XHSC 小华半导体 XIAOHUA SEMICONDUCTOR

HC32F336 Series

32-bit ARM® Cortex®-M4 Microcontroller

Reference Manual

Rev1.00 November 2025

Statement

★ Xiaohua Semiconductor Co., Ltd. (hereinafter referred to as "XHSC") reserves the right to change, correct, enhance, modify Xiaohua Semiconductor products and/or this document at any time without prior notice. Users can get the latest information before placing orders. XHSC products are sold in accordance with theterms and conditions of sale set forth in the Basic Contract for Purchase and Sales.

★ 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.

★ Any graphics or words marked "®" or "™" are trademarks of XHSC. All other products or services shown on XHSC products are the property of their respective owners.

★ Information in this notification replaces and replaces information in previous versions.

©2025 Xiaohua Semiconductor Co., Ltd. all rights reserved

Contents

Statement **2**
Contents **3**
Table Index **30**
Figure Index **35**
Document Conventions **44**
Register-Related Abbreviations 44
Glossary 44
About this Manual 44
Overview **45**
1 Memory Mapping **46**
1.1 Memory mapping 46
1.2 Bit band region 53
1.3 Address Remapping 53
1.4 Register Description 54
1.4.1 Access Protection Register (MMF_REMPRT) 55
1.4.2 Remap Control Register (MMF_REMCRx, x=0, 1) 55
2 Bus Architecture (BUS) **56**
2.1 Overview 56
2.2 Bus Architecture 57
2.3 Bus Function 58
3 Reset Management Unit (RMU) **59**
3.1 Introduction 59
3.2 Reset Mode and Reset Flag Bit 60
3.3 Reset Timing 62
3.3.1 Power-on reset 62
3.3.2 NRST Pin Reset 62
3.3.3 Brown-out Reset 63
3.3.4 Programmable Voltage Detection 1 Reset, Programmable Voltage Detection 2 Reset 63
3.3.5 Watchdog Reset, Special Watchdog Reset 65
3.3.6 Power-down Wake-up Reset 65
3.3.7 Software Reset 66
3.3.8 MPU Error Reset 66
3.3.9 RAM Parity Reset 67
3.3.10 RAMECC Reset 67
3.3.11 Clock Frequency Exception Reset 67

3.3.12 External High Speed Oscillator Abnormal Shock Reset 68
3.3.13 Cortex-M4 Lockup Reset 68
3.3.14 Judgment of Reset Mode 69
3.3.15 Module Function Reset 69
3.3.16 Reset Conditions for Each Module 70
3.4 Register Description 71
3.4.1 Function Reset Control 0 (RMU_FRST0) 72
3.4.2 Function Reset Control 1 (RMU_FRST1) 73
3.4.3 Function Reset Control 2 (RMU_FRST2) 74
3.4.4 Function Reset Control 3 (RMU_FRST3) 75
3.4.5 Reset Control Register (RMU_PRSTCR0) 76
3.4.6 Reset Flag Register (RMU _ RSTF0) 77
4 Clock Management Unit (CMU) 79
4.1 Overview 79
4.2 System Block Diagram 80
4.2.1 System Block Diagram 80
4.2.2 Clock Frequency Measurement Block Diagram 81
4.3 Clock Source Specification 82
4.4 Working Clock Specification 83
4.5 Crystal Oscillator Circuit 84
4.5.1 External High-speed Oscillator 84
4.5.2 External High-speed Oscillator Fault Detection 86
4.5.3 External High Speed Oscillator Fractional Frequency 88
4.5.4 External Low-speed Oscillator 89
4.6 Internal RC clock 90
4.6.1 HRC clock 90
4.6.2 MRC Clock 90
4.6.3 LRC Clock 91
4.6.4 SWDTLRC Clock 91
4.7 PLL Clock 92
4.8 Clock Switching Step 92
4.8.1 Clock Source Switch 93
4.8.2 Clock Divider Switching 94
4.9 Clock Output Function 95
4.10 Clock Frequency Measurement 96
4.10.1 Clock Frequency Measurement 96
4.10.2 Digital Filtering Function 97
4.10.3 Interrupt/Reset Function 97

4.11 Register Description 98
4.11.1 CMU XTAL Configuration Register (CMU _ XTALCFGR) 100
4.11.2 CMU XTAL Stability Configuration Register (CMU_XTALSTBCR) 100
4.11.3 CMU XTAL Fractional Divider Configuration Register (CMU_XTALDIVR) 101
4.11.4 CMU XTAL Fractional Divider Control Register (CMU_XTALDIVCR) 101
4.11.5 CMU XTAL Control Register (CMU _ XTALCR) 102
4.11.6 CMU XTAL Oscillatory Fault Control Register (CMU _ XTALSTDCR) 102
4.11.7 CMU XTAL Oscillatory Fault State Register (CMU _ XTALSTDSR) 103
4.11.8 CMU XTAL32 Configuration Register (CMU_XTAL32CFGR) 103
4.11.9 CMU XTAL32 Filter Register (CMU_XTAL32NFR) 103
4.11.10 CMU XTAL32 Control Register (CMU_XTAL32CR) 104
4.11.11 CMU HRC Calibration Register (CMU_HRCTRM) 104
4.11.12 CMU HRC Control Register (CMU _ HRCCR) 105
4.11.13 CMU MRC Calibration Register (CMU_MRCTRM) 105
4.11.14 CMU MRC Control Register (CMU_MRCCR) 106
4.11.15 CMU LRC Calibration Register (CMU_LRCTRM) 106
4.11.16 CMU LRC Control Register (CMU _ LRCCR) 107
4.11.17 CMU PLLH Configuration Register (CMU_PLLHCFGR) 108
4.11.18 CMU MPLL Control Register (CMU_PLLHCR) 110
4.11.19 CMU Timer Stabilizer (CMU _ OSCSTBSR) 110
4.11.20 Timer Switch Register of CMU System (CMU _ CKSWR) 111
4.11.21 CMU Clock Divider Configuration Register (CMU_SCFGR) 112
4.11.22 CMU CAN Clock Configuration Register (CMU_CANCKCFGR) 114
4.11.23 CMU ADC/DAC Clock Configuration Register (CMU _ PERICKSEL) 115
4.11.24 CMU Debug Clock Config Register (CMU_TPIUCKCFGR) 115
4.11.25 CMU MCO1 Clock Output Configuration Register (CMU_MCO1CFGR) 116
4.11.26 CMU MCO2 Clock Output Configuration Register (CMU_MCO2CFGR) 116
4.11.27 FCM Lower Limit Compare Value Register (FCM_LVR) 117
4.11.28 FCM Upper Limit Comparison Value Register (FCM_UVR) 117
4.11.29 FCM Counter Value Register (FCM_CNTR) 117
4.11.30 FCM Start Stop Register (FCM_STR) 118
4.11.31 FCM Measurement Object Control Register (FCM_MCCR) 118
4.11.32 FCM Measurement Reference Control Register (FCM_RCCR) 119
4.11.33 FCM Interrupt Reset Control Register (FCM_RIER) 120
4.11.34 FCM Flags Register (FCM_SR) 120
4.11.35 FCM Flag Clear Register (FCM_CLR) 121
5 Power Control (PWC) 122
5.1 Overview 122

