description

26.3.1 System Block Diagram

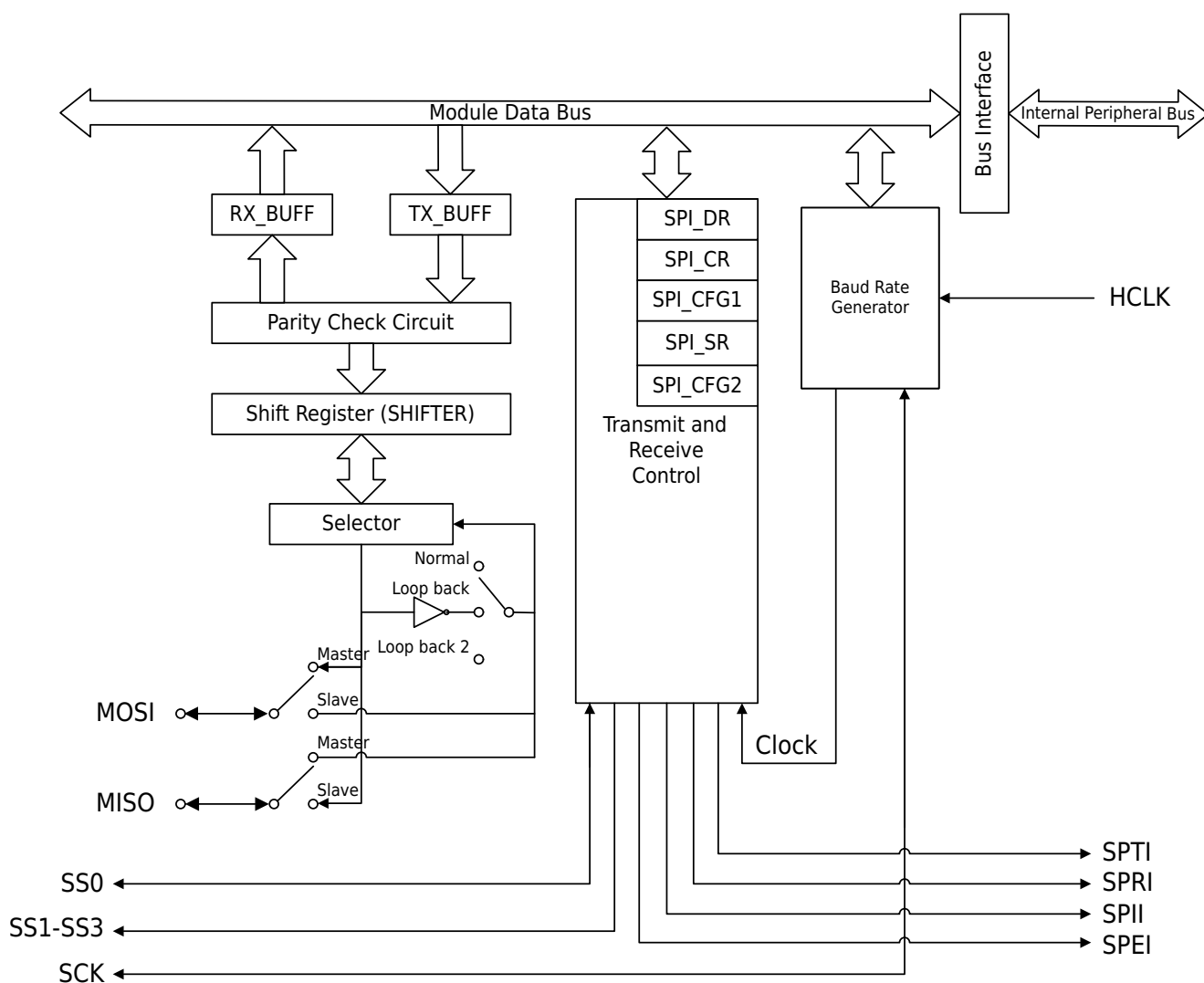


Figure 26-1 SPI System Block Diagram

Table 26-2 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

Pin Name	Port direction	Function
SS0	Input/Output	Slave select input/output pin
SS1	Output	Slave select output pin
SS2	Output	Slave select output pin
SS3	Output	Slave select output pin

26.3.2 SPI Action System Description

26.3.2.1 Pin Status in Master Mode

When SPI works in master mode, the status of each pin is shown below.

Table 26-3 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 "Register-General Control Register (PCRxy) " in the "General-Purpose IO (GPIO) " chapter.

26.3.2.2 Pin Status in Slave Mode

When SPI works in slave mode, the status of each pin is shown below.

Table 26-4 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

Mode		Pin Name	Pin Status
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 "Register-General Control Register (PCRxy)" in the "General-Purpose IO (GPIO)" chapter.

26.3.2.3 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.

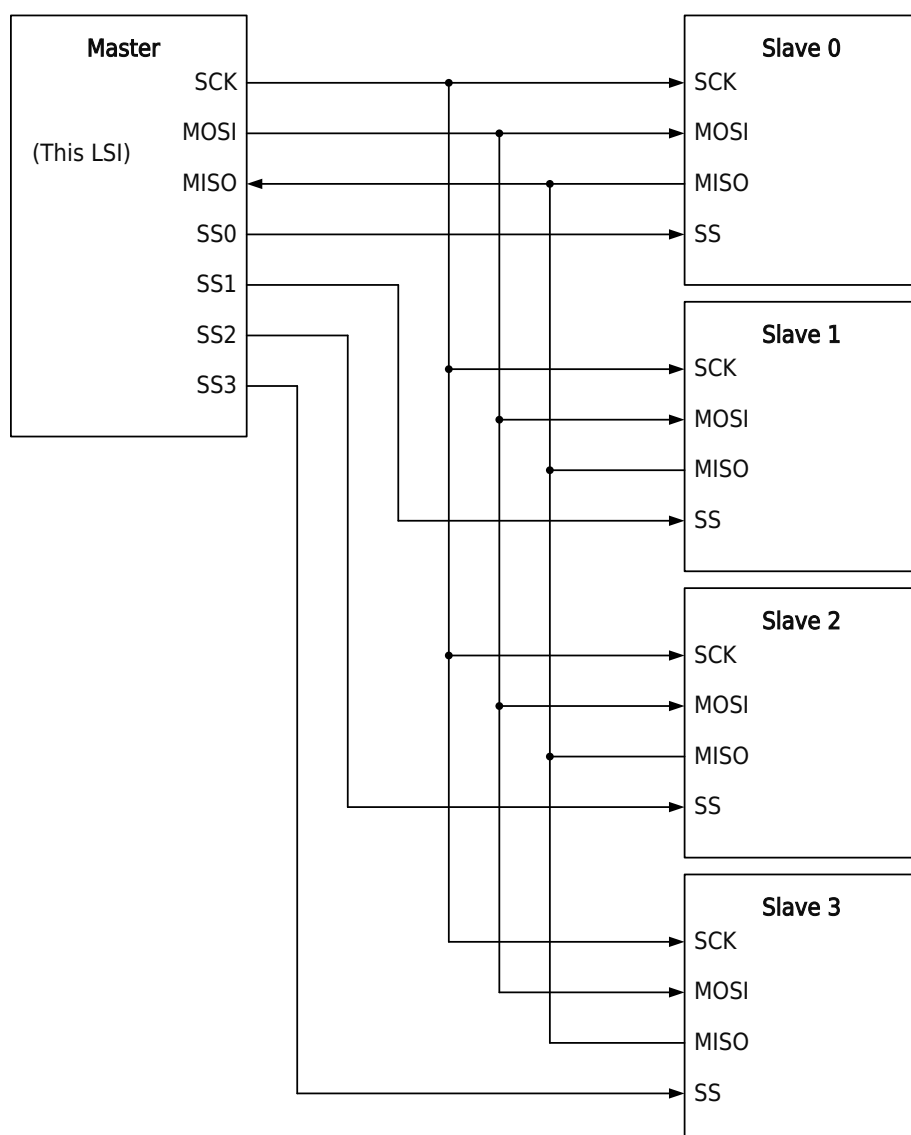


Figure 26-2 Master Mode Structure

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.

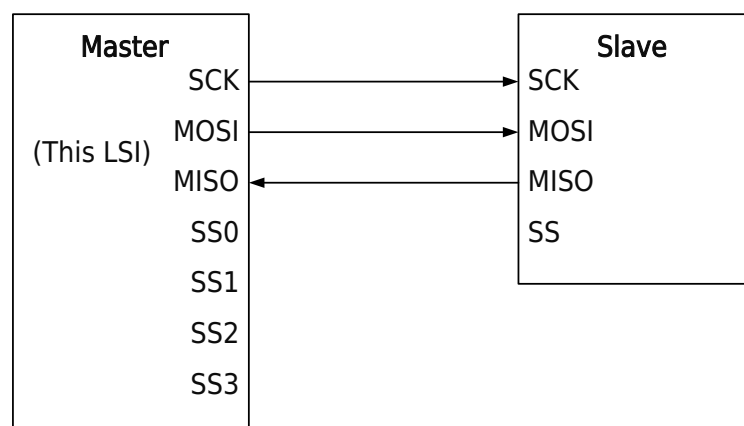


Figure 26-3 Three-wire Clock Synchronization Operation

26.3.3 Data Communication Description

26.3.3.1 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:

$$\text{Baud Rate} = \frac{f_{\text{HCLK}}}{2^{N+1}}$$

Where:

- N: SPI_CFG2.MBR bit setting value, the range is from 0-7.

Table 26-5 Bit Speed of Some Set Values

MBR Bit Setting Value (N)	Frequency Dividing Ratio	Baud Rate			
		HCLK=5MHz	HCLK=10MHz	HCLK=20MHz	HCLK=40MHz
0	2	2.50Mbps	5.00Mbps	10.0Mbps	20.0Mbps
1	4	1.25Mbps	2.50Mbps	5.00Mbps	10.0Mbps
2	8	625kbps	1.25Mbps	2.50Mbps	5.00Mbps
3	16	313kbps	625kbps	1.25Mbps	2.50Mbps
4	32	156kbps	313kbps	625kbps	1.25Mbps
5	64	78kbps	156kbps	313kbps	625kbps
6	128	39kbps	78kbps	156kbps	313kbps
7	256	20kbps	39kbps	78kbps	156kbps

26.3.3.2 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 8/16-bit. 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.

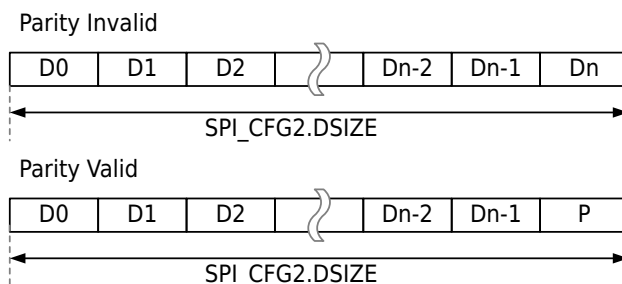


Figure 26-4 Data Format

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 D15-D0 is copied sequentially from TX_BUFF to the SHIFTER and shifted out from the highest bit of the SHIFTER in the sequence of D15-D0.

During receiving, data D15-D0 is shifted in from the least significant bit of the SHIFTER and copied to RX_BUFF after all bits have been received.

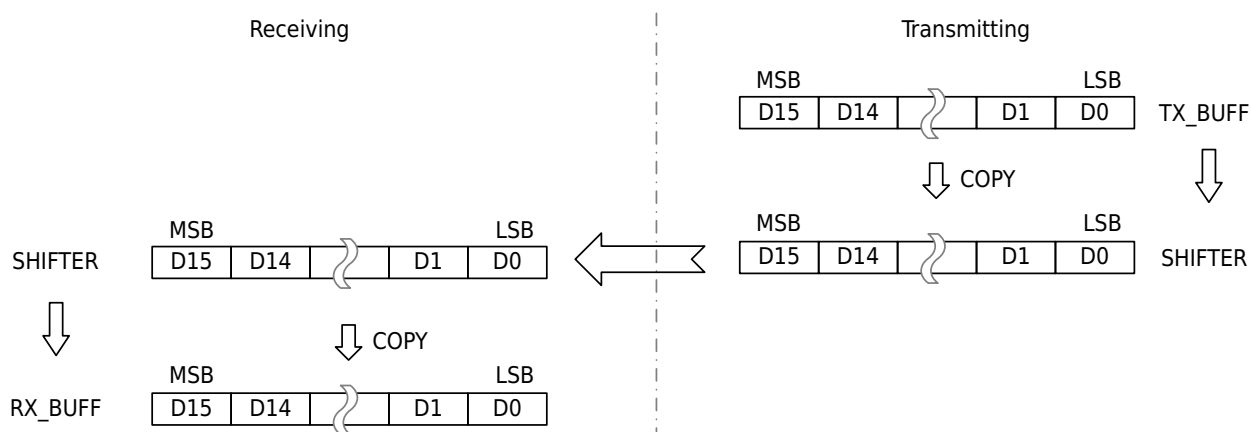


Figure 26-5 MSB Transmitted First, Parity Invalid

● LSB Transmitted First, Parity Invalid

During transmitting, D15-D0 is copied from TX_BUFF to SHIFTER in the sequence of D0-D15, and shifted out from the highest bit of the SHIFTER in the sequence of D0-D15.

During receiving, data D0-D15 is moved in from the lowest bit of SHIFTER, and then copied from SHIFTER to RX_BUFF in the order D15-D0 after all data is moved in.

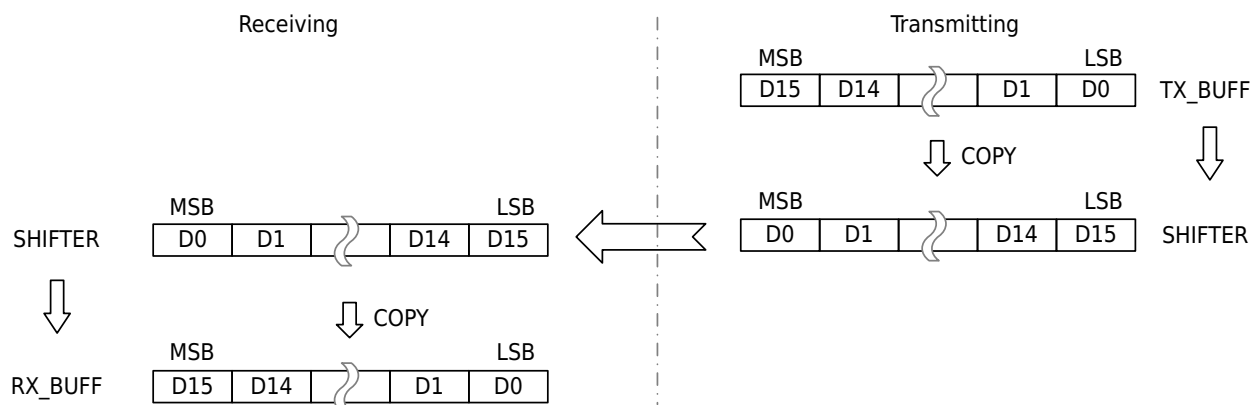


Figure 26-6 LSB Transmitted First, Parity Invalid

● MSB Transmitted First, Parity Valid

During transmitting, the value of parity bit P is calculated according to the value of D15-D1, then P is replaced by D0, copied to SHIFTER according to the order of D15-D1, P, and shifted out from the highest bit of the SHIFTER in the sequence of D15-P.

During receiving, the data D15-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.

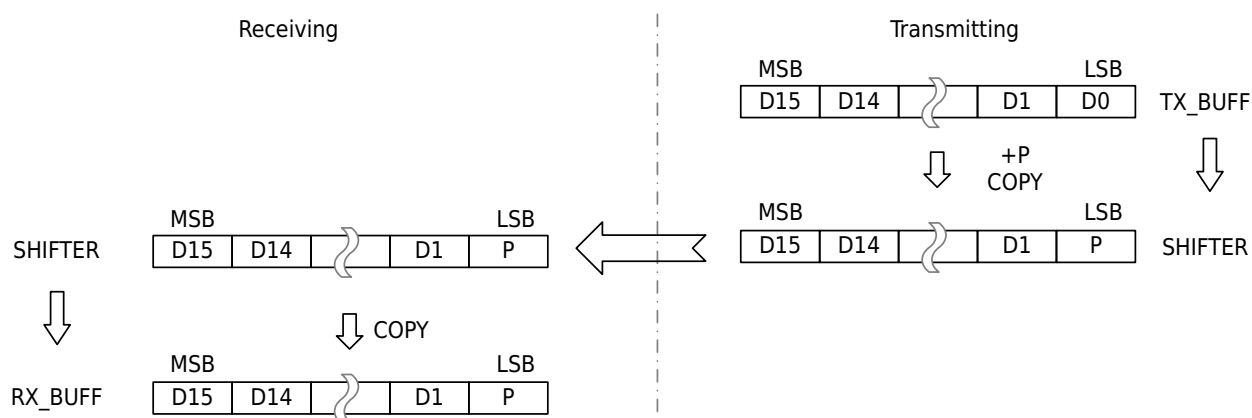


Figure 26-7 MSB Transmitted First, Parity Valid

● LSB Transmitted First, Parity Valid

During transmitting, the value of parity bit P is calculated according to the value of D14-D0, and then P is used instead of D15, 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.

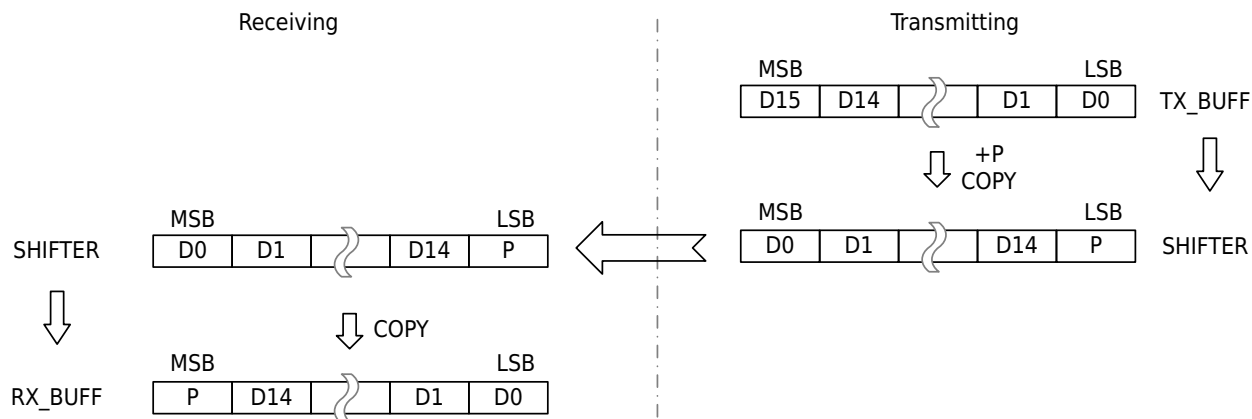


Figure 26-8 LSB Transmitted First, Parity Valid

26.3.3.3 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.

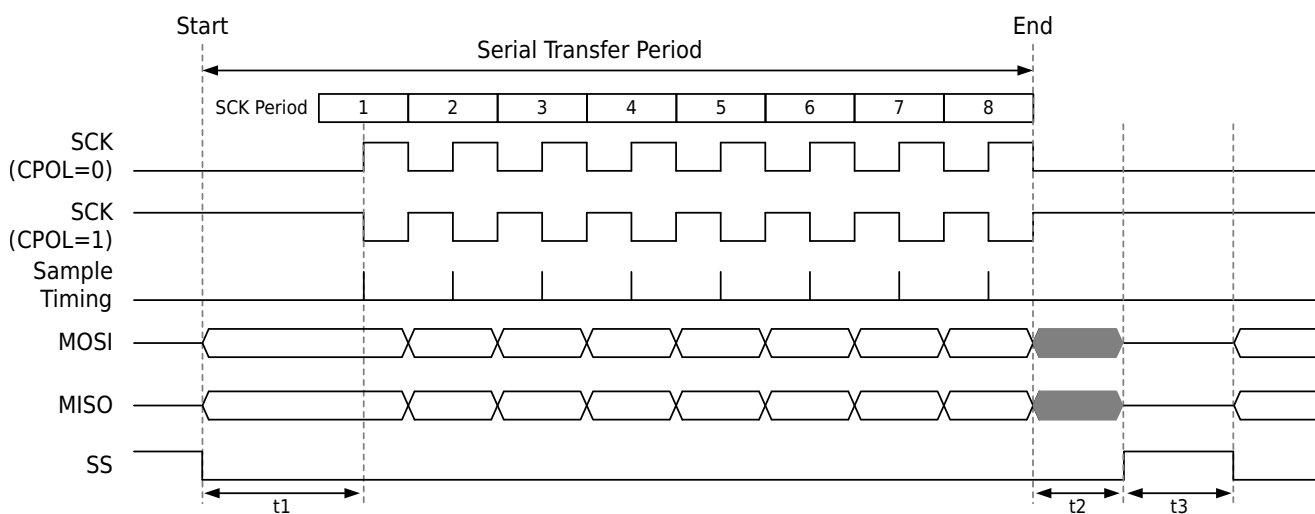


Figure 26-9 Data Transmission Format Diagram (CPHA=0)

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.

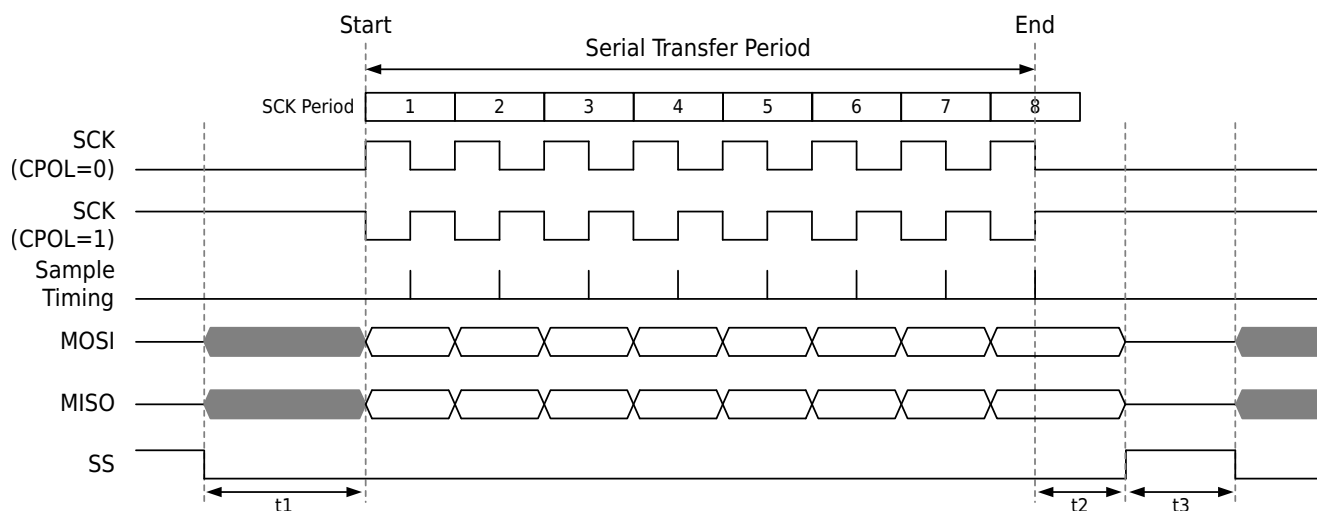


Figure 26-10 Data Transmission Format Diagram (CPHA=1)

26.3.3.4 Communication Method

This SPI supports full-duplex synchronous serial communication mode, transmit-only communication mode, which can be selected through the SPI_CR.TXMDS.

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_CFG2.CPHA bit is 1 and the SPI_CFG2.CPOL bit is 0, and the SPI performs 8-bit serial transmission.

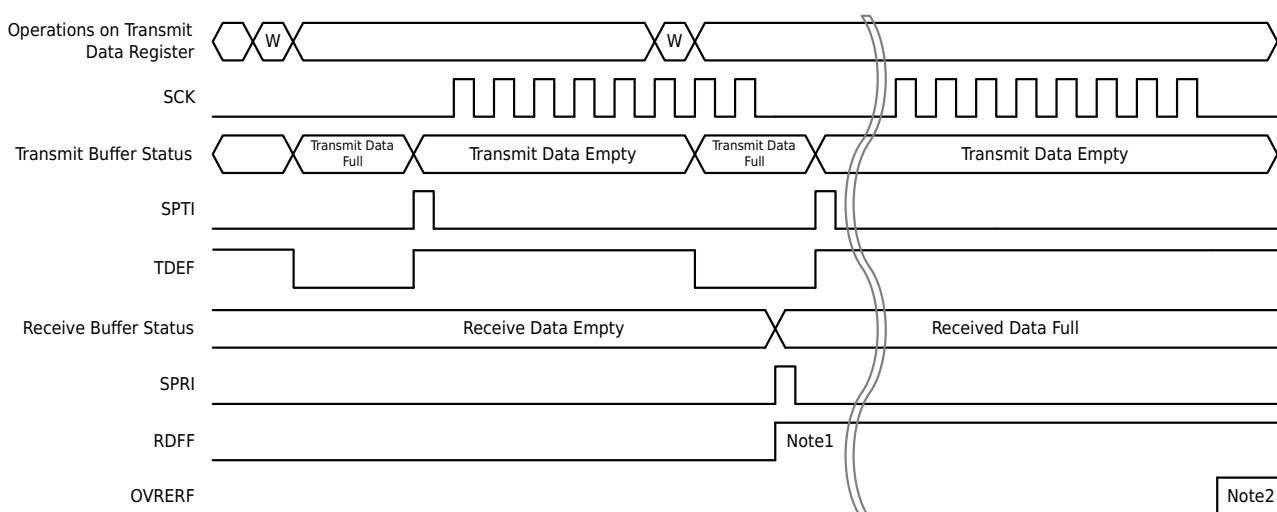


Figure 26-11 Full-duplex Synchronous Serial Communication



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_CFG2.CPHA bit is 1 and the SPI_CFG2.CPOL bit is 0, and the SPI performs 8-bit serial transmission.

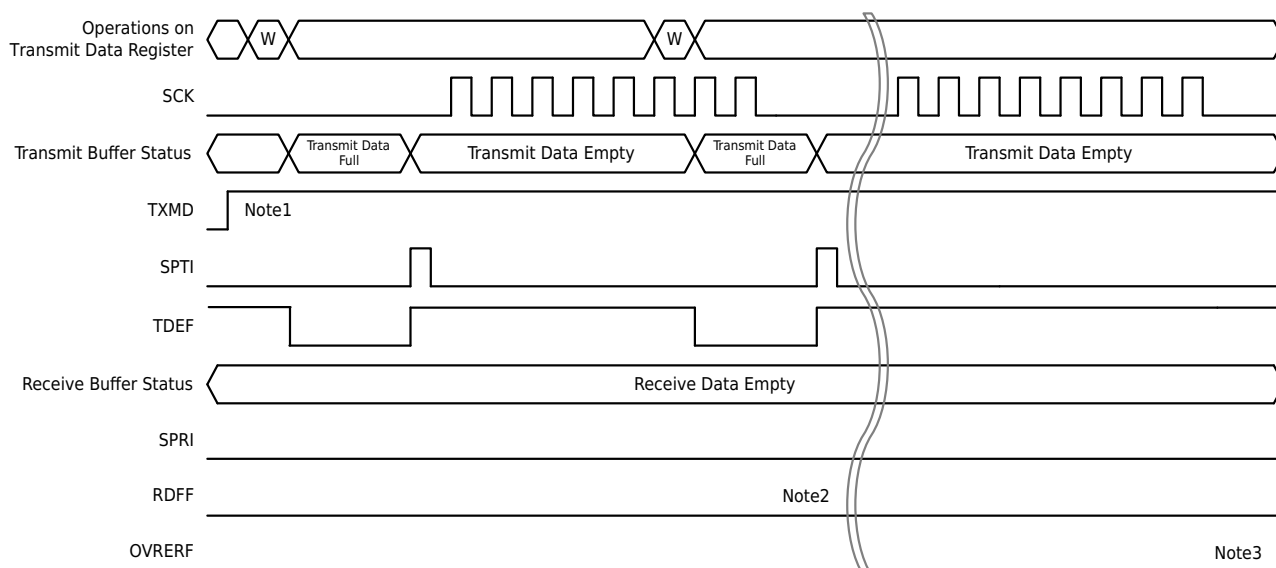


Figure 26-12 Transmit-only Communication



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.

26.3.3.5 Communication Interval

Normal Communication

When the SPI_CFG1.CTMDS bit is set to 0, the SPI uses normal communication setting. There is an interval between two consecutive communications. 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.

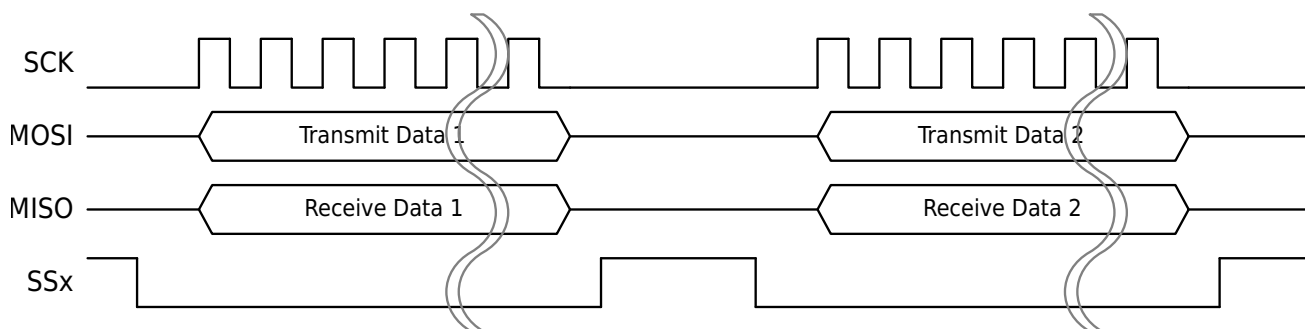


Figure 26-13 Normal Communication

Continuous Communication

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.

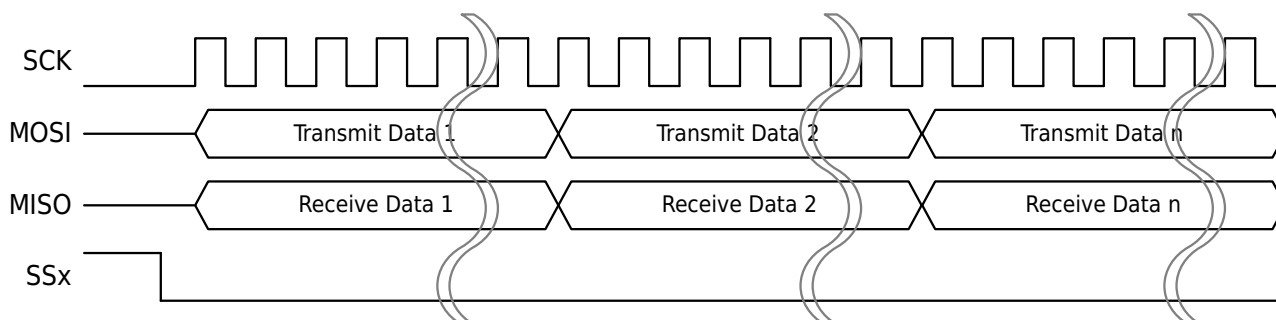


Figure 26-14 Continuous Communication



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.

26.3.4 Operating Instructions

26.3.4.1 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 26-6 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)

Mode	Master (SPI Operation)	Slave (SPI Operation)	Master (Clock Synchronous Operation)	Slave (Clock Synchronous Operation)
SS1-SS3 signal	Output	Hi-Z	Hi-Z (not used)	Hi-Z (not used)
SS polarity change function	Yes	Yes	-	-
Maximum transmission rate	HCLK/2	HCLK/6	HCLK/2	HCLK/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	1 specie (CHPA=1)
Start transfer bit	MSB/LSB	MSB/LSB	MSB/LSB	MSB/LSB
Transmission data length	8/16-bit	8/16-bit	8/16-bit	8/16-bit
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
Transmit buffer empty detection	Yes	Yes	Yes	Yes
Receive buffer full detection	Yes ^{*1}	Yes ^{*1}	Yes ^{*1}	Yes ^{*1}
Overload error detection	Yes ^{*1}	Yes ^{*1}	Yes ^{*1}	Yes ^{*1}
Parity error detection	Yes ^{*1 *2}	Yes ^{*1 *2}	Yes ^{*1 *2}	Yes ^{*1 *2}
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.

26.3.4.2 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 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, etc.;
- Step2.** Set the SPI_CFG2 register, including SS level setting, data shift sequence 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.

26.3.4.3 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;
- 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.

26.3.4.4 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. Due to the absence of SSi input, the mode fault error cannot be detected.

Action Description as Master

When the SPI data transmit buffer (TX_BUFF) is empty (SPI_SR.TDEF is 1), write the data 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, etc.;

- Step2.** Set the SPI_CFG2 register, including the data shift sequence setting, 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.

26.3.4.5 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, normal communication cannot be carried out when the CPHA bit is set to 0.

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;
- 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.

26.3.4.6 SS0 Line Wake-up from Stop Mode Function

When the SPI communication is idle, the system can enter Stop mode to save current consumption. Without changing the SPI PORT configuration, the SPI unit 1 in Slave mode can wake the system from Stop mode via the SS0 line. The specific steps are as follows:

- Step1.** When SPI communication is idle, configure the SPI_1_WUPI interrupt vector and set the INTC_WKEN.SPI1WKEN bit to enable the SPI SS0 signal line wake-up from Stop mode function.
- Step2.** The system enters Stop mode.
- Step3.** When the system detects a falling edge on the SS0 line, it returns from Stop mode. Disable this function in the SPI_1_WUPI interrupt service routine.



Notice

When the communication partner needs to wake up this system, ensure that SPI1 is in Slave mode and not in three-wire mode. Furthermore, the communication partner must wait for the required time for the system to wake from Stop mode before initiating SPI data transmission.

26.3.4.7 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.

26.3.5 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.