5.2 Distribution of Power 122
5.3 Power Voltage Detection Unit (PVD) Description 123
5.3.1 Description of Action of Power-on/Power-off Reset 123
5.3.2 Brown-out Reset (BOR) Description 124
5.3.3 Programmable Voltage Detection 1 (PVD1), Programmable Voltage Detection 2 (PVD2) 125
5.3.4 PVD1, PVD2 Interrupt/Reset Block Diagram 126
5.3.5 Input/Output Pin 126
5.3.6 PVD1 Interrupt and Reset 126
5.3.7 PVD2 Interrupt and Reset 127
5.3.8 Internal Voltage Sampling and Detection Function 128
5.4 Wake-up timer 129
5.5 Action Mode and Low Power Mode 129
5.5.1 Sleep Mode 132
5.5.2 Stop Mode 133
5.5.3 Power Down Mode 135
5.5.4 Reduce System Clock Speed 138
5.5.5 Turn off unused clock source 138
5.5.6 Stop Functional Clock 139
5.5.7 Turn-off Unused RAM 139
5.6 Register Protection Function 140
5.7 Register Description 141
5.7.1 Power Mode Control Register 0 (PWC_PWRC0) 143
5.7.2 Power Mode Control Register 1 (PWC_PWRC1) 143
5.7.3 Power Mode Control Register 4 (PWC_PWRC4) 144
5.7.4 Power-Down Wake-up Enable Register 0 (PWC_PDWKE0) 144
5.7.5 Power-Down Wake-up Enable Register 1 (PWC_PDWKE1) 145
5.7.6 Power-Down Wake-up Enable Register 2 (PWC_PDWKE2) 145
5.7.7 Power-down Wake-up Event Edge Selection Register (PWC_PDWKES) 146
5.7.8 Power-down Wake-up Flag Register 0 (PWC_PDWKF0) 146
5.7.9 Power-down Wake-up Flag Register 1 (PWC_PDWKF1) 147
5.7.10 Power Mode Control Register 5 (PWC_PWRC5) 147
5.7.11 Power Mode Control Register 6 (PWC_PWRC6) 148
5.7.12 Low Power Status Register (PWC_LPMCSR) 148
5.7.13 Low Power State Clear Register (PWC_LPMCSCR) 148
5.7.14 Function Clock Control 0 (PWC_FCG0) 149
5.7.15 Function Clock Control 1 (PWC_FCG1) 150
5.7.16 Function Clock Control 2 (PWC_FCG2) 151

5.7.17 Function Clock Control 3 (PWC_FCG3) 152
5.7.18 PWC_FCG0 Protection Control (PWC_FCG0PC) 153
5.7.19 Function Protection Control Register (PWC_FPRC) 153
5.7.20 Stop Mode Control Register (PWC_STPMCR) 154
5.7.21 RAM Power Control Register 0 (PWC_RAMPC0) 154
5.7.22 Peripheral RAM Low Power Control Register (PWC_PRAMLPC) 155
5.7.23 PVD Control Register 0 (PWC_PVDCR0) 155
5.7.24 PVD Control Register 1 (PWC_PVDCR1) 156
5.7.25 PVD Filter Control Register (PWC_PVDFCR) 157
5.7.26 PVD Level Control Register (PWC_PVDLCR) 158
5.7.27 PVD Interrupt Control Register (PWC_PVDICR) 159
5.7.28 PVD Detection Status Register (PWC_PVDDSR) 159
5.7.29 Wake-up Timer Control Register (PWC_WKTCR) 160
6 Initialization Configuration (ICG) 161
6.1 Overview 161
6.2 Register Description 162
6.2.1 Initialization Configuration Register 0 (ICG0) 163
6.2.2 Initialization Configuration Register 1 (ICG1) 165
6.2.3 Initialization Configuration Register 3 (ICG3) 165
7 Embedded Flash Memory (EFM) 166
7.1 Overview 166
7.2 Main Features 166
7.3 Embedded Flash 166
7.4 Read interface 168
7.4.1 Relationship between CPU clock and Flash read time 168
7.5 Flash Read Cache 169
7.6 Flash Programming and Erasing Operations 170
7.6.1 Unlock the EFM_KEY1 Register 170
7.6.2 Write Protection Function 170
7.6.3 Single Programming Without Read BackFunction 170
7.6.4 Single Program Readback Function 171
7.6.5 Continuous ProgrammingFunction 172
7.6.6 Erase Function 172
7.6.7 Full Erase Function 173
7.6.8 Data Security Protection 173
7.6.9 D-BUS Read Protection Function 174
7.6.10 Bus Hold Function 174
7.6.11 Interrupt 175

7.7 One-Time Programmable Byte (OTP) 176
7.8 Boot Swap 177
7.9 Register Description 179
7.9.1 Access Protection Register (EFM _ FAPRT) 180
7.9.2 Flash Key 1 Register (EFM_KEY1) 180
7.9.3 Flash Key 2 Register (EFM_KEY2) 180
7.9.4 Flash Stop Register (EFM_FSTP) 181
7.9.5 Read Mode Register (EFM _ FRMC) 181
7.9.6 Flash Erase Mode Register (EFM_FWMC) 182
7.9.7 Flash Status Register (EFM_FSR) 183
7.9.8 Flash Status Clear Register (EFM_FSCLR) 184
7.9.9 Flash Interrupt Enable Register (EFM_FITE) 184
7.9.10 Flash Boot Swap Status Register (EFM_FSWP) 185
7.9.11 Chip Specific Flag Register (EFM_CHIPID) 185
7.9.12 UNIQUE ID Register (EFM_UQID0) 185
7.9.13 UNIQUE ID Register1 (EFM_UQID1) 186
7.9.14 UNIQUE ID Register2 (EFM_UQID2) 186
7.9.15 Flash Write Protection Latch Register (EFM_WLOCK) 186
7.9.16 Flash Write Protection Register (EFM_F0NWPRT) 187
7.10 Precautions 188
8 Built-in SRAM (SRAM) 189
8.1 Overview 189
8.2 Register Description 191
8.2.1 SRAM Check Control Register (SRAM_CKCR) 192
8.2.2 SRAM Check Control Protection Register (SRAMC_CKPR) 194
8.2.3 SRAM Check Status Register (SRAMC_CKSR) 194
8.2.4 SRAM0 Error Injection Enable Register (SRAM0_EIEN) 195
8.2.5 SRAM0 Error Injection Bit Enable Register 0 (SRAM0_EIBIT0) 195
8.2.6 SRAM0 Error Injection Bit Enable Register 1 (SRAM0_EIBIT1) 195
8.2.7 SRAM0 Check Error Address Register (SRAM0_ECCERRADDR) 196
8.2.8 SRAMH Error Injection Enable Register (SRAMH_EIEN) 196
8.2.9 SRAMH Error Injection Bit Enable Register 0 (SRAMH_EIBIT0) 196
8.2.10 SRAMH Error Injection Bit Enable Register 1 (SRAMH_EIBIT1) 197
8.2.11 SRAMH Check Error Address Register (SRAMH_ECCERRADDR) 197
8.2.12 Ret SRAM Error Injection Enable Register (SRAMB_EIEN) 197
8.2.13 Ret SRAM Error Injection Bit Enable Register 0 (SRAMB_EIBIT0) 198
8.2.14 Ret SRAM Error Injection Bit Enable Register 1 (SRAMB_EIBIT1) 198
8.2.15 Ret SRAM Check Error Address Register (SRAMB_ECCERRADDR) 198

8.3 Precautions 199
9 General-Purpose I/Os (GPIO) 200
9.1 Introduction 200
9.2 Functional Description 201
9.2.1 GPIO Port Basic Structure 201
9.2.2 General purpose Input/Output Function 202
9.2.3 Peripheral Functions 202
9.2.4 Dual Peripheral Function 203
9.2.5 Event Port Input and Output Functions 203
9.2.6 External Interrupt EIRQ Input Function 204
9.2.7 Analog Function 204
9.2.8 General-Purpose Control 204
9.3 Register Description 206
9.3.1 General Purpose Input Data Register (PIDRx) 207
9.3.2 General PurposeOutput Data Register (PODRx) 207
9.3.3 General Purpose Output License Register (POERx) 208
9.3.4 General Purpose Output Set Register (POSRx) 208
9.3.5 General Purpose Output Reset Register (PORRx) 209
9.3.6 General Purpose Output Flip Register (POTRx) 209
9.3.7 HRPWM Control Register (HRPWM) 210
9.3.8 Special Control Register (PSPCR) 212
9.3.9 Common Control Register (PCCR) 212
9.3.10 Write Protection Register (PWPR) 213
9.3.11 General Purpose Control Register (PCRxy) 214
9.3.12 Function Select Register (PFSRxy) 216
9.3.13 Event Port Direction Selection Register (PEVNTDIRRm) 217
9.3.14 Event Port Input Data Register (PEVNTIDRm) 217
9.3.15 Event Port Output Data Register (PEVNTODRm) 218
9.3.16 Event Port Output Data Reset Register (PEVNTORRm) 218
9.3.17 Event Port Output Data Set Register (PEVNTOSRm) 219
9.3.18 Event Port Rising Edge Input Permission Register (PEVNTRISRm) 219
9.3.19 Event Port Falling Edge Input Enable Register (PEVNTFALRm) 220
9.3.20 Event Port Input Filter Control Register (PEVNTNFCR) 221
9.4 Precautions 222
10 Interrupt controller (INTC) 223
10.1 Introduction 223
10.2 System Block Diagram 224
10.3 Directional Scale 225

10.3.1 Interrupt Digraph 225
10.3.2 Interrupt Event List 230
10.4 Functional Description 246
10.4.1 Nomaskable Interrupt 246
10.4.2 External Interrupt 246
10.4.3 Interrupt Event Multiplexer 247
10.4.4 Software Interrupt 247
10.4.5 Event Input 247
10.4.6 WFE Event Management 248
10.4.7 Noise Filter 248
10.4.8 Low Power Mode Return 249
10.5 Register Description 250
10.5.1 Nomaskable Interrupt Enable Register (INTC _ NMIER) 253
10.5.2 Nomaskable Interrupt Flag Register (INTC _ NMIFR) 254
10.5.3 Non-Maskable Interrupt Flag Clear Register (INTC_NMIFCR) 255
10.5.4 External Interrupt Control Register (INTC_EIRQCRx, x=0~15) 256
10.5.5 External Interrupt Flag Register (INTC_EIFR) 257
10.5.6 External Interrupt Flag Clear Register (INTC_EIFCR) 257
10.5.7 Interrupt Event Selection Register (INTC_INTSEL0~15) 257
10.5.8 Interrupt Event Enable Register (INTC_INTEN0~15) 258
10.5.9 Stop Mode Wake-up Event Enable Register (INTC_WKEN) 259
10.5.10 Software Interrupt Register (INTC_SWIER) 260
10.5.11 Event Enable Register (INTC_EVTER) 260
10.5.12 Interrupt Request Enable Register (INTC_IER) 260
10.5.13 FPU Interrupt Enable Register (INTC_FPUIER) 261
10.6 Precautions 262
11 Automatic Operation System (AOS) 263
11.1 Overview 263
11.1.1 Function Overview 263
11.2 Functional Description 264
11.2.1 AOS Block Diagram 264
11.2.2 AOS Source Eventsand Targets 265
11.2.3 Dedicated Trigger Source 266
11.2.4 Common Trigger Source 266
11.3 Register Description 267
11.3.1 Peripheral Trigger Event Register (INTSFTTRG) 268
11.3.2 DMA Transfer Start Trigger Source Selection Register (DMA_TRGSELx, x=0~7) 268
11.3.3 DMA Channel Reset Trigger Source Selection Register (DMA_RC_TRGSEL) 269