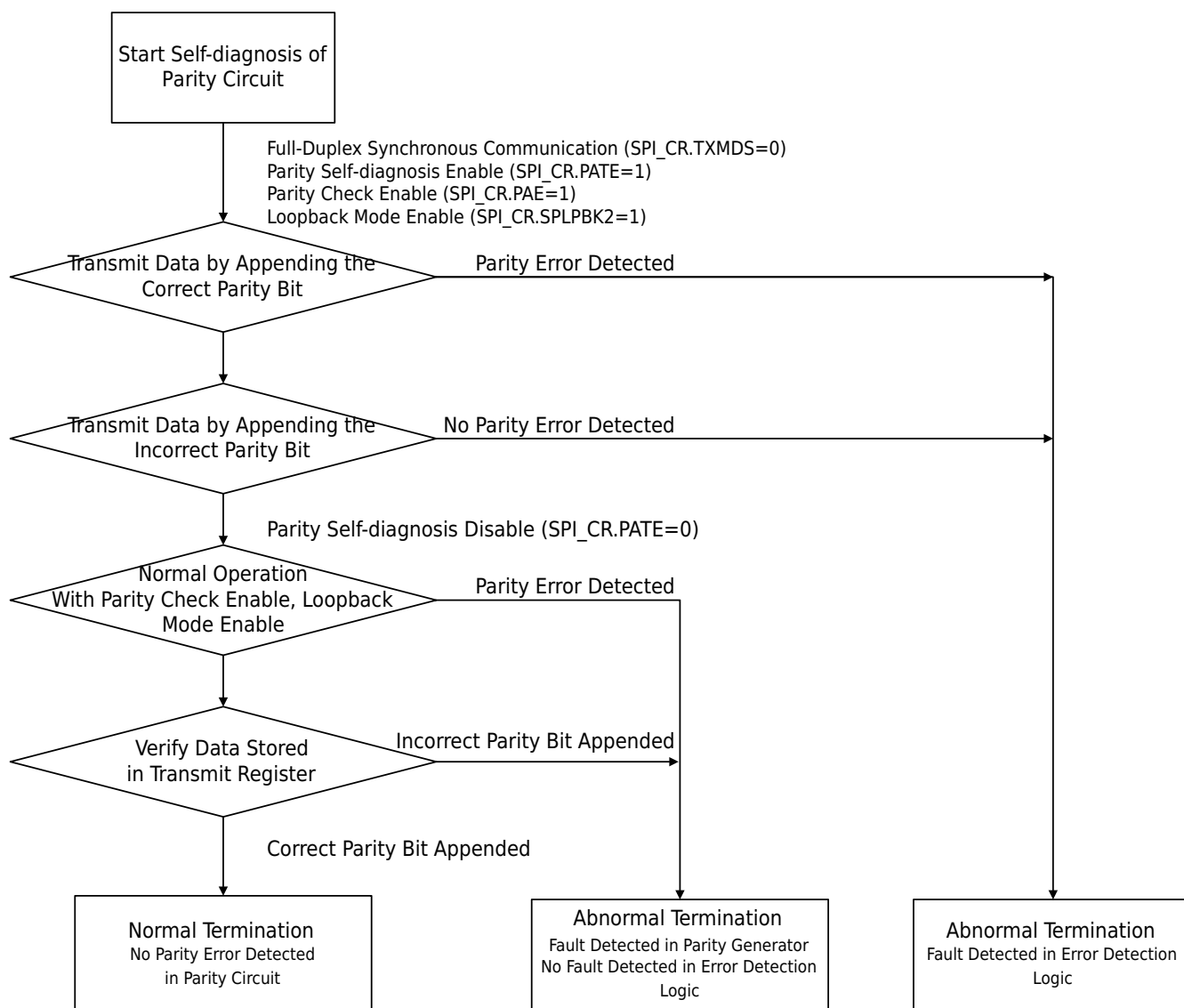


Figure 26-15 Parity Check Process

26.3.6 Error Detection

26.3.6.1 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 26-7 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.

26.3.6.2 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.

26.3.6.3 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.

26.3.6.4 Overload Error

SPI_SR.OVRERF flag is set to 1 (Overload error occurred) if serial delivery is ended when receive buffer is full. 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.

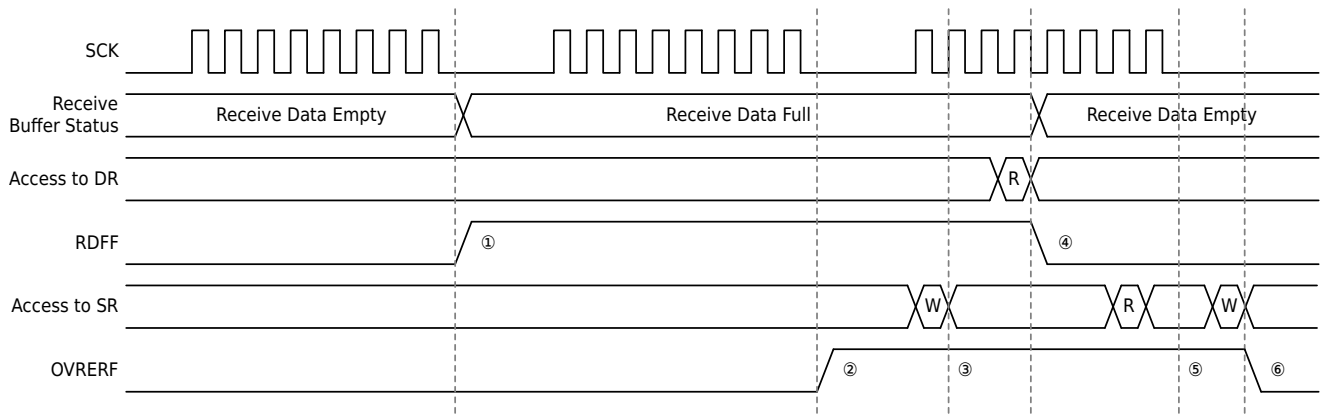


Figure 26-16 Overload Error Processing

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.

26.3.6.5 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.

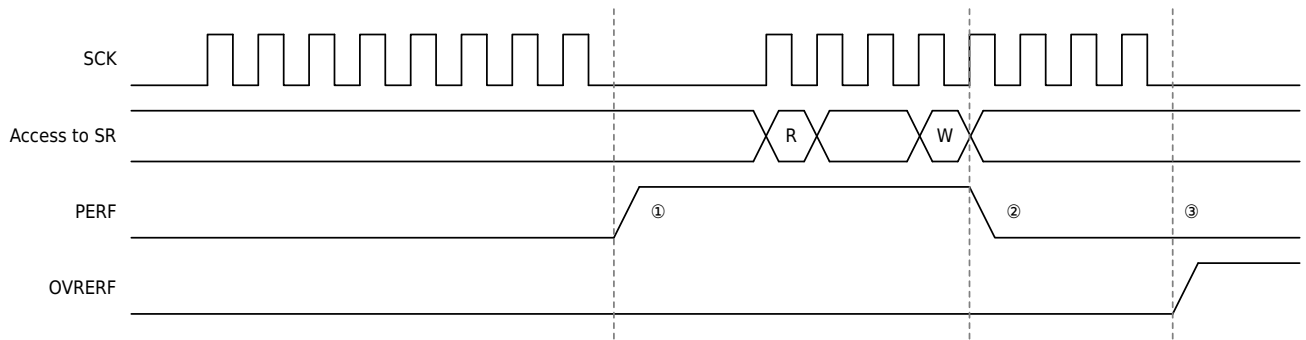


Figure 26-17 Parity Error

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.

26.3.7 SPI Initialization

Write operation or mode fault error detection is used to clear SPE bit, which can inactivate SPI functions and initialize some SPI functions. If a system reset occurs, 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 will initialize all SPI control bits, status bits and data registers.

26.3.8 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 26-8 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: ● Master mode: Data transmissions is completed or SPI_CR.SPE is written from 1 to 0. ● Slave mode: SPI_CR.SPE is written from 1 to 0.
Data transmission completed	SPI_SPEND	-	√	● Master mode: Data transmission completed. ● 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

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.

26.4 Register Description

26.4.1 Register List

- SPI1 base address: 0x40003800
- SPI2 base address: 0x40003C00

Table 26-9 Serial Peripheral Interface (SPI) Register Offset

Offset	Register	Description	Width
0x00	SPI_DR	SPI Data Register	32
0x04	SPI_CR	SPI Control Register	32
0x0C	SPI_CFG1	SPI Config Register 1	32
0x14	SPI_SR	SPI Status Register	32
0x18	SPI_CFG2	SPI Config Register 2	32

26.4.2 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	Reserved																SPD															
Access																																

Bit	Name	Function	Description	Access
31:16	Reserved	Reserved	-	-
15:0	SPD	Serial Data	SPI data storage	RW

26.4.3 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
Reset	0x00000000															
Name	Reserved												PAE	PAOE	PATE	MODFE
Access													RW	RW	RW	RW
													Reserved	SPE	SPLPBK2	SPLPBK
													Reserved	MSTR	Reserved	TXMDS
															RW	SPIMDS

Bit	Name	Function	Description	Access
31:16	Reserved	Reserved	-	-
15	PAE	Parity Enabled	0b0: Transmit data without parity bit, receive data without parity check 0b1: Transmit data with parity bit, receive data with parity check (SPI_CR.TXMDS=0); Transmit data with parity bit, receive data without parity check (SPI_CR.TXMDS=1)	RW
14	PAOE	Parity Mode Selection	0b0: Select even check for sending and receiving 0b1: Select odd check for sending and receiving	RW
13	PATE	Parity Self-diagnosis	0b0: Parity self-diagnosis function inactive 0b1: Parity self-diagnosis function active	RW
12	MODFE	Mode Fault Error Detection Enabled	0b0: Disable mode fault error detection 0b1: Enable mode fault error detection	RW
11	IDIE	SPI Idle Interrupt Enabled	0b0: Disable generation of idle interrupt requests 0b1: Enable generation of idle interrupt requests	RW
10	RXIE	SPI Receive Interrupt Enabled	0b0: Disable generation of SPI receive interrupt requests 0b1: Enable generation of SPI receive interrupt requests	RW
9	TXIE	SPI Transmit Interrupt Enabled	0b0: Disable generation of SPI transmit interrupt requests 0b1: Enable generation of SPI transmit interrupt requests	RW
8	EIE	SPI Error Interrupt Enabled	0b0: Disable generation of SPI error interrupt requests 0b1: Enable generation of SPI error interrupt requests	RW
7	Reserved	Reserved	-	-

Bit	Name	Function	Description	Access
6	SPE	SPI Function Enabled	● 0b0: SPI function disabled ● 0b1: SPI function enabled	RW
5	SPLPBK2	SPI Loop 2-bit	● 0b0: Normal mode ● 0b1: Loopback mode (transmit data=receive data)	RW
4	SPLPBK	SPI Loopback Position	● 0b0: Normal mode ● 0b1: Loopback mode (inverted phase of transmit data=received data)	RW
3	MSTR	SPI Master-slave Mode Selection	● 0b0: Slave mode ● 0b1: Master mode	RW
2	<i>Reserved</i>	<i>Reserved</i>	-	-
1	TXMDS	Communication Mode Selection	● 0b0: Full-duplex synchronous serial communication ● 0b1: Transmit serial communication only	RW
0	SPIMDS	SPI Mode Selection	● 0b0: SPI operation (4-wire) ● 0b1: Clock synchronous operation (3-wire)	RW

26.4.4 SPI Config 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	0x00000000																															
Name	Reserved																				SS3PV	SS2PV	SS1PV	SS0PV	Reserved				CTMDS	Reserved		
Access																					RW	RW	RW	RW					RW			

Bit	Name	Function	Description	Access
31:12	Reserved	Reserved	-	-
11	SS3PV	Polarity Setting of SS3 Signal	0b0: Low level active 0b1: High level active	RW
10	SS2PV	Polarity Setting of SS2 Signal	0b0: Low level active 0b1: High level active	RW
9	SS1PV	Polarity Setting of SS1 Signal	0b0: Low level active 0b1: High level active	RW
8	SS0PV	Polarity Setting of SS0 Signal	0b0: Low level active 0b1: High level active	RW
7:3	Reserved	Reserved	-	-
2	CTMDS	Continuous Communication Mode Selection	0b0: Normal communication mode 0b1: Continuous communication mode	RW
1:0	Reserved	Reserved	-	-

26.4.5 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 0b0: The receive buffer is not full 0b1: The receive buffer is full	RO
6	FMIDLF	IDLE Status Flag of Master in Continuous Mode	Hardware set and clear 0b0: Continuous mode master is in normal operation state 0b1: Continuous mode master is in idle state	RO
5	TDEF	Transmit Buffer Empty Flag	Hardware set and clear 0b0: Data present in transmit buffer 0b1: No data present in the transmit buffer	RO
4	UDRERF	Underload Error Flag	Hardware set; read this bit as 1 and then write 0 to clear the status bit 0b0: No underload error occurred 0b1: Underload error occurred (MODFERF=1)  Note When MODFERF=0, this bit will be initialized.	RW
3	PERF	Parity Error Flag	Hardware set; read this bit as 1 and then write 0 to clear the status bit 0b0: No parity error occurred 0b1: Parity error occurred	RW
2	MODFERF	Mode Fault Error Flag	Hardware set; read this bit as 1 and then write 0 to clear the status bit 0b0: No mode fault error occurred 0b1: Mode fault error occurred	RW
1	IDLNF	SPI Idle Flag	Hardware set and clear 0b0: SPI is in idle state 0b1: SPI is in transmission state	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 ● 0b0: No overload error occurred ● 0b1: Overload error occurred	RW

26.4.6 SPI Config 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	0x0000011D																															
Name	Reserved														LSBF	Reserved		DSIZE	SSA		MBR		CPOL	CPHA								
Access															RW			RW	RW		RW		RW	RW								

Bit	Name	Function	Description	Access
31:13	Reserved	Reserved	-	-
12	LSBF	SPI LSB First Bit	0b0: MSB first 0b1: LSB first	RW
11:9	Reserved	Reserved	-	-
8	DSIZE	SPI Data Length Setting Bit	0b0: 8 bits 0b1: 16 bits	RW
7:5	SSA	SS Signal Enabled Setting Bit	0b000: SS0 0b001: SS1 0b010: SS2 0b011: SS3 - Others: Setting prohibited	RW
4:2	MBR	Bit Rate Frequency Division Setting Bit	0b000: Select the 2-division of the reference communication clock 0b001: Select the 4-division of the reference communication clock 0b010: Select the 8-division of the reference communication clock 0b011: Select the 16-division of the reference communication clock 0b100: Select the 32-division of the reference communication clock 0b101: Select the 64-division of the reference communication clock 0b110: Select the 128-division of the reference communication clock 0b111: Select the 256-division of the reference communication clock	RW
1	CPOL	SCK Polarity Setting Bit	0b0: SCK is low level when idle 0b1: SCK is high level when idle	RW
0	CPHA	SCK Phase Setting Bit	0b0: Data sampling at odd edges, changing at even edges 0b1: Data changing at odd edges, sampling at even edges	RW

27 Controller Area Network (MCAN)

27.1 Introduction

The Controller Area Network (CAN) subsystem consists of one CAN modules, a shared message RAM memory for the base addresses of these components, see the "Memory Mapping" chapter.

CAN modules (MCAN) comply with ISO 11898-1:2015 (CAN Protocol Specification Version 2.0 Parts A, B) and CAN with Flexible Data-Rate Specification Version 1.0.

The 1KB 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 features ECC (Error Checking and Correcting). The ECC is a single-error-correcting, double-error-detecting (SECCDED) code, meaning it can correct a single-bit error and detect a double-bit error.

27.2 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
- Supports programmable loopback test mode
- Supports maskable module interrupts
- Two clock domains: CAN core clock and AHB bus interface clock

27.3 Functional Description

27.3.1 MCAN System Block Diagram

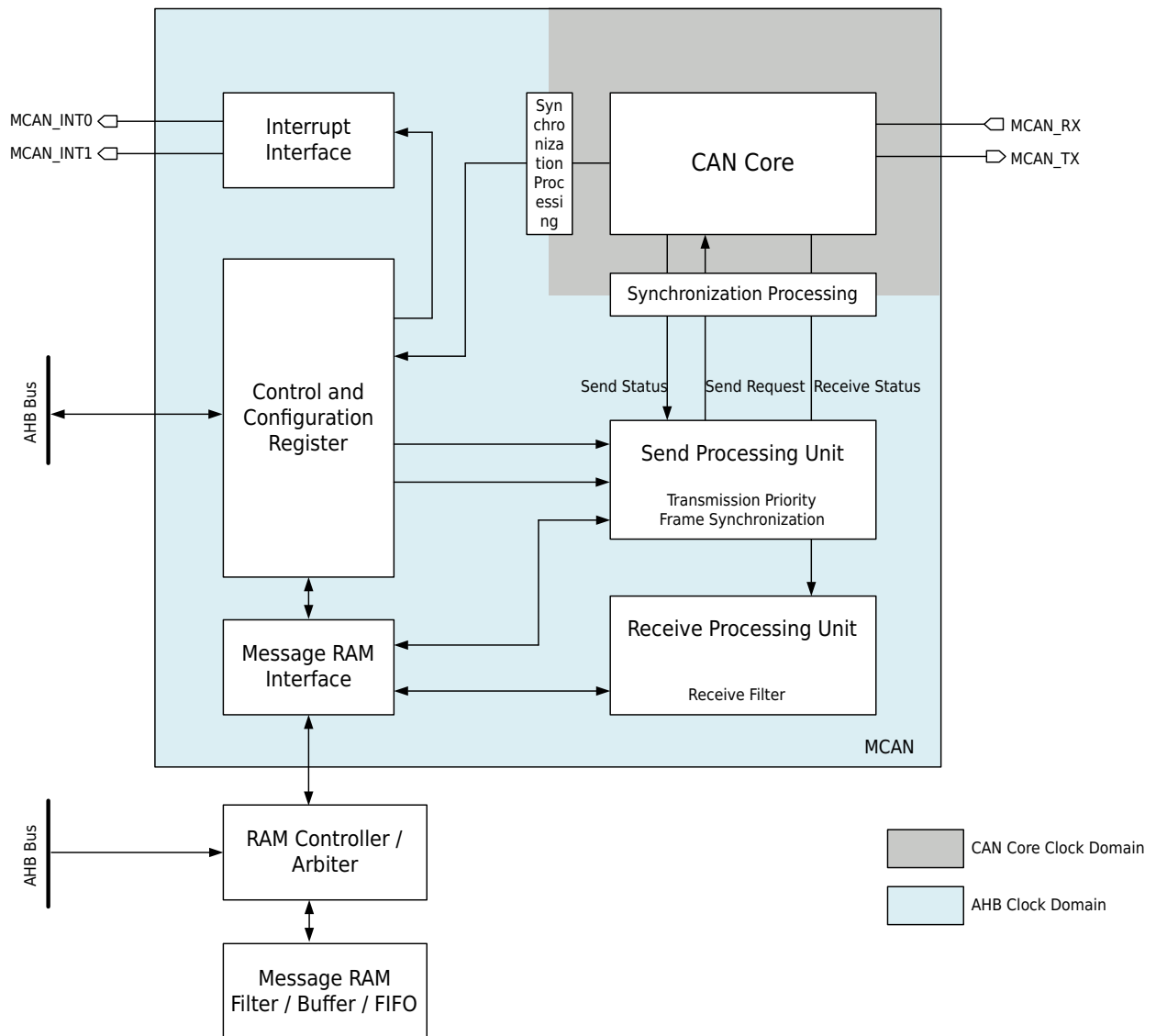


Figure 27-1 MCAN System Block Diagram

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 AHB clock domain to the CAN core clock domain, and synchronizes the signal from the CAN core clock domain to the AHB clock domain.

The AHB 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.

AHB Interface

Connects the MCAN to the AHB bus.

Message RAM

The interface connects MCAN access to an external 1KB 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.

27.3.2 Pin Description

Table 27-1 MCAN Pin Description

Pin Name	Direction	Description
MCAN_RX	Input	MCAN receive pin
MCAN_TX	Output	MCAN transmit pin

27.3.3 Working Mode

27.3.3.1 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 27-2 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.

27.3.3.2 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.

27.3.3.3 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 27-3 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.

27.3.3.4 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.

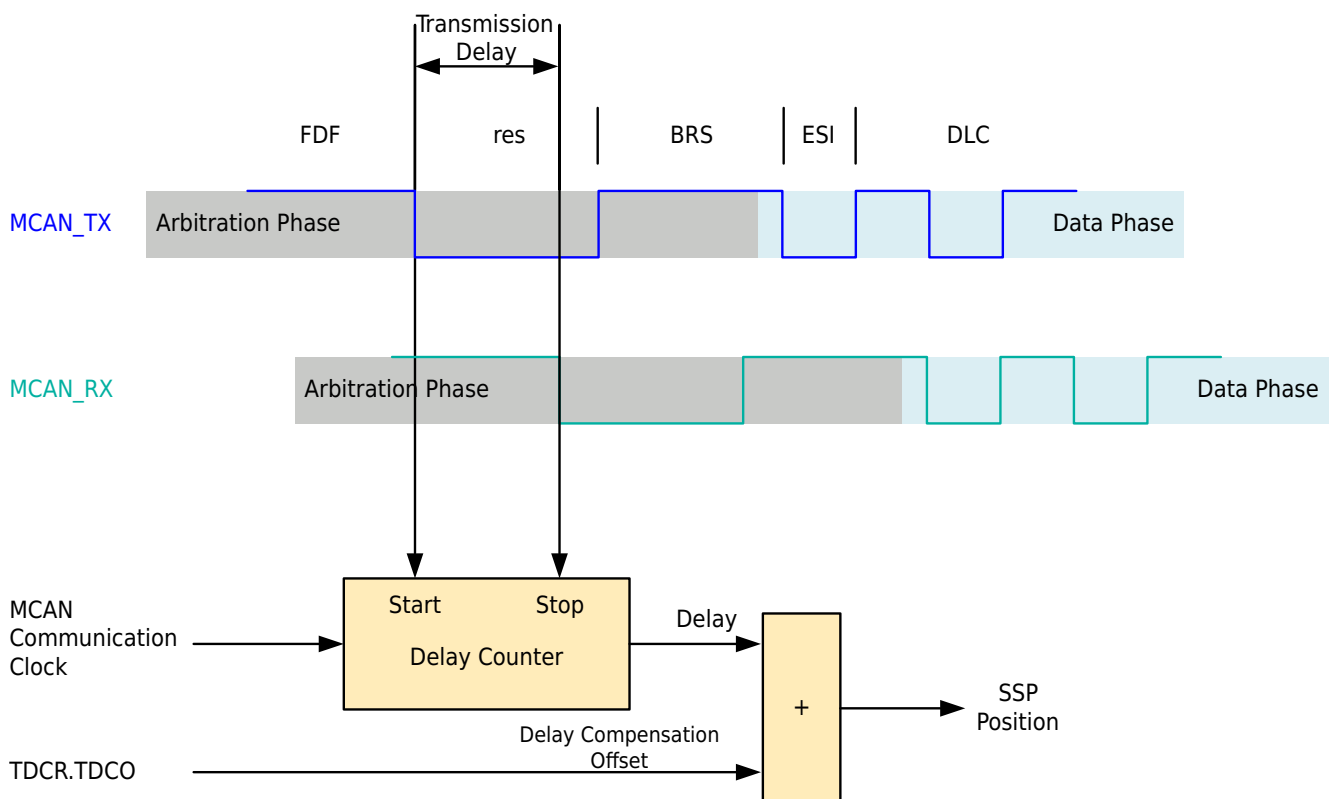


Figure 27-2 Transceiver Delay Measurement

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.

27.3.3.5 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).

27.3.3.6 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 recessive bits on the CAN bus. If MCAN needs to send a dominant bit (such as the ACK bit, overload flag, or active error flag), that bit is internally rerouted to allow MCAN to listen to the dominant bit, while the CAN bus remains in the recessive 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 dominant bits. The following figure shows the connection between MCAN_TX and MCAN_RX signals and MCAN in bus monitoring mode.

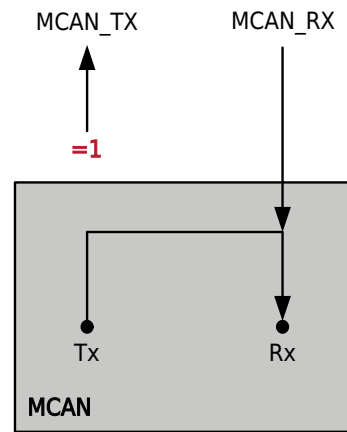


Figure 27-3 Pin Control in Bus Monitoring Mode

27.3.3.7 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).

27.3.3.8 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.

27.3.3.9 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.

27.3.3.10 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.

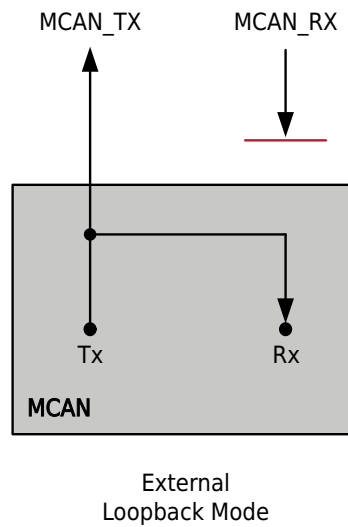


Figure 27-4 External Loopback Mode Pin Control

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).

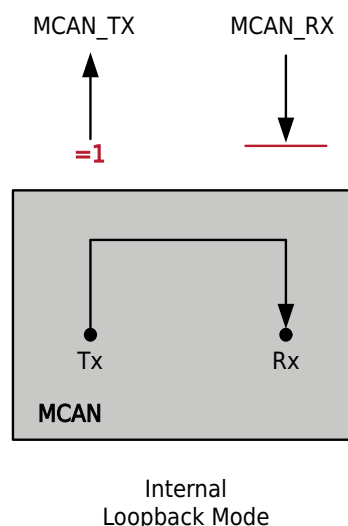


Figure 27-5 Pin Control for Internal Loopback Mode

27.3.4 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.

27.3.5 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.

27.3.6 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.

27.3.6.1 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. 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.

27.3.6.2 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

27.3.6.3 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.

27.3.6.4 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.

27.3.6.5 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.

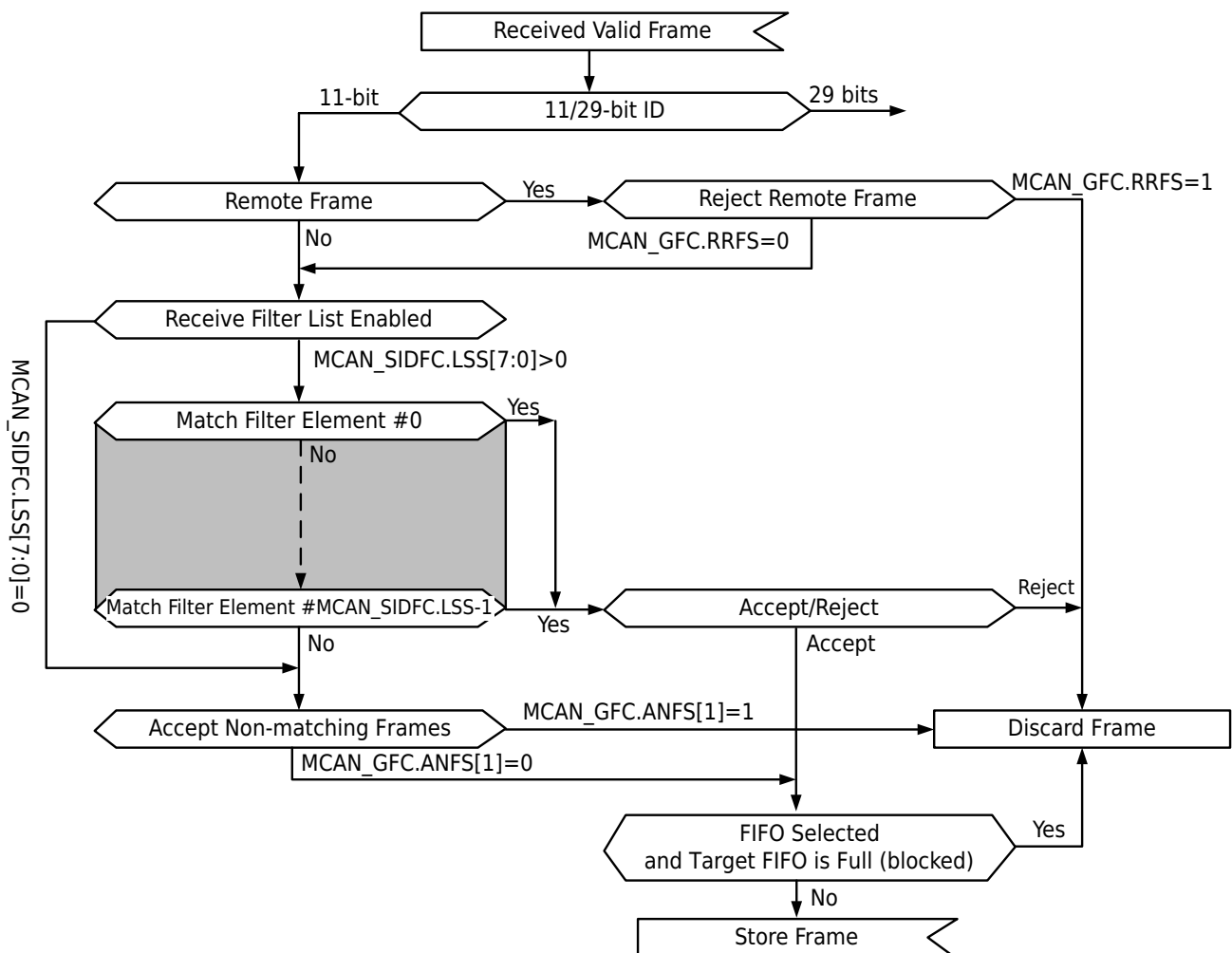


Figure 27-6 Standard Message ID Filter Path

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.

27.3.6.6 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.

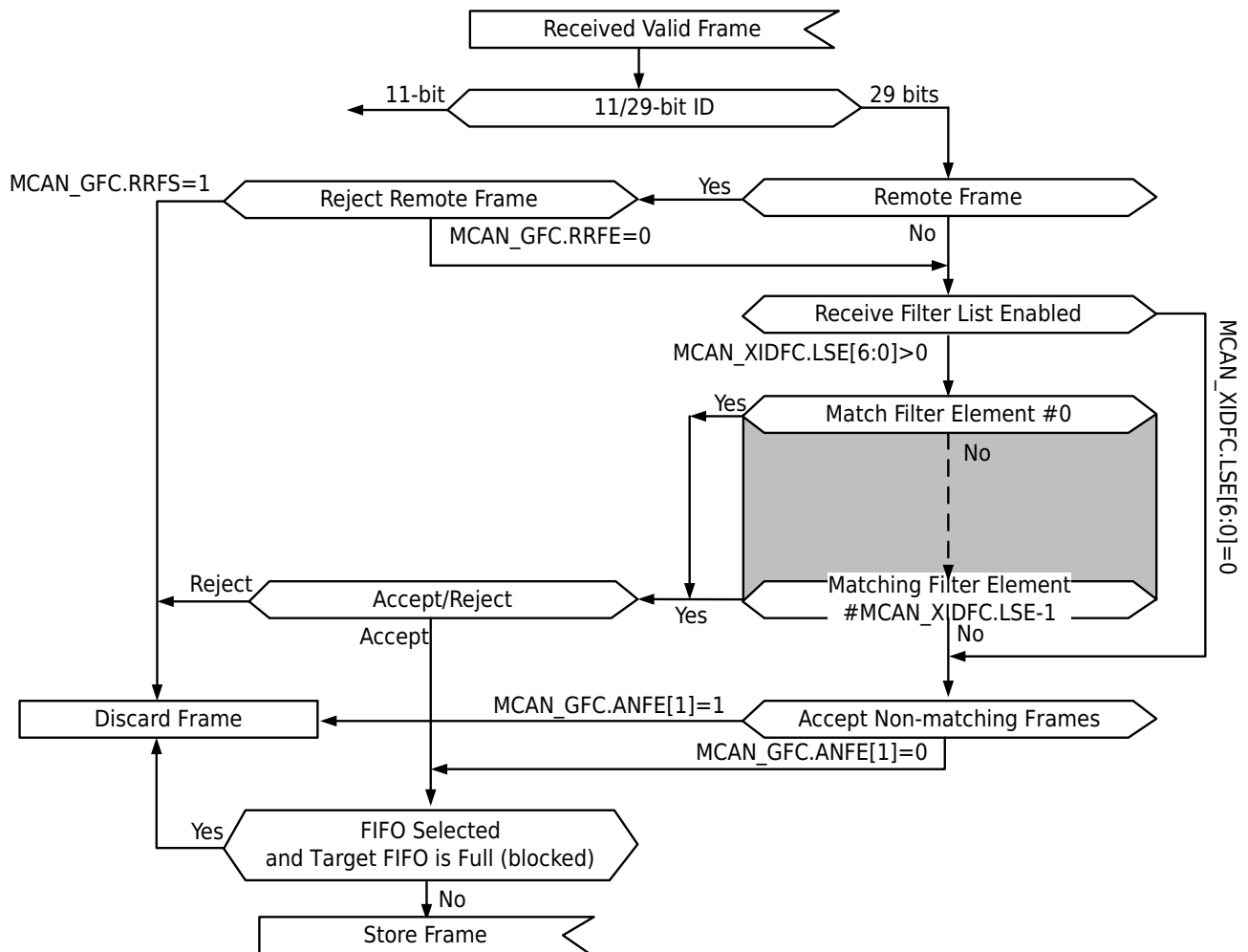


Figure 27-7 Extended Message ID Filter Path

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.

27.3.6.7 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.

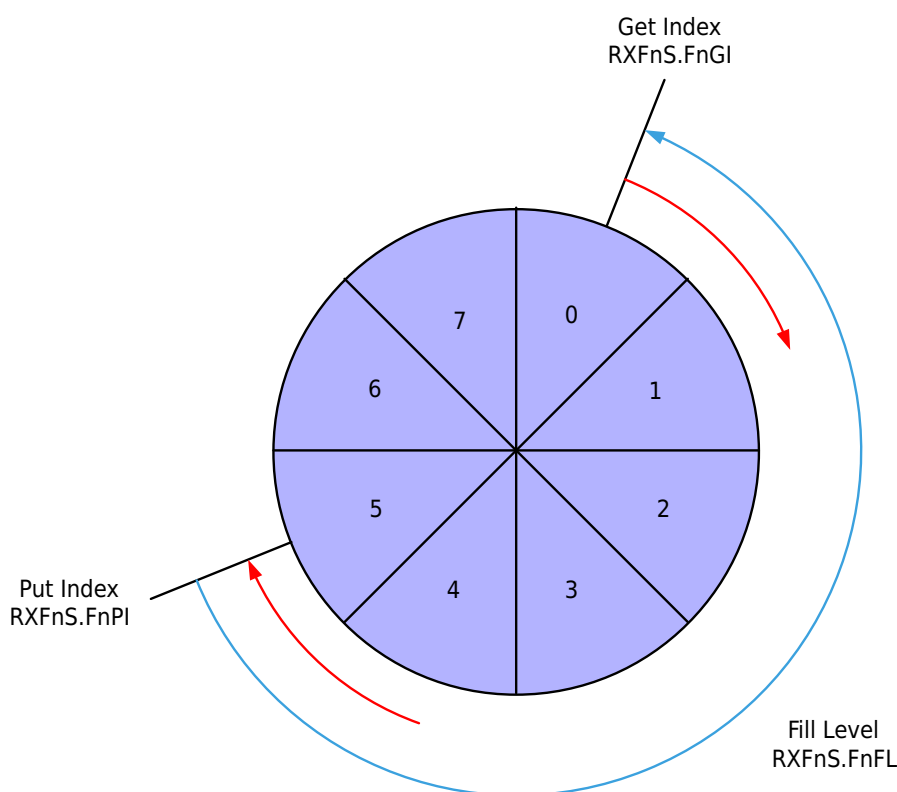


Figure 27-8 Receive FIFO Status

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 27-4 Receive buffer and FIFO element size

<code>RXESC.RBDS[2:0]</code> <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

27.3.6.8 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.

27.3.6.9 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.