11.3.4 Timer6 Hardware Trigger Event Selection Register (TMR6_TRGSELx) (x=0~3) 270
11.3.5 HRPWM Hardware Trigger Event Selection Register (HRPWM_TRGSELx, x=0~3) ..271
11.3.6 Timer4 Hardware Trigger Event Selection Register (TMR4_TRGSEL).....272
11.3.7 Event Port Trigger Source Select Register (PEVNT_TRGSEL12, PEVNT_TRGSEL34) 273
11.3.8 Timer0 Hardware Trigger Event Selection Register (TMR0_TRGSEL).....274
11.3.9 TimerA Hardware Trigger Event Selection Register (TMRA_TRGSELx, x = 0 ~ 3)...275
11.3.10 ADC1 Conversion Start On-chip Trigger Source Selection Register (ADC1_TRGSELx, x = 0~1) 276
11.3.11 ACD2 Conversion Start On-chip Trigger Source Selection Register (ADC2_TRGSELx, x = 0~1) 277
11.3.12 ADC3 Conversion Start On-chip Trigger Source Selection Register (ADC3_TRGSELx, x=0, 1) 278
11.3.13 Common Trigger Source Selection Register (AOS_COMTRGSELx, x = 1~2).....278
12 Memory Protection Unit (MPU) 279
12.1 Overview 279
12.2 Functional Description 279
12.2.1 Locale Setting 279
12.2.2 Permission Settings.....279
12.2.3 MPU Action Selection 280
12.2.4 Start MPU.....280
12.3 Application Examples 280
12.3.1 Only Allow Partial Space Access.....280
12.3.2 Only Some Space Access Prohibited 281
12.4 Register Description 282
12.4.1 Region Range Description Register (MPU_RGDn, n=0~15) 283
12.4.2 Status Flag Register (MPU_SR).....283
12.4.3 Flag Clear Register (MPU_ECLR).....284
12.4.4 Write Protection Register (MPU_WP) 284
12.4.5 IP Access Protection Register (MPU_IPPR) 285
12.4.6 MSP Protection Start Address Register (MPU_MSPPBA) 286
12.4.7 MSP Protection Control Register (MPU_MSPPCTL) 286
12.4.8 PSP Protection Start Address Register (MPU_PSPPBA).....287
12.4.9 PSP Protection Control Register (MPU_PSPPCTL).....287
12.4.10 SMPU1 Region Enable Register (MPU_S1RGE) 288
12.4.11 SMPU1 Region Write Permission Register (MPU_S1RGWP).....289
12.4.12 SMPU1 Region Read Permission Register (MPU_S1RGRP) 290
12.4.13 SMPU1 Control Register (MPU_S1CR).....291
13 Internal Clock Calibrator (CTC) 292

13.1 Overview 292
13.2 Main Features 292
13.3 Functional Description 292
13.3.1 CTC Block Diagram 292
13.3.2 Reference Clock 293
13.3.3 Frequency Calibration 299
13.4 Interrupt 300
13.5 Application Examples 300
13.6 Register Description 302
13.6.1 Clock Calibration Control Register1 (CTC _ CR1) 303
13.6.2 Clock Calibration Control Register 2 (CTC _ CR2) 305
13.6.3 Clock Calibration Status Register (CTC _ STR) 306
13.6.4 Clock Calibration Counter (CTC_CNT) 306
14 Direct Memory Access Controller (DMA) 307
14.1 Overview 307
14.2 Main Features 307
14.3 Functional Description 308
14.3.1 DMA Block Diagram 308
14.3.2 Enable DMA Controller 308
14.3.3 Channel Selection and Channel Priority 308
14.3.4 Start DMA 309
14.3.5 Data Block 309
14.3.6 Transmission Address Control 309
14.3.7 Number of Transmissions 310
14.3.8 Chain Transmission 310
14.3.9 Discontinuous Address Transmission 312
14.3.10 Channel Reset 314
14.3.11 Premature Interrupt of Transmission 315
14.4 Interrupt 316
14.5 Application Examples 316
14.5.1 Memory-to-memory transmission 316
14.5.2 Memory-to-Peripheral Transmission 318
14.5.3 Memory-to-Memory Linkage 319
14.6 Register Description 322
14.6.1 DMA Enable Register (DMA _ EN) 324
14.6.2 Interrupt Status Register 0 (DMA_INTSTAT0) 324
14.6.3 Interrupt Status Register 1 (DMA_INTSTAT1) 325
14.6.4 Interrupt Mask Register 0 (DMA_INTMASK0) 325