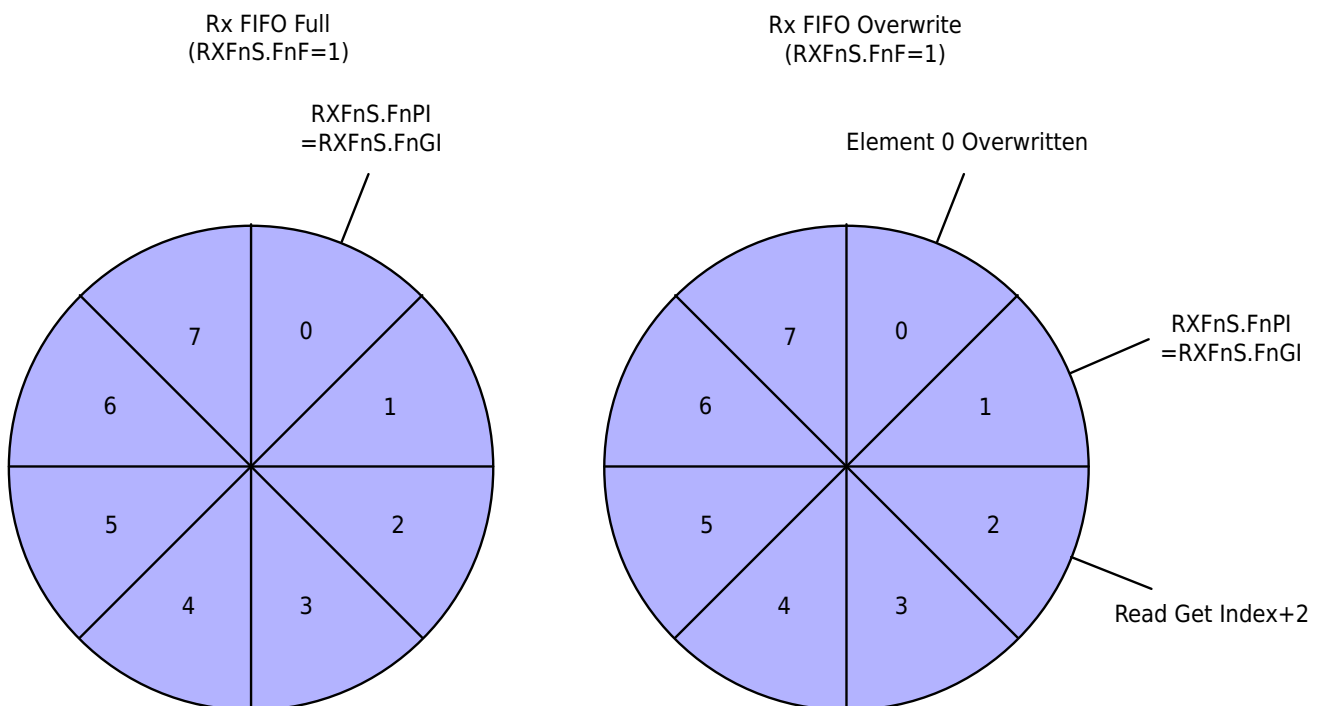


Figure 27-9 Receive FIFO Overflow Processing

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`).

27.3.6.10 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 27-5 Receive Buffer Filter Configuration Example

Filter Elements	SFID1[10:0] EFID1[28:0]	SFID2[10:9] EFID2[10:9]	SFID2[5:0] 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.

27.3.6.11 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

27.3.6.12 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 27-6 Debug Message Filter Configuration Example

Filter Elements	SFID1[10:0] EFID1[28:0]	SFID2[10:9] EFID2[10:9]	SFID2[5:0] EFID2[5:0]
0	ID Debug Message A	0b01	0b111101
1	ID Debug Message B	0b10	0b111110
2	ID Debug Message C	0b11	0b111111

27.3.7 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 27-7 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.

27.3.7.1 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.

27.3.7.2 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 27-8 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

27.3.7.3 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.

27.3.7.4 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.

27.3.7.5 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.

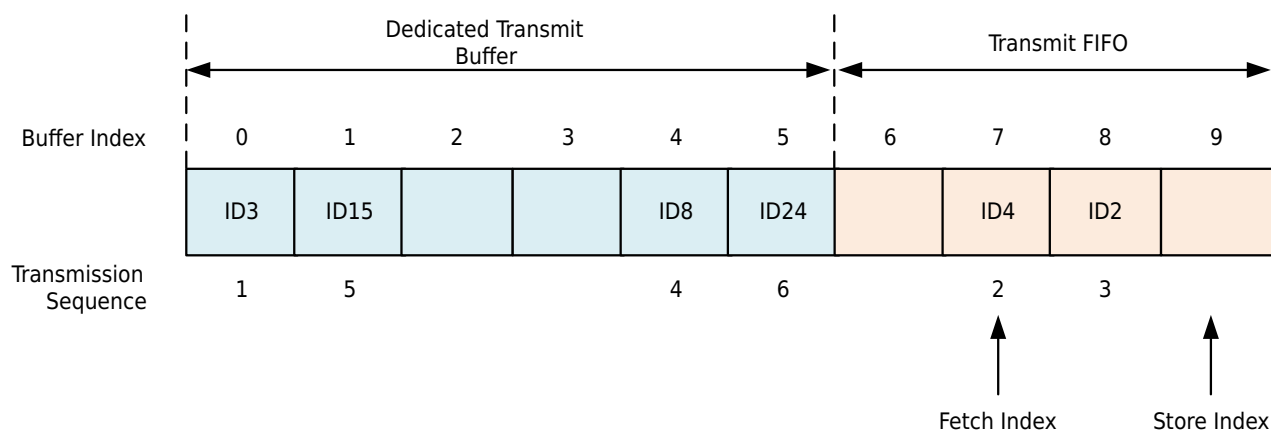


Figure 27-10 Example of Mixed Configuration of Dedicated Transmit Buffers and Transmit FIFO

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.

27.3.7.6 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.

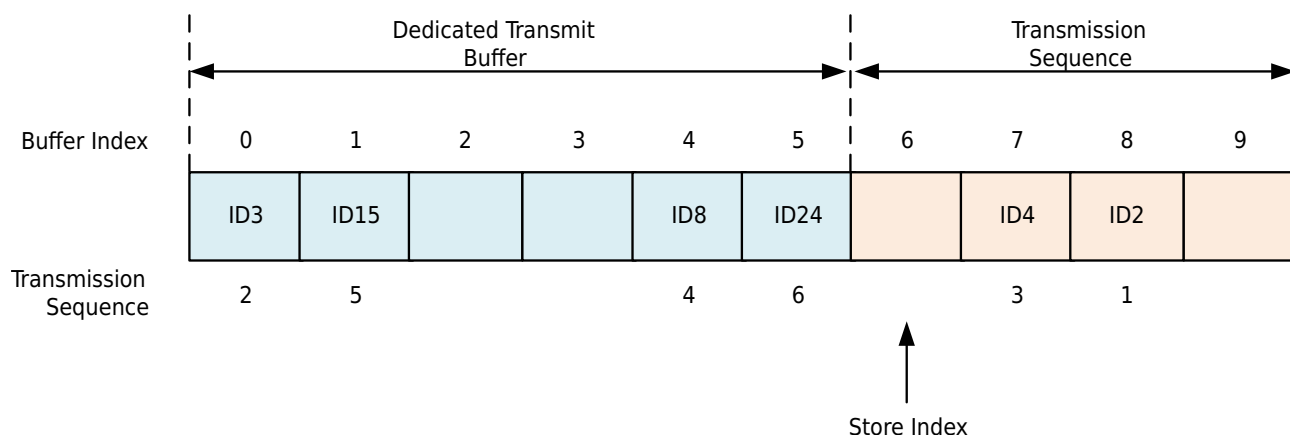


Figure 27-11 Example of Mixed Configuration of Dedicated Transmit Buffers and Transmit Queue

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.

27.3.7.7 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.

27.3.7.8 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.

27.3.8 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.

27.3.9 Message RAM

27.3.9.1 Message RAM Configuration

The width of the message RAM is 32 bits. The MCAN module can allocate up to 256 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.

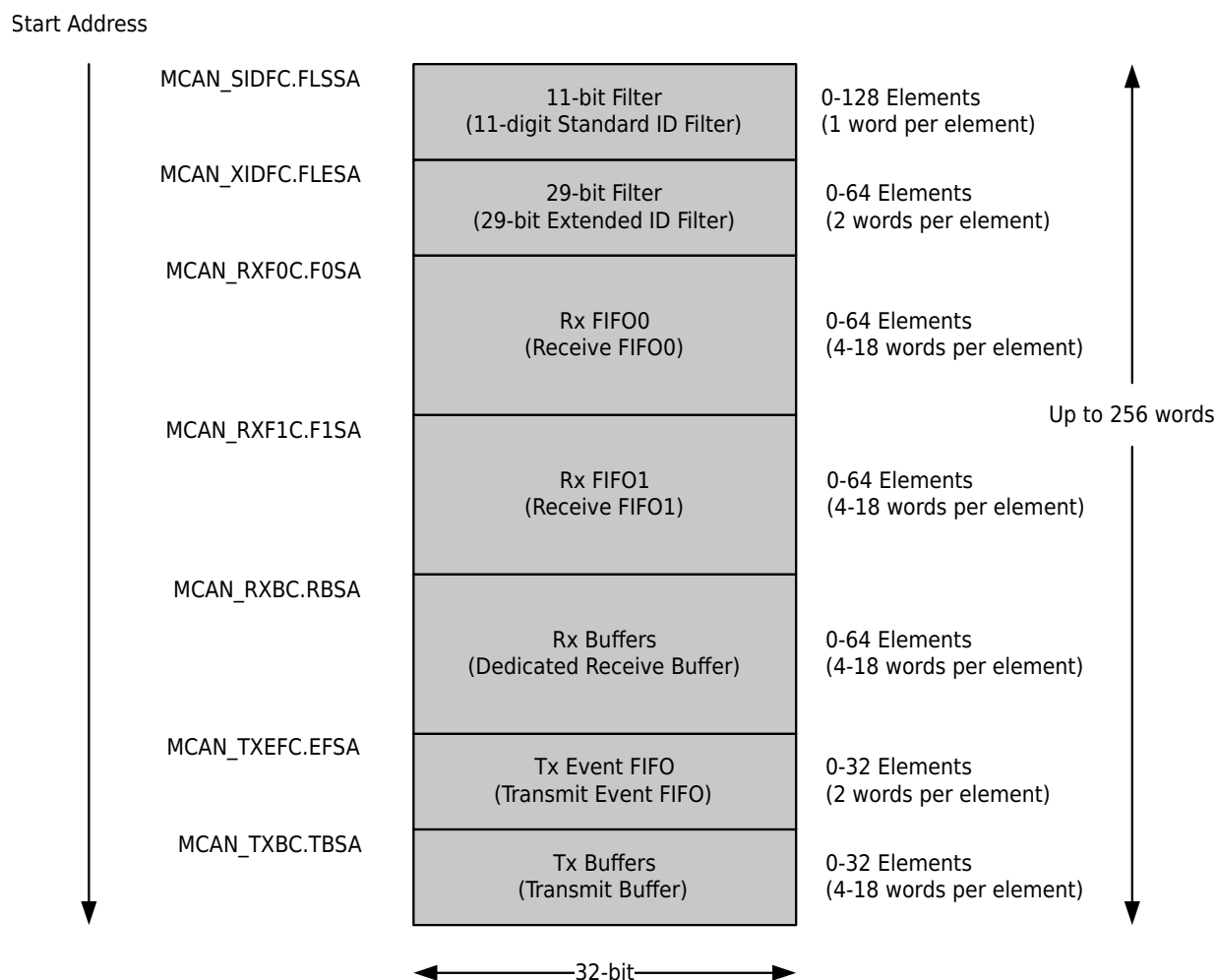


Figure 27-12 Message RAM Configuration

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.

27.3.9.2 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 27-9 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 27-10 Receive Buffer and FIFO Elements Description

Bit	Name	Description
R0[31]	ESI	Error State Indicator ● 0b0: The sending node is in error active state ● 0b1: The sending node is in error passive state
R0[30]	XTD	Extended Identifier Indicates to the host whether the ID of the received frame is a standard ID or an extended ID. ● 0b0: 11-bit standard ID ● 0b1: 29-bit extended ID
R0[29]	RTR	Remote Transmission Request Indicates to the host whether the received frame is a data frame or a remote frame. ● 0b0: The received frame is a data frame ● 0b1: The received frame is a remote frame  Note 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.

Bit	Name	Description
R0[28:0]	ID[28:0]	ID (Identifier) 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 Acceptance of non-match frames can be enabled via MCAN_GFC.ANFS and MCAN_GFC.ANFE. ● 0b0: The received frame matches the filter element pointed to by filter index FIDX ● 0b1: The received frame does not match any receive filter element
R1[30:24]	FIDX[6:0]	Filter Index ● 0-127: Index of the matching receive filter element (inactive when ANMF=1) The range is 0 to MCAN_SIDFC.LSS] minus 1 or MCAN_XIDFC.LSE minus 1.
R1[21]	FDF	FD Format ● 0b0: Classic CAN frame format. ● 0b1: CAN FD frame format (new DLC encoding and CRC).
R1[20]	BRS	Bit Rate Switch ● 0b0: Receive frame without bit rate switching ● 0b1: Receive frame with bit rate switching
R1[19:16]	DLC[3:0]	Data Length Code ● 0-8: Classic CAN and CAN FD: The received frame contains 0-8 data bytes ● 9-15: CAN: The received frame contains 8 data bytes ● 9-15: CAN FD: The received frame contains 12/16/20/24/32/48/64 data bytes
R1[15:0]	RXTS[15:0]	Rx Timestamp The timestamp counter value captured when a SOF (Start Of Frame) is received. 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

27.3.9.3 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 27-11 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 27-12 Transmit Buffer Elements Description

Bit	Name	Description
T0[31]	ESI	Error State Indicator ● 0b0: The ESI bit of the CAN FD format depends only on the error passive flag ● 0b1: The ESI bit of the CAN FD format is sent as a recessive bit  Note 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.
T0[30]	XTD	Extended Identifier ● 0b0: 11-bit standard ID ● 0b1: 29-bit extended ID
T0[29]	RTR	Remote Transmission Request ● 0b0: Transmit data frame ● 0b1: Transmit remote frame  Note 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.
T0[28:0]	ID[28:0]	ID (Identifier) Standard ID or extended ID, depending on the XTD bit, the standard ID must be written in ID[28:18]
T1[31:24]	MM[7:0]	Message Marker Written by the CPU during transmit buffer configuration. Copied to the send event FIFO element to identify the send message status.
T1[23]	EFC	Event FIFO Control ● 0b0: Do not store transmission events ● 0b1: Store transmission events
T1[21]	FDF	FD Format ● 0b0: Send frames in classic CAN format ● 0b1: Send frames in CAN FD format

Bit	Name	Description
T1[20]	BRS	Bit Rate Switch ● 0b1: Switch bitrate when sending CAN FD frame ● 0b0: Do not switch bitrate when sending CAN FD frame  Note 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.
T1[19:16]	DLC[3:0]	Data Length Code ● 0-8: Classic CAN and CAN FD: The transmitted frame contains 0 to 8 data bytes. ● 9-15: CAN: The transmitted frame contains 8 data bytes. ● 9-15: CAN FD: The transmitted frame contains 12/16/20/24/32/48/64 data bytes.
T1[15:8]	MM[15:8]	High byte of Wide Message Marker 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.
T2[31:24]	DB3[7:0]	Data Byte 3  Note 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.
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

27.3.9.4 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 27-13 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 27-14 Transmit Event FIFO Element Description

Bit	Name	Description
E0[31]	ESI	Error State Indicator ● 0b0: The sending node is in error active state ● 0b1: The sending node is in error passive state
E0[30]	XTD	Extended Identifier ● 0b0: 11-bit standard ID ● 0b1: 29-bit extended ID

Bit	Name	Description
E0[29]	RTR	Remote Transmission Request ● 0b0: The transmitted frame is a data frame. ● 0b1: The transmitted frame is a remote frame.
E0[28:0]	ID[28:0]	Identifier 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 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 ● 0b00: Reserved ● 0b01: Transmit Event ● 0b10: Transmit after cancellation (this value is always used when transmitting in DAR mode). ● 0b11: Reserved
E1A/B[21]	FDF	FD Format ● 0b0: Classic CAN frame format. ● 0b1: CAN FD frame format (new DLC encoding and CRC).
E1A/B[20]	BRS	Bit Rate Switch ● 0b0: No bitrate switching when sending CAN FD frame ● 0b1: Bitrate switching when sending CAN FD frame.
E1A/B[19:16]	DLC[3:0]	Data Length Code ● 0-8: Classic CAN and CAN FD: The transmitted frame contains 0 to 8 data bytes. ● 9-15: Classic CAN: The transmitted frame contains 8 data bytes. ● 9-15: CAN FD: The transmitted frame contains 12/16/20/24/32/48/64 data bytes.
E1A[15:0]	TXTS[15:0]	Tx Timestamp 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 Written by the CPU during the configuration of the transmit buffer. Copied to the send event FIFO element to identify the send message status.

27.3.9.5 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 27-15 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 27-16 Standard Message ID Filter Element Description

Bit	Name	Description
S0[31:30]	SFT[1:0]	Standard Filter Type ● 0b00: Range filter from SFID1 to SFID2 ($SFID2 \geq SFID1$) ● 0b01: Dual ID filter defined by SFID1 or SFID2 ● 0b10: Classic filter: $SFID1=ID$, $SFID2=mask$ ● 0b11: Disable this filter element  Note When SFT=0b11, this filter element is disabled and acceptance filtering continues (same behavior as when SFEC=0b000).
S0[29:27]	SFEC[2:0]	Standard Filter Element Configuration 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. ● 0b000: Disable this filter element ● 0b001: If matched, stored in receive FIFO0 ● 0b010: If a match is found, store it in receive FIFO1 ● 0b011: If match, reject ID ● 0b100: If match, set priority, do not store ● 0b101: If match, set priority and store in FIFO0 ● 0b110: If match, set priority and store in FIFO1 ● 0b111: Store in receive buffer or as debug message, ignore configuration of SFT[1:0]

Bit	Name	Description
S0[26:16]	SFID1[10:0]	Standard Filter ID 1 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 This bit field has different meanings, depending on the configuration of SFEC: ● When SFEC=0b001-0b110: The second ID of the standard ID filter element ● When SFEC=0b111: Filter for receive buffer or debug message
	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. ● 0b00: Store the message into the receive buffer ● 0b01: Debug message A ● 0b10: Debug message B ● 0b11: Debug message C
	SFID2[5:0]	Defines the offset (buffer index) from the receive buffer where the matching message is stored to the starting address RXBC.RBSA.

27.3.9.6 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 27-17 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 27-18 Extended Message ID Filter Element Description

Bit	Name	Description
F0[31:29]	EFEC[2:0]	Extended Filter Element Configuration 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]	● 0b000: Disable this filter element ● 0b001: If matched, stored in receive FIFO0 ● 0b010: If a match is found, store it in receive FIFO1 ● 0b011: If match, reject ID ● 0b100: If matched, set priority ● 0b101: If match, set priority and store into FIFO0 ● 0b110: If match, set priority and store in FIFO1 ● 0b111: Store in the receive buffer or as a debug message, ignoring the configuration of EFT[1:0] 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.
F0[28:0]	EFID1[28:0]	Extended Filter ID 1 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.
F1[31:30]	EFT[1:0]	Extended Filter Type ● 0b00: Range filter from EFID1 to EFID2 ($EFID2 \geq EFID1$) ● 0b01: Dual ID filter defined by EFID1 or EFID2 ● 0b10: Classic filter: $EFID1=ID$, $EFID2=mask$ ● 0b11: Range filter from EFID1 to EFID2 ($EFID2 \geq EFID1$), do not use MCAN_XIDAM mask
F1[28:0]	EFID2[28:0]	Extended Filter ID 2 This bit field has different meanings, depending on the configuration of EFEC: ● EFEC=0b001-0b110: The second ID of the extended ID filter element ● EFEC=0b111: Filter for receiving buffer or debug messages
	EFID2[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. ● 0b00: Store the message into the receive buffer ● 0b01: Debug message A ● 0b10: Debug message B ● 0b11: Debug message C
	EFID2[5:0]	Defines the offset (buffer index) of the receive buffer where matching messages are stored to the starting address MCAN_RXBC.RBSA.