14.6.5 Interrupt Mask Register 1 (DMA_INTMASK1) 326
14.6.6 Interrupt Reset Register 0 (DMA_INTCLR0) 326
14.6.7 Interrupt Reset Register 1 (DMA_INTCLR1) 327
14.6.8 Channel Enable Register (DMA _ CHEN) 327
14.6.9 Channel Enable Reset Register (DMA _ CHENCLR) 328
14.6.10 Channel Reset Control Register (DMA_RCFGCTL) 329
14.6.11 Software Start Register (DMA_SWREQ) 330
14.6.12 Transfer Request Status Register (DMA_REQSTAT) 330
14.6.13 Channel State Observation Register (DMA _ CHSTAT) 331
14.6.14 Transfer Source Address Register (DMA_SARx, x=0~7) 331
14.6.15 Transfer Destination Address Register (DMA_DARx, x=0~7) 331
14.6.16 Data Control Register (DMA_DTCTLx, x=0~7) 332
14.6.17 Repeat Region Size Register (DMA_RPTx, x=0~7) 333
14.6.18 Repeat Region Size Register (DMA_RPTBx, x=0~7) 334
14.6.19 Source Address Discontinuous Transfer Control Register (DMA_SNSEQCTLx, x=0~7) 335
14.6.20 Source Address Discontinuous Transfer Control Register B (DMA_SNSEQCTLBx, x=0~7) 336
14.6.21 Target Address Discontinuous Transfer Control Register (DMA_DNSEQCTLx, x=0~7) 337
14.6.22 Target Address Discontinuous Transfer Control Register B (DMA_DNSEQCTLBx, x=0~7) 338
14.6.23 Link Pointer Register (DMA_LLPx, x=0~7) 339
14.6.24 Channel Control Register (DMA_CHCTLx, x=0~7) 340
14.6.25 Transfer Source Address Monitoring Register (DMA_MONSARx, x=0~7) 341
14.6.26 Transfer Target Address Monitoring Register (DMA_MONDARx, x=0~7) 341
14.6.27 (DMA_MONDTCTLx, x=0~7) 342
14.6.28 Repeat Region Counter Monitoring Register (DMA_MONRPTx, x = 0 ~ 7) 342
14.6.29 Source Address Discontinuous Transfer Counter Monitoring Register (DMA_MONSNSEQCTLx, x = 0 ~ 7) 343
14.6.30 Target Address Discontinuous Transfer Counter Monitoring Register (DMA_MONDNSEQCTLx, x = 0 ~ 7) 344
14.7 Precautions 345
15 Voltage Comparator (CMP) 346
15.1 Overview 346
15.2 Main Features 346
15.3 Functional Description 347
15.3.1 CMP Functional Block Diagram 347

15.3.2 Input Selection 348
15.3.3 Normal Comparison 348
15.3.4 Window Comparison 349
15.3.5 Input Scan 351
15.3.6 Window Output 352
15.3.7 Hysteresis Function 355
15.3.8 Noise Filter 355
15.3.9 External Pin output 355
15.4 Interrupt and Events 355
15.5 Register Description 356
15.5.1 Comparator Operating Mode Register (CMPx_MDR, x=1~3) 357
15.5.2 Comparator Filter and Interrupt Register (CMPx_FIR, x=1~3) 358
15.5.3 Comparator Output Control Register (CMPx_OCR, x=1~3) 359
15.5.4 Comparator Result Monitoring Register (CMPx_OMR, x=1~3) 359
15.5.5 Comparator Positive and Negative Input Selection Register (CMPx_PMSR, x=1~3) 360
15.5.6 Comparator Output Blank Window Setting Register 1 (CMPx_BWSR1, x=1~3) 361
15.5.7 Comparator Output Blank Window Setting Register 2 (CMPx_BWSR2, x=1~3) 362
15.5.8 Comparator Input Scan Control Register (CMPx_SCCR, x=1~3) 363
15.5.9 Comparator Input Scan Monitor Register (CMPx_SCMR, x=1~3) 364
15.6 Precautions 365
15.6.1 Module Stop Function 365
15.6.2 Power Consumption when the Module is Stopped 365
15.6.3 Power Consumption When Stopping Low Power Mode 365
16 Analog-to-Digital Converter (ADC) 366
16.1 Overview 366
16.2 Main Features 366
16.3 Functional Description 367
16.3.1 ADC System Block Diagram 367
16.3.2 ADC Clock 368
16.3.3 Channel Selection 369
16.3.4 Trigger Source Selection 370
16.3.5 Sequence A single scan mode 371
16.3.6 Sequence A continuous scanning mode 372
16.3.7 Sequence A Data Buffer Mode 373
16.3.8 Double Sequence Scanning Mode 374
16.3.9 Analog Watchdog Function 376
16.3.10 Sampling Time and Conversion Time of Analog Input 377

16.3.11 ADC Data Register Automatic Clear Function 379
16.3.12 Conversion Data Average Calculation Function 379
16.3.13 Oversampling 380
16.3.14 Multi-ADC Cooperative Working Mode 380
16.4 Interrupt and Events 386
16.5 Register Description 388
16.5.1 ADC Start Register (ADC_STR) 390
16.5.2 ADC Control Register 0 (ADC_CR0)) 391
16.5.3 ADC control register 1 (ADC_CR1) 392
16.5.4 ADC Control Register 2 (ADC_CR2) 392
16.5.5 ADC Conversion Start Trigger Register (ADC_TRGSR) 393
16.5.6 ADC Channel Select Register A (ADC_CHSELRA) 394
16.5.7 ADC Channel Select Register B (ADC_CHSELRB) 395
16.5.8 ADC Average Channel Select Register (ADC_AVCHSELR) 396
16.5.9 ADC Extended Channel Select Register (ADC_EXCHSELR) 396
16.5.10 ADC Sampling Status Register (ADC_SSTRx) 397
16.5.11 ADC Channel Mapping Control Register (ADC_CHMUXR) 398
16.5.12 ADC Interrupt Status Register (ADC_ISR) 399
16.5.13 ADC Interrupt Enable Register (ADC_ICR) 399
16.5.14 ADC Interrupt Status Reset Register (ADC_ISCLRR) 399
16.5.15 ADC Synergy Mode Control Register (ADC_SYNCCR) 400
16.5.16 ADC Data Register (ADC_DRx) 401
16.5.17 Analog Watchdog Control Register (ADC_AWDCR) 402
16.5.18 Analogwatchdog Status Register (AD _ AWDSR) 403
16.5.19 Analogwatchdog Status Reset Register (ADC _ AWDSCLRR) 403
16.5.20 Analog Watchdog Threshold Registers (ADC_AWD0DR0, ADC_AWD0DR1, ADC_AWD1DR0, ADC_AWD1DR1) 404
16.5.21 Analog Watchdog Comparison Channel Selection Register (ADC _ AWD0 CHSR, ADC _ AWD1 CHSR) 404
16.6 Precautions 405
16.6.1 Considerations when Reading Data Register 405
16.6.2 Scan Finish Interrupt Handling Considerations 405
16.6.3 Precautions for Module Stop and Low Power Consumption Setting 405
16.6.4 ADC Conversion Analog Channel Input Pin Setting 405
16.6.5 Noise Control 405
17 Digital-to-Analog Converter (DAC) 406
17.1 Overview 406
17.2 Main Features 406

17.3 Functional Description 406
17.3.1 DAC Functional Block Diagram 406
17.3.2 DAC Conversion 407
17.3.3 Synchronous conversion 408
17.3.4 ADC conversion priority mode 408
17.3.5 Data Transfer Mode 409
17.4 Register Description 410
17.4.1 DAC Data Register (DADRy, y=1~2) 411
17.4.2 DAC Control Register (DACR) 412
17.4.3 DAC Analog Output Control Register (DAOCR) 413
17.4.4 DAC ADC Conversion Priority Control Register (DAADPCR) 414
17.4.5 DAC Control Register 2 (DACR2) 415
17.4.6 DAC Data ACTIVE Value Register (DADACTRy, y=1~2) 415
17.5 Precautions 416
17.5.1 Setting Module Stop Function 416
17.5.2 DAC Operation when the Module is Stopped 416
17.5.3 DAC Operation in Stop Mode 416
17.5.4 DAC Operation in Power-Down Mode 416
17.5.5 Notes on Using Output Amplifiers 416
18 Advanced Control Timer (Timer6) 417
18.1 Overview 417
18.2 Main Features 417
18.3 Functional Description 418
18.3.1 Timer6 Block Diagram 418
18.3.2 Waveform Mode 419
18.3.3 Timer Selection 419
18.3.4 Counting Direction 420
18.3.5 Comparison Output 420
18.3.6 Capture Input 421
18.3.7 Counter Refresh 422
18.3.8 Software Synchronization 423
18.3.9 Hardware Synchronization 424
18.3.10 Pulse Width Measurement 426
18.3.11 Periodic Measurement 427
18.3.12 Cache Function 427
18.3.13 Digital Filtering 433
18.3.14 Universal PWM Output 434
18.3.15 Periodic Interval Response 441

18.3.16 Orthogonal Coding Count 441
18.3.17 EMB Control 447
18.3.18 Function Summary Table 448
18.4 Interrupt and Events 449
18.4.1 Interrupt 449
18.4.2 Events 450
18.5 Application Examples 452
18.5.1 Basic Counting and Interrupt Actions 452
18.5.2 Comparison Output and Interrupt Action 452
18.5.3 Capture Input and Interrupt Actions 453
18.5.4 Buffer Transfer Action (Period Reference Value) 453
18.5.5 Cache Transfer Action (Common Benchmark Value) 454
18.5.6 Cache Transfer Action (Dedicated Benchmark Value) 454
18.5.7 Buffer Transfer Action (Deadband Reference Value) 455
18.5.8 Synchronous Start Action (Software Method) 455
18.5.9 Synchronous Start Action (Hardware Method) 455
18.5.10 Quadrature Encoder Counting Action (2 Phases) 456
18.5.11 Quadrature Encoder Counting Action (3 Phases) 456
18.5.12 Single PWM Output 457
18.5.13 Complementary PWM Output (Software Dead Time) 457
18.5.14 Complementary PWM Output (Hardware Dead Zone) 458
18.5.15 EMB Monitoring and Interrupt Action 458
18.6 Register Description 460
18.6.1 Universal Counter Value Register (TMR6_CNTER) 462
18.6.2 Universal Refresh Value Register (TMR6_UPDAR) 462
18.6.3 Universal Period Reference Value Register (TMR6_PER, m=A~C) 462
18.6.4 Universal Comparison Reference Value Register (TMR6_GCMmR, m=A~F) 463
18.6.5 Dedicated Comparison Reference Value Register (TMR6_SCMmR, m=A~F) 463
18.6.6 Dead Time Reference Value Register (TMR6_DTmnR, m=D, U, n=A, B) 463
18.6.7 Universal Control Register (TMR6_GCONR) 464
18.6.8 Interrupt Control Register (TMR6_ICONR) 465
18.6.9 Cache Control Register (TMR6_BCONR) 466
18.6.10 Dead Band Control Register (TMR6_DCONR) 468
18.6.11 Port Control Register A (TMR6_PCNAR) 469
18.6.12 Port Control Register B (TMR6_PCNBR) 471
18.6.13 Filter Control Register (TMR6_FCNGR) 473
18.6.14 Valid Period Register (TMR6_VPERR) 474
18.6.15 Status Flag Register (TMR6_STFLR) 475

18.6.16 Hardware Start Event Select Register (TMR6_HSTAR) 476
18.6.17 Hardware Stop Event Select Register (TMR6_HSTPR) 477
18.6.18 Hardware Clear Event Select Register (TMR6_HCLRR) 478
18.6.19 Hardware Refresh Event Select Register (TMR6_HUPDR) 479
18.6.20 Hardware Capture Event Select Register A (TMR6_HCPAR) 480
18.6.21 Hardware Capture Event Select Register B (TMR6_HCPBR) 482
18.6.22 Hardware Increment Event Select Register (TMR6_HCUPR) 484
18.6.23 Hardware Decrement Event Select Register (TMR6_HCDOR) 486
18.6.24 Common Filter Control Register (TMR6_FCNTR) 488
18.6.25 Common Software Synchronous Start Control Register (TMR6_SSTAR) 489
18.6.26 Common Software Synchronous Stop Control Register (TMR6_SSTPR) 489
18.6.27 Common Software Synchronous Clear Control Register (TMR6_SCLRR) 490
18.6.28 Common Software Synchronous Refresh Control Register (TMR6_SUPDR) 490
18.7 Precautions 491
19 High-resolution PWM (HRPWM) 493
19.1 Overview 493
19.2 Basic Block Diagram 495
19.3 Functional Description 496
19.3.1 HRWPM Block Diagram 496
19.3.2 Counter Mode 498
19.3.3 Counter Clock 498
19.3.4 Calibration Function 499
19.3.5 Counting Direction 499
19.3.6 Capture Function 500
19.3.7 PWM Action Qualifer(AQ) 501
19.3.8 Dead Zone Module 505
19.3.9 Idle Control 506
19.3.10 Phase Function 516
19.3.11 External Events 518
19.3.12 Software Synchronization 525
19.3.13 Hardware Synchronization 526
19.3.14 Synchronous Input Port Digital Filtering 527
19.3.15 Sync Output 528
19.3.16 Shadow Function 528
19.3.17 PWM Output 535
19.3.18 Periodic Interval Response 546
19.3.19 EMB(Fault) Control 547
19.3.20 Output Management 548