27.4 Register Description

27.4.1 Register List

Base address: 0x40011400

Table 27-19 Controller Area Network (MCAN) Register Offset

Offset	Register	Description	Width
0x04	MCAN_ENDN	MCAN Endian Register	32
0x0C	MCAN_DBTP	MCAN Data Bit Timing 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 Timing 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 Select 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 Cancellation Request Register	32
0xD8	MCAN_TXBTO	MCAN Transmit Buffer Transmission Occurred Register	32
0xDC	MCAN_TXBCF	MCAN Transmit Buffer Cancellation Finished Register	32
0xE0	MCAN_TXBTIE	MCAN Transmit Buffer Transmission Interrupt Enable Register	32
0xE4	MCAN_TXBCIE	MCAN Transmit Buffer Cancellation Finished 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: 0x40011800

Table 27-20 MCANRAM register offset

Offset	Register	Description	Width
0x00	MCANRAM_CKCR	MCAN Message RAM Check Control Register	32
0x04	MCANRAM_CKSR	MCAN Message RAM Check Status Register	32
0x08	MCANRAM_EIEN	MCAN Message RAM Error Injection Enable Register	32
0x0C	MCANRAM_EIBIT0	MCAN Message RAM Error Injection Bit Enable Register 0	32
0x10	MCANRAM_EIBIT1	MCAN Message RAM Error Injection Bit Enable Register 1	32
0x14	MCANRAM_ECCERRADDR	MCAN Message RAM Check Error Address Register	32

27.4.2 MCAN Endian 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	Endian Test Value	The test value is 0x87654321	RO

27.4.3 MCAN Data Bit Timing 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). $tq = (DBRP + 1) \cdot mtq$.

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	0b0: Disable 0b1: Enable	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.

27.4.4 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	0b0: Value of TXBNS disabled 0b1: Value of TXBNS enabled	RO
20:16	TXBNS	The Buffer Number of the Buffer That Has Started Transmitting.	The buffer number of the last buffer that started transmitting. 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	0b0: Value of TXBNP disabled 0b1: Value of TXBNP enabled	RO
12:8	TXBNP	The Number of Buffers That Are Ready to Send	Transmitting buffer number ready to send 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 0b0: CAN bus is dominant (MCAN_RX=0) 0b1: CAN bus is recessive (MCAN_RX=1)	RO
6:5	TX	Transmit Pin Control	0b00: Reset value, MCAN_TX pin is controlled by the CAN core and updated at the end of the CAN bit time 0b01: The sampling point can be monitored on the MCAN_TX pin 0b10: MCAN_TX pin outputs a dominant level (0) 0b11: MCAN_TX pin outputs a recessive level (1)	RWA

Bit	Name	Function	Description	Access
4	LBCK	Loop mode	● 0b0: Reset value, disable loopback mode ● 0b1: Enable loopback mode (see "Loopback Mode" for details)	RWA
3:0	<i>Reserved</i>	<i>Reserved</i>	-	-

27.4.5 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. If the initial value 0x00 is used, the counter is disabled.	RWA

27.4.6 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	RO	RWA	RWA	RW	RO	RWA	RWA	RW

Bit	Name	Function	Description	Access
31:16	Reserved	Reserved	-	-
15	NISO	Non-ISO Mode Selection	0b0: Use the CAN FD frame format specified by ISO 11898-1:2015 0b1: Use the CAN FD frame format specified by Bosch CAN FD specification V1.0	RWA
14	TXP	Transmission Pause	0b0: Disable transmission pause 0b1: Enable transmission pause  Note 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.	RWA
13	EFBI	Edge Filtering during Bus Synchronization	0b0: Edge filtering disabled 0b1: Two consecutive dominant tq's are required to detect a hard sync edge	RWA
12	PXHD	Protocol Exception Handling Disabled	0b0: Enable protocol exception processing 0b1: Disable protocol exception processing  Note When protocol exception handling is disabled, MCAN will transmit error frames upon detection of a protocol exception condition.	RWA
11	WMM	Wide Message Marker	0b0: Use 8-bit message marker 0b1: Use 16-bit message marker, replace the 16-bit timestamp in the transmit event FIFO  Note When using 16-bit wide message marker (WMM=1), the 16-bit internal timestamp in the transmit event FIFO is disabled.	RWA
10	Reserved	Reserved	-	-

Bit	Name	Function	Description	Access
9	BRSE	Bit Rate Switching	0b0: Disable bit rate switching when sending 0b1: Enable bit rate switching when sending  Note When CAN FD mode is disabled (FDOE=0), the setting of BRSE has no effect.	RWA
8	FDOE	FD Mode Enable	0b0: Disable FD Mode 0b1: Enable FD Mode	RWA
7	TEST	Test Mode Enable	Write 1 to enable test mode, write 0 has no effect Read: 0b0: Normal mode, test register MCAN_TEST keeps reset value 0b1: Test mode, enable write access to test register MCAN_TEST	RWA
6	DAR	Disable Automatic Resend	0b0: Enable automatic resend of unsuccessfully sent messages 0b1: Disable automatic resend	RWA
5	MON	Bus Listening Mode	Write 1 to enable the bus monitoring mode, write 0 has no effect Read: 0b0: Disables bus monitoring mode 0b1: Enables bus monitoring mode  Note 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.	RWA
4	CSR	Clock Stop Request	0b0: No clock stop requested 0b1: Clock stop requested  Note 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.	RW
3	CSA	Clock Stop Acknowledgment	0b0: Clock stop request not acknowledged 0b1: Clock stop request acknowledged, MCAN can be set to Power-down state by stopping the control logic clock and communication clock.	RO

Bit	Name	Function	Description	Access
2	ASM	Restricted Working Mode	Write 1 to enable restricted working mode, write 0 has no effect Read: ● 0b0: Normal mode ● 0b1: Restricted working mode  Note 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. For a description of restricted working mode, see "Restricted Working Mode".	RWA
1	CCE	Configuration Change Enabled	● 0b0: CPU has no write access to protected configuration registers ● 0b1: CPU has write access to protected configuration registers (when INIT=1)	RWA
0	INIT	Initialization	● 0b0: Normal operation ● 0b1: Start initialization  Note 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.	RW

27.4.7 MCAN Nominal Bit Timing 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								RWA								RWA									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	Reserved	Reserved	-	-
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.

27.4.8 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	0b00: The timestamp counter value is always 0x0000 0b01: The timestamp counter value is incremented according to TCP 0b10: Setting prohibited 0b11: The timestamp counter value is always 0x0000 (equivalent to the set value "0b00")	RWA

27.4.9 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	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.  Note ● "Wraparound" means that the value of the timestamp counter changes from non-zero to zero, not due to a write operation of MCAN_TSCV. ● When accessing bytes, writing any byte of this register will cause the timestamp register to be cleared.	RWC

27.4.10 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: 0b00: Continuous mode 0b01: Timeout controlled by the transmit event FIFO 0b10: Timeout controlled by the receive FIFO0 0b11: Timeout controlled by the receive FIFO1	RWA
0	ETOC	Timeout Counter Enabled	0b0: Disable 0b1: Enable	RWA

27.4.11 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	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.  Note 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.	RW

27.4.12 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	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. 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.  Note When accessing by byte, reading byte 2 of the register will clear CEL, and reading byte 3, 1 or 0 has no effect.	RC
15	RP	Receive Error Passive	0b0: Receive error counter is below error passive level 128 0b1: Receive error counter has reached error passive level 128	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.

27.4.13 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																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																							
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	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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. 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	0b0: No protocol exception event has occurred since the last read access 0b1: A protocol exception event has occurred  Note During byte access, reading byte 0 of the register will clear this bit, reading bytes 3/2/1 has no effect.	RO
13	RFDF	Received CAN FD Message	This bit is set regardless of receive filtering. 0b0: No CAN FD formatted message has been received since the last time it was cleared by the CPU 0b1: Received CAN FD formatted message with FDF flag set to 1  Note During byte access, reading byte 0 of the register will clear this bit, reading bytes 3/2/1 has no effect.	RO

Bit	Name	Function	Description	Access
12	RBRS	BRS Flag of the Last Received CAN FD Message	This bit is set together with RFDF and is independent of receive filtering. 0b0: BRS flag of the last received CAN FD message is not set 0b1: BRS flag of the last received CAN FD message is set  Note During byte access, reading byte 0 of the register will clear this bit, reading bytes 3/2/1 has no effect.	RO
11	RESI	ESI Flag of the Last Received CAN FD Message	This bit is set in conjunction with RFDF, regardless of the acceptance filter. 0b0: ESI flag of the last received CAN FD message is not set 0b1: ESI flag of the last received CAN FD message is set  Note During byte access, reading byte 0 of the register will clear this bit, reading bytes 3/2/1 has no effect.	RO
10:8	DLEC	Data Phase Last Error Code	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.  Note For byte access, reading byte 0 of the register will set this bit to 0b111, reading bytes 3/2/1 has no effect.	RO
7	BO	Bus_Off State	0b0: MCAN is not in Bus_Off state 0b1: MCAN is in Bus_Off state	RO
6	EW	Warning Status	0b0: Both error counters are less than the error warning (Error_Warning) upper limit of 96 0b1: At least one error counter has reached the error warning (Error_Warning) upper limit of 96	RO
5	EP	Error Passive	0b0: MCAN is in Error_Active state. MCAN participates in bus communication normally and sends active error flag when an error is detected 0b1: MCAN is in Error_Passive state	RO

Bit	Name	Function	Description	Access
4:3	ACT	Communication Status	Monitor the communication status of MCAN: ● 0b00: Synchronizing, the node is synchronizing CAN communication ● 0b01: Idle, the node is neither a receiving node nor a sending node ● 0b10: Receiving, the node works as a receiving node ● 0b11: Sending, the node works as a sending node  Note Protocol abnormal events will set this bit to 0b00.	RO
2:0	LEC	Last Error Code	● 0b000: No Error. No error has occurred since LEC was reset by successful reception or transmission ● 0b001: Stuff Error. More than 5 identical bits appeared in a part of the received message, which is not disabled ● 0b010: Form Error. The fixed format part of the received frame has a format error ● 0b011: ACK Error. The message sent by MCAN is not acknowledged by other nodes ● 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) ● 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) ● 0b110: CRC Error. The CRC of the received message is incorrect. The received CRC does not match the CRC calculated using the received data ● 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.	RS

Bit	Name	Function	Description	Access
2:0	LEC	Last Error Code	 Note - 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. - 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. - For byte operations, reading byte 0 of the register will set this bit to 0b111, and reading bytes 3/2/1 has no effect.	RS

27.4.14 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

27.4.15 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	0b0: No access to reserved address 0b1: Access to reserved address	RW
28	PED	Data Phase Protocol Error (Using Data Bit Time)	0b0: No data phase protocol error detected 0b1: Data phase protocol error detected (MCAN_PSR.DLEC≠0b000/0b111)	RW
27	PEA	Arbitration Phase Protocol Error (Using Nominal Bit Time)	0b0: No arbitration phase protocol error detected 0b1: Arbitration phase protocol error detected (MCAN_PSR.LEC≠0b000/0b111)	RW
26	WDI	Watchdog Interrupt	0b0: No message RAM watchdog event occurred 0b1: A message RAM watchdog event occurred due to the loss of the READY signal	RW
25	BO	Bus_Off State	0b0: Bus_Off status has not changed 0b1: Bus_Off status has changed	RW
24	EW	Warning State	0b0: Error_Warning status has not changed 0b1: Error_Warning status has changed	RW
23	EP	Error Passive	0b0: Error_Passive state has not changed 0b1: Error_Passive state has changed	RW
22	ELO	Error Record Overflow	0b0: CAN error record counter has not overflowed 0b1: CAN error record counter has overflowed	RW

Bit	Name	Function	Description	Access
21	BEU	Bit Error Uncorrected	0b0: No bit error detected when reading from the message RAM 0b1: Bit error detected and uncorrected when reading from the message RAM when MCANRAM_CKCR.ECCMOD is not 0  Note BEU is set while MCAN_CCCR.INIT is set.	RW
20	BEC	Bit Error Corrected	0b0: No bit error detected when reading from the message RAM 0b1: When MCANRAM_CKCR.ECCMOD is 0b10 or 0b11, a bit error was detected and corrected when reading from the message RAM	RW
19	DRX	Message Stored in Dedicated Receive Buffer	0b0: Receive buffer not updated 0b1: At least one message has been stored in receive buffer  Note This flag is set whenever a received message has been stored in dedicated receive buffer.	RW
18	TOO	Timeout Occurred	0b0: No timeout occurred 0b1: Timeout occurred	RW

Bit	Name	Function	Description	Access
17	MRAF	Message RAM Access Failure	● 0b0: No message RAM access failure occurred ● 0b1: Message RAM access failure occurred Set during receive processing: ● 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. ● Failed to write a message to the message RAM, message storage is terminated.  Note 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. Set condition in the send processing: ● 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 "Restricted Working Mode" for details). To exit restricted operating mode, the host CPU must reset MCAN_CCCR.ASM.	RW
16	TSW	Timestamp Wrap	● 0b0: Timestamp counter has not wrapped ● 0b1: Timestamp counter has wrapped	RW
15	TEFL	Transmit Event FIFO Element Lost	● 0b0: Transmit event FIFO element has not been lost ● 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	RW
14	TEFF	Transmit Event FIFO is Full	● 0b0: Transmit event FIFO is not full ● 0b1: Transmit event FIFO is full	RW
13	TEFW	Transmit Event FIFO Reaches Watermark	● 0b0: Transmit event FIFO fill level is below watermark ● 0b1: Transmit event FIFO fill level reaches watermark	RW
12	TEFN	Transmit Event FIFO New Entry	● 0b0: Transmit event FIFO unchanged ● 0b1: Transmit processing unit writes to transmit event FIFO element	RW
11	TFE	Transmit FIFO Empty	● 0b0: Transmit FIFO is not empty ● 0b1: Transmit FIFO is empty	RW

Bit	Name	Function	Description	Access
10	TCF	Send Cancel Completed	● 0b0: Send cancel not completed ● 0b1: Send cancel completed	RW
9	TC	Sending Completed	● 0b0: Sending not completed ● 0b1: Sending completed	RW
8	HPM	High Priority Message	● 0b0: High priority message not received ● 0b1: High priority message received	RW
7	RF1L	Receive FIFO1 Message Lost	● 0b0: Receive FIFO1 message not lost ● 0b1: Receive FIFO1 message lost; this bit is also set when writing to receive FIFO1 with space size 0	RW
6	RF1F	Receive FIFO1 is Full	● 0b0: Receive FIFO1 not full ● 0b1: Receive FIFO1 full	RW
5	RF1W	Receive FIFO1 Reaches Watermark	● 0b0: Receive FIFO1 fill level is below watermark ● 0b1: Receive FIFO1 fill level reaches watermark	RW
4	RF1N	Receive New Message in FIFO1	● 0b0: No new message is written into FIFO1 ● 0b1: New message is written into FIFO1	RW
3	RF0L	Receive FIFO0 Message Lost	● 0b0: Receive FIFO0 message not lost ● 0b1: Receive FIFO0 message lost; this bit is also set when writing to receive FIFO0 with space size 0	RW
2	RF0F	Receive FIFO0 is Full	● 0b0: Receive FIFO0 not full ● 0b1: Receive FIFO0 full	RW
1	RF0W	Receive FIFO0 Reaches Watermark	● 0b0: Receive FIFO0 fill level is below watermark ● 0b1: Receive FIFO0 fill level reaches watermark	RW
0	RF0N	Receive New Message in FIFO0	● 0b0: No new message is written into FIFO0 ● 0b1: New message is written into FIFO0	RW

27.4.16 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	RW

Bit	Name	Function	Description	Access
31:30	Reserved	Reserved	-	-
29	ARAE	Access Re-served Address Interrupt Enabled	0b0: Disable interrupt 0b1: Enable interrupt	RW
28	PEDE	Data Phase Protocol Error Interrupt Enabled	0b0: Disable interrupt 0b1: Enable interrupt	RW
27	PEAE	Arbitration Phase Protocol Error Interrupt Enabled	0b0: Disable interrupt 0b1: Enable interrupt	RW
26	WDIE	Watchdog Interrupt Enable	0b0: Disable interrupt 0b1: Enable interrupt	RW
25	BOE	Bus_Off Status Interrupt Enabled	0b0: Disable interrupt 0b1: Enable interrupt	RW
24	EWE	Warning Status Interrupt nabled	0b0: Disable interrupt 0b1: Enable interrupt	RW
23	EPE	Error Passive Interrupt Enabled	0b0: Disable interrupt 0b1: Enable interrupt	RW
22	ELOE	Error Record Overflow Interrupt Enabled	0b0: Disable interrupt 0b1: Enable interrupt	RW
21	BEUE	Bit Error Uncorrected Interrupt Enabled	0b0: Disable interrupt 0b1: Enable interrupt	RW

Bit	Name	Function	Description	Access
20	BECE	Bit Error Corrected Interrupt Enabled	● 0b0: Disable interrupt ● 0b1: Enable interrupt	RW
19	DRXE	Message Storage to Dedicated Receive Buffer Interrupt Enabled	● 0b0: Disable interrupt ● 0b1: Enable interrupt	RW
18	TOOE	Timeout Interrupt Enabled	● 0b0: Disable interrupt ● 0b1: Enable interrupt	RW
17	MRAFE	Message RAM Access Failure Interrupt Enabled	● 0b0: Disable interrupt ● 0b1: Enable interrupt	RW
16	TSWE	Timestamp Wrap Interrupt Enabled	● 0b0: Disable interrupt ● 0b1: Enable interrupt	RW
15	TEFLE	Transmit Event FIFO Element Lost Interrupt Enabled	● 0b0: Disable interrupt ● 0b1: Enable interrupt	RW
14	TEFFE	Transmit Event FIFO Full Interrupt Enabled	● 0b0: Disable interrupt ● 0b1: Enable interrupt	RW
13	TEFWE	Transmit Event FIFO Reaches Water Level Interrupt Enabled	● 0b0: Disable interrupt ● 0b1: Enable interrupt	RW
12	TEFNE	Transmit Event FIFO New Entry Interrupt Enabled	● 0b0: Disable interrupt ● 0b1: Enable interrupt	RW
11	TFEE	Transmit FIFO Empty Interrupt Enabled	● 0b0: Disable interrupt ● 0b1: Enable interrupt	RW
10	TCFE	Transmit Cancel Completed Interrupt Enabled	● 0b0: Disable interrupt ● 0b1: Enable interrupt	RW
9	TCE	Transmit Complete Interrupt Enabled	● 0b0: Disable interrupt ● 0b1: Enable interrupt	RW
8	HPME	High Priority Message Interrupt Enabled	● 0b0: Disable interrupt ● 0b1: Enable interrupt	RW

Bit	Name	Function	Description	Access
7	RF1LE	Receive FIFO1 Message Lost Interrupt Enabled	● 0b0: Disable interrupt ● 0b1: Enable interrupt	RW
6	RF1FE	Receive FIFO1 Full Interrupt Enabled	● 0b0: Disable interrupt ● 0b1: Enable interrupt	RW
5	RF1WE	Receive FIFO1 Reaches Water Level Interrupt Enabled	● 0b0: Disable interrupt ● 0b1: Enable interrupt	RW
4	RF1NE	Receive FIFO1 New Message Interrupt Enabled	● 0b0: Disable interrupt ● 0b1: Enable interrupt	RW
3	RF0LE	Receive FIFO0 Message Lost Interrupt Enabled	● 0b0: Disable interrupt ● 0b1: Enable interrupt	RW
2	RF0FE	Receive FIFO0 Full Interrupt Enabled	● 0b0: Disable interrupt ● 0b1: Enable interrupt	RW
1	RF0WE	Receive FIFO0 Reaches Water Level Interrupt Enabled	● 0b0: Disable interrupt ● 0b1: Enable interrupt	RW
0	RF0NE	Receive FIFO0 New Message Interrupt Enabled	● 0b0: Disable interrupt ● 0b1: Enable interrupt	RW

27.4.17 MCAN Interrupt Line Select 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	TEFL	TEFL	TEFWL	TEFNL	TEFEL	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	RW

Bit	Name	Function	Description	Access
31:30	Reserved	Reserved	-	-
29	ARAL	Access Re-served Address Interrupt Line	0b0: Select interrupt line 0 0b1: Select interrupt line 1	RW
28	PEDL	Data Phase Protocol Error Interrupt Line	0b0: Select interrupt line 0 0b1: Select interrupt line 1	RW
27	PEAL	Arbitration Phase Protocol Error Interrupt Line	0b0: Select interrupt line 0 0b1: Select interrupt line 1	RW
26	WDIL	Watchdog In-terrupt Line	0b0: Select interrupt line 0 0b1: Select interrupt line 1	RW
25	BOL	Bus_Off State Interrupt Line	0b0: Select interrupt line 0 0b1: Select interrupt line 1	RW
24	EWL	Warning State Interrupt Line	0b0: Select interrupt line 0 0b1: Select interrupt line 1	RW
23	EPL	Error Passive Interrupt Line	0b0: Select interrupt line 0 0b1: Select interrupt line 1	RW
22	ELOL	Error Record Overflow Inter-rupt Line	0b0: Select interrupt line 0 0b1: Select interrupt line 1	RW
19	DRXL	Messages are Stored in a Dedicated Re-ceive Buffer In-terrupt Line	0b0: Select interrupt line 0 0b1: Select interrupt line 1	RW
18	TOOL	Timeout Inter-rupt Line Oc-curred.	0b0: Select interrupt line 0 0b1: Select interrupt line 1	RW

Bit	Name	Function	Description	Access
17	MRAFL	Message RAM Access Failure Interrupt Line	● 0b0: Select interrupt line 0 ● 0b1: Select interrupt line 1	RW
16	TSWL	Timestamp wrap-around interrupt line	● 0b0: Select interrupt line 0 ● 0b1: Select interrupt line 1	RW
15	TEFLL	Transmission Event FIFO Element Lost Interrupt Line	● 0b0: Select interrupt line 0 ● 0b1: Select interrupt line 1	RW
14	TEFFL	Transmission Event FIFO Full Interrupt Line	● 0b0: Select interrupt line 0 ● 0b1: Select interrupt line 1	RW
13	TEFWL	Transmission Event FIFO Watermark Reached Interrupt Line	● 0b0: Select interrupt line 0 ● 0b1: Select interrupt line 1	RW
12	TEFNL	Transmission Event FIFO New Entry Interrupt Line	● 0b0: Select interrupt line 0 ● 0b1: Select interrupt line 1	RW
11	TFEL	Transmission FIFO Empty Interrupt Line	● 0b0: Select interrupt line 0 ● 0b1: Select interrupt line 1	RW
10	TCFL	Transmission Cancellation Finished Interrupt Line	● 0b0: Select interrupt line 0 ● 0b1: Select interrupt line 1	RW
9	TCL	Transmission Complete Interrupt Line	● 0b0: Select interrupt line 0 ● 0b1: Select interrupt line 1	RW
8	HPML	High Priority Message Interrupt Line	● 0b0: Select interrupt line 0 ● 0b1: Select interrupt line 1	RW
7	RF1LL	Receive FIFO1 Message Lost Interrupt Line	● 0b0: Select interrupt line 0 ● 0b1: Select interrupt line 1	RW
6	RF1FL	Receive FIFO1 Full Interrupt Line	● 0b0: Select interrupt line 0 ● 0b1: Select interrupt line 1	RW
5	RF1WL	Receive FIFO1 Watermark Interrupt Line	● 0b0: Select interrupt line 0 ● 0b1: Select interrupt line 1	RW
4	RF1NL	Receive FIFO1 New Message Interrupt Line	● 0b0: Select interrupt line 0 ● 0b1: Select interrupt line 1	RW

Bit	Name	Function	Description	Access
3	RF0LL	Receive FIFO0 Message Lost Interrupt Line	● 0b0: Select interrupt line 0 ● 0b1: Select interrupt line 1	RW
2	RF0FL	Receive FIFO0 Full Interrupt Line	● 0b0: Select interrupt line 0 ● 0b1: Select interrupt line 1	RW
1	RF0WL	Receive FIFO0 Watermark Interrupt Line	● 0b0: Select interrupt line 0 ● 0b1: Select interrupt line 1	RW
0	RF0NL	Receive FIFO0 New Message Interrupt Line	● 0b0: Select interrupt line 0 ● 0b1: Select interrupt line 1	RW

27.4.18 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	● 0b0: Disable ● 0b1: Enable	RW
0	EINT0	Interrupt Line 0 Enabled	● 0b0: Disable ● 0b1: Enable	RW

27.4.19 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: 0b00: Accept non-matching standard frames in Receive FIFO0 0b01: Accept non-matching standard frames in Receive FIFO1 0b10: Reject non-matching standard frames 0b11: Reject non-matching standard frames	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: 0b00: Accept non-matching extended frames in Receive FIFO0 0b01: Accept non-matching extended frames in Receive FIFO1 0b10: Reject non-matching extended frames 0b11: Reject non-matching extended frames	RWA
1	RRFS	Reject Remote Standard Frames	0b0: Filter remote frames with 11-bit standard ID 0b1: Reject all remote frames with 11-bit standard ID	RWA
0	RRFE	Reject Remote Extension Frames	0b0: Filter remote frames with 29-bit extension ID 0b1: Reject all remote frames with 29-bit extension ID	RWA

27.4.20 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	0: No standard ID filter 1-128: Number of standard ID filter elements >128: Setting values greater than 128 are interpreted as 128	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 " Message RAM Configuration ")	RWA
1:0	Reserved	Reserved	-	-

27.4.21 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								LSE								FLESA[15:2]								Reserved							
Access									RWA								RWA															

Bit	Name	Function	Description	Access
31:23	Reserved	Reserved	-	-
22:16	LSE	Standard ID Filter List Size	0: No extended ID filter 1-64: Number of extended ID filter elements >64: Setting values greater than 64 are interpreted as 64	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 " Message RAM Configuration ")	RWA
1:0	Reserved	Reserved	-	-

27.4.22 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

27.4.23 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							BIDX																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																											
Access																	FLST																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																			

Bit	Name	Function	Description	Access
31:16	Reserved	Reserved	-	-
15	FLST	Filter List	Indicates the matching filter list: 0b0: Standard filter list 0b1: Extended filter list	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	0b00: FIFO not selected 0b01: FIFO message lost 0b10: Message stored in FIFO0 0b11: Message stored in FIFO1	RO
5:0	BIDX	Buffer Index	The index of the receive FIFO element storing the message. Only active when MSI[1]=1	RO

27.4.24 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	Corresponding to the new data flag of the receive buffer 31-0: ● 0b0: The receive buffer is not updated ● 0b1: The receive buffer is updated by a new message 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.  Note Each bit corresponds to a buffer.	RW

27.4.25 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	Corresponding to the new data flag of the receive buffer 63-32: ● 0b0: The receive buffer is not updated ● 0b1: The receive buffer is updated by a new message 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.  Note Each bit corresponds to a buffer.	RW

27.4.26 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	0b0: Receive FIFO0 is in blocking mode 0b1: Receive FIFO0 is in overwriting mode	RWA
30:24	F0WM	Receive FIFO0 Water Level	0: Disable receive FIFO0 water level interrupt 1-64: Level of receive FIFO0 waterline interrupt (MCAN_IR.RF0W) >64: Disable receive FIFO0 water level interrupt	RWA
23	Reserved	Reserved	-	-
22:16	F0S	Receive FIFO0 Size	0: No receive FIFO0 1-64: Number of receive FIFO0 elements >64: Setting values greater than 64 are interpreted as 64 The receive FIFO0 elements are indexed from 0 to F0 S minus 1.	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 " Message RAM Configuration ")	RWA
1:0	Reserved	Reserved	-	-

27.4.27 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							RF0L		F0F		Reserved	F0PI					Reserved	F0GI					Reserved	F0FL							
Access							RO		RO			RO						RO						RO								

Bit	Name	Function	Description	Access
31:26	Reserved	Reserved	-	-
25	RF0L	Receive FIFO0 Message Lost	This bit is a copy of the interrupt flag MCAN_IR.RF0L. When MCAN_IR.RF0L is reset, this bit is also reset: 0b0: No message lost in the receive FIFO0 0b1: Message lost in the receive FIFO0; this bit is also set when writing to a receive FIFO0 with a space size of 0  Note When MCAN_RXF0C.F0OM=1, the oldest message is overwritten and this flag is not set.	RO
24	F0F	Receive FIFO0 is Not Full	0b0: Receive FIFO0 is not full 0b1: Receive FIFO0 is full	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

27.4.28 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

27.4.29 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	-	-

27.4.30 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	0b0: Receive FIFO1 is in blocking mode 0b1: Receive FIFO1 is in overwriting mode	RWA
30:24	F1WM	Receive FIFO1 Water Level	0: Disable receive FIFO1 water level interrupt 1-64: Level of receive FIFO1 waterline interrupt (MCAN_IR.RF1W) >64: Disable receive FIFO1 water level interrupt	RWA
23	Reserved	Reserved	-	-
22:16	F1S	Receive FIFO1 Size	0: No receive FIFO1 1-64: Number of receive FIFO1 elements >64: Setting values greater than 64 are interpreted as 64 The receive FIFO1 elements are indexed from 0 to F1S minus 1.	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 " Message RAM Configuration ")	RWA
1:0	Reserved	Reserved	-	-

27.4.31 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		Reserved	F1PI				Reserved	F1GI				Reserved	F1FL																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																										

Bit	Name	Function	Description	Access
31:30	DMS	Debug Message Status	0b00: Idle state, waiting to receive debug message 0b01: Received debug message A 0b10: Received debug messages A and B 0b11: Received debug messages A, B, and C	RO
29:26	Reserved	Reserved	-	-
25	RF1L	Receive FIFO1 Message Lost	This bit is a copy of the interrupt flag MCAN_IR.RF1L. When MCAN_IR.RF1L is reset, this bit is also reset: 0b0: No message lost in the receive FIFO1 0b1: Message lost in the receive FIFO1; this bit is also set when writing to a receive FIFO1 with a space size of 0  Note When MCAN_RXF1C.F1OM=1, the oldest message is overwritten and this flag is not set.	RO
24	F1F	Receive FIFO1 is Full	0b0: Receive FIFO1 is not full 0b1: Receive FIFO1 is full	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

27.4.32 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

27.4.33 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	0b000: 8 bytes 0b001: 12 bytes 0b010: 16 bytes 0b011: 20 bytes 0b100: 24 bytes 0b101: 32 bytes 0b110: 48 bytes 0b111: 64 bytes	RWA
7	Reserved	Reserved	-	-
6:4	F1DS	Receive FIFO1 Data Field Size	0b000: 8 bytes 0b001: 12 bytes 0b010: 16 bytes 0b011: 20 bytes 0b100: 24 bytes 0b101: 32 bytes 0b110: 48 bytes 0b111: 64 bytes	RWA
3	Reserved	Reserved	-	-
2:0	F0DS	Receive FIFO0 Data Field Size	0b000: 8 bytes 0b001: 12 bytes 0b010: 16 bytes 0b011: 20 bytes 0b100: 24 bytes 0b101: 32 bytes 0b110: 48 bytes 0b111: 64 bytes	RWA

27.4.34 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	0b0: Transmit FIFO mode 0b1: Transmit queue mode	RWA
29:24	TFQS	Transmit FIFO/Queue Size	0: No transmit FIFO or queue 1-32: Transmit FIFO queue buffer size >32: Setting values greater than 32 are interpreted as 32	RWA
23:22	Reserved	Reserved	-	-
21:16	NDTB	Dedicated Transmit Buffer Size	0: No dedicated transmit buffer 1-32: Dedicated transmit buffer size >32: Setting values greater than 32 are interpreted as 32	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 "Message RAM Configuration")	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.

27.4.35 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	0b0: Transmit FIFO or queue not full 0b1: Transmit FIFO or queue full	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.

27.4.36 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	0b000: 8 bytes 0b001: 12 bytes 0b010: 16 bytes 0b011: 20 bytes 0b100: 24 bytes 0b101: 32 bytes 0b110: 48 bytes 0b111: 64 bytes	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).

27.4.37 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 (n=31-0)	Transmit request pending flag Corresponding to the transmit request pending bits of transmit buffers 31-0: 0b0: No transmit request pending 0b1: Transmit request pending 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. The TXBRP bit corresponding to the transmit buffer configured by MCAN_TXBC is set. When the TXBRP bit is set, a transmit scan (see "Transmission Processing") is started to check the highest priority pending transmit request (i.e. the transmit buffer with the smallest message ID). 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. After requesting cancellation, the following situations indicate cancellation completion through register MCAN_TXBCF: - After successful transmission and corresponding MCAN_TXBTO bit is set - Transmission has not started at the time of cancellation - Transmission is aborted due to arbitration failure - Error occurs during frame transmission In DAR mode, all transmissions are automatically canceled after failure. For all failed transmissions, the corresponding MCAN_TXBCF bit is set.  Note Each bit corresponds to a buffer. 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.	RO

27.4.38 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 (n=31-0)	Add Request	Corresponding to the add request bit of the transmit buffer 31-0: Write 1 to set the corresponding add request bit, write 0 has no effect Read: ● 0b0: No transmit request added ● 0b1: Transmit request added 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.  Note Each bit corresponds to a buffer. 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.	RW

27.4.39 MCAN Transmit Buffer Cancellation 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

Bit	Name	Function	Description	Access
31:0	CRn (n=31-0)	Cancel Request	Corresponding to the cancel request bit of the transmit buffer 31-0: Write 1 to set the corresponding cancel request bit, write 0 has no effect Read: 0b0: No cancel request pending 0b1: Cancel request pending 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.  Note Each bit corresponds to a buffer.	RW

27.4.40 MCAN Transmit Buffer Transmission 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 (n=31-0)	Transmit Has Occurred	The corresponding transmit buffer 31-0 has transmit flag bit: ● 0b0: Not transmit ● 0b1: Transmit 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.  Note Each bit corresponds to a buffer.	RO

27.4.41 MCAN Transmit Buffer Cancellation Finished 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 (n=31-0)	Cancellation Completed	Corresponding to the cancelation completed flag of the transmit buffer 31-0: 0b0: No transmit buffer cancellation 0b1: Transmit buffer canceled When the cancellation requested by MCAN_TXBCR is completed and the corresponding bit of MCAN_TXBRP is cleared, the corresponding cancelation completed flag is set. If the corresponding bit of MCAN_TXBRP is not set at the time of cancellation, the cancelation completed flag is set immediately. When writing 1 to the corresponding bit of MCAN_TXBAR to request a new transmission, the corresponding cancelation completed flag is reset.  Note Each bit corresponds to a buffer.	RO

27.4.42 MCAN Transmit Buffer Transmission 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 _n (n=31-0)	Transmit Interrupt Enabled	Corresponding to the transmit interrupt enable bit of the transmit buffer 31-0: 0b0: Disable transmit interrupt 0b1: Enable transmit interrupt  Note Each bit corresponds to a buffer.	RW

27.4.43 MCAN Transmit Buffer Cancellation Finished 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 _n (n=31-0)	Cancel the Completed Interrupt Enabled	Corresponding to the cancel completed interrupt enable bit of the transmit buffer 31-0: 0b0: Disable cancellation of completed interrupts 0b1: Enable cancellation of completed interrupts  Note Each bit corresponds to a buffer.	RW

27.4.44 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	0: Disable water level interrupt 1-32: Level of transmit event FIFO water level interrupt (MCAN_IR.TEFW) >32: Disable water level interrupt	RWA
23:22	Reserved	Reserved	-	-
21:16	EFS	Transmit Event FIFO Size	0: Disable transmit event FIFO 1-32: Number of elements in transmit event FIFO >32: Setting values greater than 32 are interpreted as 32 The index of the transmit event FIFO element is 0 to EFS minus 1	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 " Message RAM Configuration ")	RWA
1:0	Reserved	Reserved	-	-

27.4.45 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								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: 0b0: No element is lost in the transmit event FIFO 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	RO
24	EFF	Transmission Event FIFO is Full	0: Transmission event FIFO is not full 1: Transmission event FIFO is full	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

27.4.46 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

27.4.47 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	● 0b00: Disable ECC check function. ● 0b01: ▶ If a single-bit error occurs, ECC corrects it without generating a single-bit error flag or interrupt. ▶ If a double-bit error occurs, ECC detects the error, sets the double-bit error flag, and generates an interrupt. ● 0b10: ▶ If a single-bit error occurs, ECC corrects the error, sets the single-bit error flag, and no interrupt is generated. ▶ If a double-bit error occurs, ECC detects the error, sets the double-bit error flag, and generates an interrupt. ● 0b11: ▶ If a single-bit error occurs, ECC corrects the error, sets the single-bit error flag, and generates an interrupt. ▶ If a double-bit error occurs, ECC detects the error, sets the double-bit error flag, and generates an interrupt.	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 MCAN generates a 1-bit error when accessing the message RAM, MCAN_IR.BEC of MCAN is set, and if the corresponding interrupt is enabled, MCAN generates an interrupt.
3. When ECCMOD is 0b01, 0b10 or 0b11, if MCAN generates a 2-bit error when accessing the message RAM, MCAN_IR.BEU of MCAN is set, and if the corresponding interrupt is enabled, MCAN generates an interrupt.