19.3.21 DAC Trigger Source 550
19.3.22 Function Summary Table 552
19.4 Interrupt 553
19.4.1 Interrupt Output 553
19.4.2 Event Output 554
19.5 Typical Application 556
19.5.1 Calibration Function 556
19.5.2 Basic Counting and Interrupt Actions 557
19.5.3 Compare Output and Interrupt Action 557
19.5.4 Burst Mode And Interrupt Action 558
19.5.5 Shadow Transfer Action (Period Reference Value) 558
19.5.6 Shadow Transfer Action (general compare) 558
19.5.7 Shadow Transfer Action (Special-compare) 559
19.5.8 Shadow Transfer Action (External Event Filter Offset Value) 559
19.5.9 Shadow Transfer Action (External Event Filter Window) 560
19.5.10 Buffer Transfer Action (Burst Mode period) 560
19.5.11 Buffer Transfer Operation (Burst Mode Compare Register) 561
19.5.12 Buffer Transfer Action (Deadband) 561
19.5.13 Synchronous Start Action (Software Method) 562
19.5.14 Synchronous Start Action (Hardware Method) 562
19.5.15 Sync Output 562
19.5.16 Single PWM Output 563
19.5.17 Complementary PWM Output (Software Dead Time) 563
19.5.18 Complementary PWM Output (Hardware Dead Zone) 563
19.5.19 EMB Monitoring and Interrupt Action 564
19.6 Register Description 565
19.6.1 HRPWM Unit 1~6 Registers 569
19.6.2 HRPWM Common Registers 637
19.7 Precautions 661
20 General Control Timer (Timer4) 663
20.1 Overview 663
20.2 Main Features 663
20.3 Functional Description 664
20.3.1 Timer4 Block Diagram 664
20.3.2 Basic Action 665
20.3.3 Cache Function 669
20.3.4 Universal PWM Output 676
20.3.5 Periodic interval response 681

20.3.6 EMB Control 683
20.3.7 Monitoring Output 684
20.3.8 Hardware Startup 685
20.3.9 Synchronous Startup 685
20.4 Interrupt and Events 685
20.4.1 Universal Purpose Count Compare Match Interruptand Event 685
20.4.2 Counting Period Match Interruptand Events 686
20.4.3 Reload Count Match Interruptand Event 686
20.4.4 Dedicated Compare Match Interrupts and Events 686
20.5 Register Description 688
20.5.1 Count Value Register (TMR4 _ CNTR) 691
20.5.2 Periodic Reference Register (TMR4 _ CPSR) 691
20.5.3 Control State Register (TMR4 _ CCSR) 692
20.5.4 Valid Period Register (TMR4 _ CVPR) 693
20.5.5 Generic Baseline Register (TMR4 _ OCCRm) 693
20.5.6 Universal Control State Register (TMR4 _ OCSRn) 694
20.5.7 Generic Extended Control Register (TMR4 _ OCERn) 695
20.5.8 Universal Mode Control Register (TMR4 _ OCMRm) 697
20.5.9 Special Reference Register (TMR4 _ SCCRm) 701
20.5.10 Special Control State Register (TMR4 _ SCSRm) 702
20.5.11 Dedicated Extended Control Register (TMR4_SCER) 703
20.5.12 Special Mode Control Register (TMR4 _ SCMRm) 703
20.5.13 PWM Basic Control Register (TMR4 _ POCRn) 704
20.5.14 PWM Status Control Register (TMR4_PSCR) 705
20.5.15 PWM Filtering Control Register (TMR4 _ PFSRn) 708
20.5.16 PWM Dead Zone Control Register (TMR4 _ PDA < B > Rn) 708
20.5.17 Overload Control Status Register (TMR4 _ RCSR) 709
20.5.18 Dedicated Interrupt Control Register (TMR4_SCIR) 711
20.5.19 Dedicated Flag Control Register (TMR4_SCFR) 711
21 Emergency Brake Module (EMB) 712
21.1 Overview 712
21.2 Functional Description 712
21.2.1 EMB Structure Block Diagram 714
21.2.2 Control PWM Signal Output When External Port Input Level Changes 715
21.2.3 Stop PWM Signal Output When The Level Of PWM Output Port Is In The Same Phase (Same High Or Same Low) 716
21.2.4 Stop PWM Signal Output According To Voltage Comparator Comparison Result 717
21.2.5 Stop PWM Signal Output When System Error Occurs 718