27.4.48 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 Read: 0b0: No 2-bit ECC error occurred. 0b1: A 2-bit ECC error occurred.	RW
0	RAM_1ERR	Message RAM ECC 1-bit Error Flag	Write 1 to clear the flag, write 0 has no effect Read: 0b0: No 1-bit ECC error occurred. 0b1: A 1-bit ECC error occurred.	RW

27.4.49 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	0b0: Disable 0b1: Enable	RW

27.4.50 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: 0b0: No inversion, no ECC error injection for this bit. 0b1: Invert, inject ECC error for this bit.  Note Each bit corresponds to an output data bit.	RW

27.4.51 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: 0b0: No inversion, no ECC error injection for this bit. 0b1: Invert, inject ECC error for this bit.  Note Each bit corresponds to an output data bit.	RW

27.4.52 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
31:11	Reserved	Reserved	-	-
10:0	ECCERRADDR	Check Error Address	Records the access address when a 1-bit or 2-bit ECC error occurs in the message RAM (the offset relative to the message RAM base address)	RO

27.5 Precautions

27.5.1 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

27.5.2 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.

28 CRC Operation (CRC)

28.1 Introduction

CRC calculation unit takes data stream or block as input and generates output number controlled by generator polynomial. This output number is typically used to verify correctness and integrity of data transmission or storage. This module supports CRC value calculation and verification.

28.2 Features

- One implementation standard: ISO/IEC13239
- Two encoding methods: CRC-16, CRC-32
- Three write bit widths: 8-bit, 16-bit, 32-bit
- Two working modes: CRC encoding mode, CRC check mode
- CRC-16 polynomial: $x^{16} + x^{12} + x^5 + 1$
- CRC-32 polynomial: $x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$

28.3 Functional Description

28.3.1 Working Mode

This module supports two working modes: CRC encoding mode and CRC check mode.

CRC encoding mode inputs a certain amount of original data to CRC module and obtains output value (CRC_RESULT). CRC check mode inputs a certain amount of encoded data sequence (original data + CRC check value) to verify match between original data and CRC check value (CRC_CR.FLAG).

28.3.2 Encoding Method

This module supports two encoding methods CRC-16 and CRC-32, with 16-bit and 32-bit results respectively, CRC_CR.CR configurable as needed.

CRC-16 Encoding

In this mode, the CRC algorithm model is CRC-16-X25:

- Polynomial: $x^{16} + x^{12} + x^5 + 1$;
- Initial value: 0xFFFFu;
- Result XOR value: 0xFFFFu;
- Input data inversion: True;
- Output data inversion: True.

CRC-32 Encoding

In this mode, the CRC algorithm model is CRC-32;

- Polynomial: $x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$;
- Initial value: 0xFFFFFFFFu;
- Result XOR value: 0xFFFFFFFFu;
- Input data inversion: True;
- Output data inversion: True.

28.3.3 Write Bit Width

This module supports three write bit widths: 8-bit, 16-bit, 32-bit. Different bit width writes must follow "bit width consistency, low-to-high" principle: "each write must use register matching data width, with lower bits written before higher bits".

Below demonstrates same data sequence written in three bit widths with identical output results.

- 8-bit width write: 0x00, 0x11, 0x22, 0x33, 0x44, 0x55, 0x66, 0x77
- 16-bit width write: 0x1100, 0x3322, 0x5544, 0x7766
- 32-bit width write: 0x33221100, 0x77665544

28.4 Operation Example

28.4.1 CRC-16 Encoding Mode

Procedure

- Step1.** Write CRC_CR.CR as 0b0 to select CRC-16.
- Step2.** Write CRC_RESLT[15:0] as 0xFFFF to initialize CRC calculation.
- Step3.** Write the raw data to be encoded sequentially to the CRC_DAT register, with selectable write bit width of 8-bit, 16-bit, or 32-bit.
- Step4.** Read CRC_RESLT[15:0] to obtain the CRC-16 calculation result.

28.4.2 CRC-16 Check Mode

Procedure

- Step1.** Write CRC_CR.CR as 0b0 to select CRC-16.
- Step2.** Write CRC_RESLT [15:0] as 0xFFFF to initialize CRC calculation.
- Step3.** Write the encoded data sequence (original data + CRC check value) into CRC_DAT register in sequence, with selectable write bit width of 8-bit, 16-bit, or 32-bit.
- Step4.** Determine whether the encoded data sequence has been tampered based on the value of CRC_CR.FLAG.

28.4.3 CRC-32 Encoding Mode

Procedure

- Step1.** Write CRC_CR.CR as 0b1 to select CRC-32.
- Step2.** Write CRC_RESLT[31:0] as 0xFFFFFFFF to initialize CRC calculation.
- Step3.** Write the raw data to be encoded sequentially to the CRC_DAT register, with selectable write bit width of 8-bit, 16-bit, or 32-bit.
- Step4.** Read CRC_RESLT[31:0] to obtain the CRC-32 calculation result.

28.4.4 CRC-32 Check Mode

Procedure

- Step1.** Write CRC_CR.CR as 0b1 to select CRC-32.
- Step2.** Write CRC_RESLT[31:0] as 0xFFFFFFFF to initialize CRC calculation.
- Step3.** Write the encoded data sequence (original data + CRC check value) into CRC_DAT register in sequence, with selectable write bit width of 8-bit, 16-bit, or 32-bit.
- Step4.** Determine whether the encoded data sequence has been tampered based on the value of CRC_CR.FLAG.

28.5 Register Description

28.5.1 Register List

Base address: 0x40015400

Table 28-1 CRC Operation (CRC) Register Offset

Offset	Register	Description	Width
0x00	CRC_CR	Control Register	32
0x04	CRC_RESLT	Result Register	32
0x80-0xFF	CRC_DAT	Data Register	32

28.5.2 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																												FLAG		RW	
Access																													CR			

Bit	Name	Function	Description	Access
31:2	Reserved	Reserved	-	-
1	FLAG	CRC Check Result	0b0: Current CRC check error 0b1: Current CRC check correct	RO
0	CR	CRC Encoding Method Selection	0b0: CRC-16 code 0b1: CRC-32 code	RW

28.5.3 Result Register (CRC_RESULT)

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	RESULT																															
Access	RW																															

Bit	Name	Function	Description	Access
31:0	RESULT	CRC Calculation Result	Read RESULT[15:0] to obtain the CRC-16 calculation result. Read RESULT[31:0] to obtain the CRC-32 calculation result. Write 0xFFFF to RESULT[15:0] to initialize CRC-16 calculation. Write 0xFFFFFFFF to RESULT[31:0] to initialize CRC-32 calculation.	RW

28.5.4 Data Register (CRC_DAT)

Offset	Bit Position																																					
0x80-0xFF	31	30	29	28	27	26	25	24	23	22	21	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	WO																																					

Bit	Name	Function	Description	Access
31:0	CRC_DAT	Data for Operation	Used to write the data to be operated on, supporting 3 write bit widths: 8-bit write mode: * ((uint8_t *)0x40015480) = 0xFF 16-bit write mode: * ((uint16_t *)0x40015480) = 0xFFFF 32-bit write mode: * ((uint32_t *)0x40015480) = 0xFFFFFFFF  Note The offset of this register is a range (0x80-0xFF). Any operation on any address within this range will be considered an operation on this register. Read always returns 0.	WO

29 Debug Controller (DBGC)

29.1 Introduction

This MCU's core is Cortex-M0+, which includes hardware for advanced debugging functions. 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. This MCU does not incorporate an MTB (Micro Trace Buffer) debugging component. When a debugger is connected to the MCU for debugging purposes, access permissions to system resources will be determined based on the security level.

Debugging interface supported:

- Serial debug trace interface SWD

29.2 Functional Description

29.2.1 Function Block Diagram

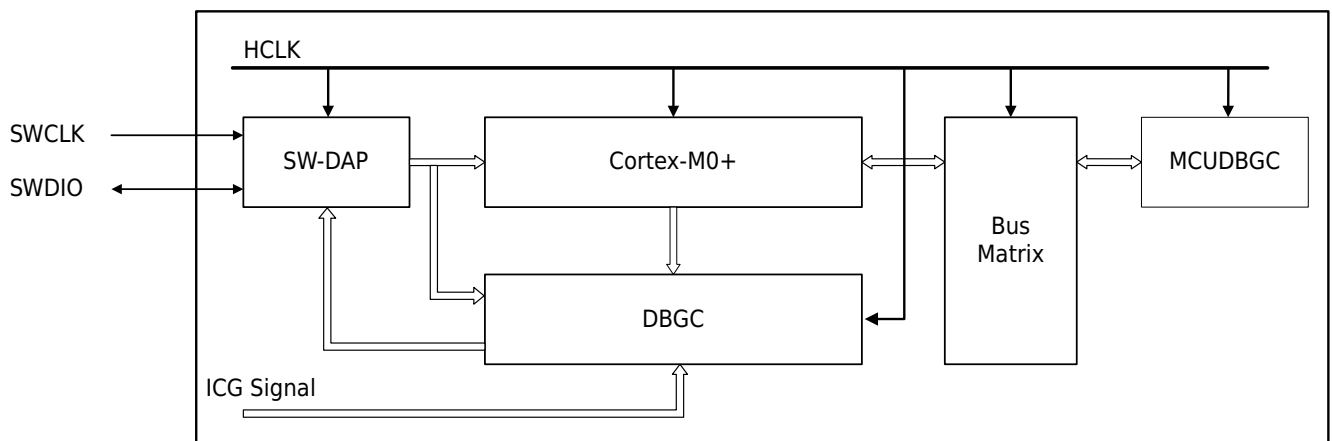


Figure 29-1 Debug Control System Block Diagram

The ARM Cortex-M0+ core provides integrated on-chip debugging support. It includes:

- SWD: Serial debug port
- BPU: Breakpoint unit
- DWT: Data watchpoint trigger
- Flexible debugging pin assignment



Note

For more information on the debugging functions supported by the ARM Cortex-M0+ core, see the "Cortex-M0+ Technical Reference Manual".

29.2.2 Reference Documentation

This product references the following ARM technical documentation:

- Cortex-M0+ Technical Reference Manual (TRM)
Available at developer.arm.com.

- ARM® debug interface V5

29.2.3 Debug Port

The MCU core integrates a Serial Wire Debug Port (SW-DP). The SW-DP is an ARM standard CoreSight debug port with an SW-DP (2-pin) interface.

29.2.4 Pin Arrangement and Debug Port Pins

The number of available pins varies depending on the MCU package. Therefore, some pin-related functions may differ by package.

Debug Port Pins

Two general-purpose I/O ports of the MCU can be used as SW-DP interface pins.

Table 29-1 Debug Port Pins

Pin Name	Debug Port		MCU Pin Assignment
	Type	Debug Assignment	
SWCLK	I	SWD serial clock	PC4
SWDIO	I/O	SWD serial wire data input/output	PA4

Flexible Pin Assignment

After reset (power-on or pin reset), all 2 pins used for SW-DP are designated as dedicated pins and are immediately available for debug tools. However, the MCU can disable part or all of the SW-DP ports, thereby releasing the related pins for use as GPIO.

For more detailed information on how to disable SW-DP port pins, refer to the "Special Control Register (PSPCR)" section in the "General-Purpose I/O (GPIO)" chapter.

Internal Pull-up on Pins

To avoid floating I/O levels, the MCU has built-in internal pull-up resistors on the SWD pins:

- SWDIO: Internal pull-up
- SWCLK: Internal pull-up

When no debugger is connected, user software can release the debug pins as general-purpose I/Os by configuring the GPIO controller. Since the internal pull-up of the chip is a weak pull-up of <100kΩ, it is recommended to use an external 10kΩ pull-up.

29.3 Register Description

29.3.1 Register List

Base address: 0x40015000

Table 29-2 Debug Controller (DBGC) Register Offset

Offset	Register	Description	Width	Access Host
0xA002D00C	CHIPID	Chip Identification Register	32	Debug IDE
0xA002D01C	TARGETID	Target Identification Register	32	Debug IDE
0x0000	MCUDBGSTAT	DBG Status Register	32	CPU/Debug IDE
0x0004	MCUSTPCTL	Peripheral Debug Stop Control Register	32	CPU/Debug IDE

29.3.2 Chip Identification Register (CHIPID)

Offset: 0xA002D00C

Used to indicate company information and product model information.

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	0x58480118																															
Name	CHIPID																															
Access	RO																															

Bit	Name	Function	Description	Access
31:0	CHIPID	Chip Identification	CHIPID[31:16]: ASCII code for XH, 0x5848 CHIPID[15:0]: product model, 0x0118	RO

29.3.3 Target Identification Register (TARGETID)

Offset: 0xA002D01C

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	0xA044DF1																															
Name	VERSION		PARTNO										DESIGNER										Reserved									
Access	RO		RO										RO																			

Bit	Name	Function	Description	Access
31:28	VERSION	Version	-	RO
27:12	PARTNO	Part Number	Read always returns 0xA044	RO
11:1	DESIGNER	JEDEC Bank and Code	DESIGNER[10:7]: JEDEC Bank, read always returns 0xD DESIGNER[6:0]: JEDEC Code, read always returns 0x78	RO
0	Reserved	Reserved	Read always returns 1	-

29.3.4 DBG Status Register (MCUDBGSTAT)

DBG debugs power-on status check register.

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																														CDBGPWRUPACK	CDBGPWRUPREQ
Access																															RW	RW

Bit	Name	Function	Description	Access
31:2	Reserved	Reserved	-	-
1	CDBGPWRUPACK	Debugger Power-on Feedback	0b0: No feedback 0b1: Debug power-on feedback	RW
0	CDBGPWRUPREQ	Debugger Power-on Request	0b0: No power-on request 0b1: Power-on request	RW

29.3.5 Peripheral Debug Stop Control Register (MCUSTPCTL)

The peripheral module pauses control when the CPU is in the debug state.

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	0x00000019																																
Name	Reserved								TMRA1STP	Reserved		TMR41STP	Reserved		TMR22STP	TMR21STP	Reserved		TMR01STP	Reserved								LVD2STP	LVD1STP	RTC5TP	Reserved		SWDTSTP
Access									RW			RW			RW	RW			RW									RW	RW	RW			RW

Bit	Name	Function	Description	Access
31:24	Reserved	Reserved	-	-
23	TMRA1STP	TimerA Count Pause Signal	0b0: Counter continues counting even when the core is stopped 0b1: Counter pauses counting when the core is stopped	RW
22:21	Reserved	Reserved	-	-
20	TMR41STP	Timer4 Count Pause Signal	0b0: Counter continues counting even when the core is stopped 0b1: Counter pauses counting when the core is stopped	RW
19	Reserved	Reserved	-	-
18	TMR22STP	Timer2_2 Count Pause Signal	0b0: Counter continues counting even when the core is stopped 0b1: Counter pauses counting when the core is stopped	RW
17	TMR21STP	Timer2_1 Count Pause Signal	0b0: Counter continues counting even when the core is stopped 0b1: Counter pauses counting when the core is stopped	RW
16:15	Reserved	Reserved	-	-
14	TMR01STP	Timer0 Count Pause Signal	0b0: Counter continues counting even when the core is stopped 0b1: Counter pauses counting when the core is stopped	RW
13:5	Reserved	Reserved	-	-
4	LVD2STP	LVD2 Interrupt/Reset Mask	0b0: LVD2 interrupt application or reset occurs even when the core is stopped 0b1: LVD2 interrupt application or reset is masked when the core is stopped	RW

Bit	Name	Function	Description	Access
3	LVD1STP	LVD1 Interrupt/ Reset Mask	● 0b0: LVD1 interrupt application or reset occurs even when the core is stopped ● 0b1: LVD1 interrupt application or reset is masked when the core is stopped	RW
2	RTCSTP	RTC Count Pause Signal	● 0b0: Counter continues counting even when the core is stopped ● 0b1: Counter pauses counting when the core is stopped	RW
1	<i>Reserved</i>	<i>Reserved</i>	-	-
0	SWDTSTP	SWDT Count Pause Signal	● 0b0: Counter continues counting even when the core is stopped ● 0b1: Counter pauses counting when the core is stopped	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 Nov. 14, 2025	First official release.