21.2.6 Write register software control PWM signal output 719
21.3 Register Description 720
21.3.1 EMB Control Register 1 (EMB_CTL1, Group0~5) 721
21.3.2 EMB Control Register 1 (EMB_CTL1, Group 6~7) 723
21.3.3 EMB Control Register 1 (EMB_CTL1, Group 8) 725
21.3.4 EMB Control Register 2 (EMB_CTL2, Group0~5) 727
21.3.5 EMB Control Register 2 (EMB_CTL2, Group 6~7) 729
21.3.6 EMB Control Register 2 (EMB_CTL2, Group 8) 731
21.3.7 EMB Software Output Enable Control Register (EMB _ SOE) 733
21.3.8 EMB Status Register (EMB _ STAT) 734
21.3.9 EMB State Reset Register (EMB _ STATCLR) 736
21.3.10 EMB Interrupt License Register (EMB _ INTEN) 738
21.3.11 EMB Control PWM Output Release Mode Selection Register (EMB_RLSSEL) 739
22 General Timer (TimerA) 740
22.1 Overview 740
22.2 Main Features 740
22.3 Functional Description 741
22.3.1 TimerA Block Diagram 741
22.3.2 Symmetrical Unit 742
22.3.3 Waveform Mode 742
22.3.4 Timer Selection 742
22.3.5 Comparison Output 743
22.3.6 Capture Input 744
22.3.7 Synchronous Startup 745
22.3.8 Digital Filtering 746
22.3.9 Cache Function 747
22.3.10 Cascade Count 748
22.3.11 PWM Output 749
22.3.12 Orthogonal Coding Count 750
22.4 Interrupt and Events 754
22.4.1 Compare Matched Interrupt and Events 754
22.4.2 Periodic Matching Interrupt and Events 755
22.5 Register Description 756
22.5.1 Generic Count Value Register (TMRA _ CNTER) 758
22.5.2 Periodic Reference Value Register (TMRA _ PERAR) 759
22.5.3 Comparison Reference Value Register (TMRA_CMPARm, m=1~4) 759
22.5.4 Control Status Register L (TMRA_BCSTRL) 760
22.5.5 Control Status Register H (TMRA_BCSTRH) 761

22.5.6 Interrupt Control Register (TMRA _ ICONR) 761
22.5.7 Event Control Register (TMRA _ ECONR) 762
22.5.8 Filter Control Register (TMRA _ FCONR) 763
22.5.9 Status Flag Register (TMRA _ STFLR) 764
22.5.10 Cache Control Register (TMRA_BCONRm, m=1~4) 765
22.5.11 Capture Control Register (TMRA _ CCONRm, m = 1 ~ 4) 766
22.5.12 Port Control Register (TMRA _ PCONRm, m = 1 ~ 4) 768
22.5.13 Hardware Trigger Event Selection Register (TMRA _ HCONR) 770
22.5.14 Hardware Added Event Selection Register (TMRA _ HCUPR) 772
22.5.15 Hardware Decrease Event Select Register (TMRA _ HCDOR) 773
23 General Timer (Timer0) 774
23.1 Overview 774
23.2 Functional Description 774
23.2.1 Timer0 Block Diagram 774
23.2.2 Timer Selection 775
23.2.3 Basic Count 775
23.2.4 Hardware Triggering 776
23.2.5 Software Synchronization Start 776
23.3 Interrupt and Events 776
23.3.1 Interrupt 776
23.3.2 Events 777
23.4 Register Description 778
23.4.1 Count Value Register (TMR0_CNTMR, m=A~B) 779
23.4.2 Reference Value Register (TMR0_CMPmR, m=A~B) 779
23.4.3 Basic Control Register (TMR0 _ BCONR) 780
23.4.4 Status Flag Register (TMR0 _ STFLR) 782
23.4.5 Count Synchronization Enable Register (TMR_SYNENR) 783
23.5 Precautions 785
24 Real Time Clock (RTC) 786
24.1 Introduction 786
24.2 Main Features 786
24.3 Functional Description 787
24.3.1 RTC Block Diagram 787
24.3.2 Power-on Setting 787
24.3.3 RTC Count Start Setting 787
24.3.4 System Low Power Mode Switching 788
24.3.5 Read Count Register 788
24.3.6 Write Count Register 788

24.3.7 Alarm Clock Setting 789
24.3.8 Clock Error Compensation 789
24.3.9 1Hz Output 789
24.4 Interrupt 791
24.4.1 Alarm Interrupt 791
24.4.2 Periodic Interrupt 791
24.5 Register Description 792
24.5.1 Control Register 0 (RTC_CR0) 793
24.5.2 Control Register 1 (RTC_CR1) 793
24.5.3 Control Register 2 (RTC_CR2) 794
24.5.4 Control Register 3 (RTC_CR3) 794
24.5.5 Second Count Register (RTC_SEC) 795
24.5.6 Minute Count Register (RTC_MIN) 795
24.5.7 Hour Count Register (RTC_HOUR) 796
24.5.8 Day Count Register (RTC_DAY) 798
24.5.9 Week Count Register (RTC_WEEK) 798
24.5.10 Month Count Register (RTC_MON) 799
24.5.11 Year Count Register (RTC_YEAR) 799
24.5.12 Minute Alarm Register (RTC_ALMMIN) 799
24.5.13 Hour Alarm register (RTC_ALMHOUR) 800
24.5.14 Week Alarm Register (RTC_ALMWEEK) 800
24.5.15 When Clock Error Compensation, Set The Counting Clock Error Compensation Registers RTC_ERRCRL, RTC_ERRCRH; 801
25 Watchdog Counter (WDT/SWDT) 803
25.1 Introduction 803
25.2 Functional Description 804
25.2.1 Start the Watchdog 804
25.2.2 Hardware Startup Mode 804
25.2.3 Software Startup Mode 805
25.2.4 Refresh Action 805
25.2.5 Flag Bit 806
25.2.6 Interrupt Reduction 806
25.2.7 Count Underflow 807
25.2.8 Refresh Error 807
25.3 Register Description 809
25.3.1 Control Registers (SWDT_CR, WDT_CR) 810
25.3.2 Status Register (SWDT_SR, WDT_SR) 812
25.3.3 Refresh Register (SWDT_RR, WDT_RR) 812

25.4 Precautions 813
26 Universal Synchronous/ Asynchronous Receiver Transmitter (USART) 814
26.1 Introduction 814
26.2 Main Features 814
26.3 Functional Description 816
26.3.1 USART System Block Diagram 816
26.3.2 UARTFunction 817
26.3.3 Multiprocessor Communication 829
26.3.4 UART-LIN 833
26.3.5 Smart Card 836
26.3.6 Clock Synchronization Mode 840
26.3.7 Digital Filtering Function 846
26.3.8 Interrupt 847
26.4 Register Description 848
26.4.1 Status Register (USART _ SR) 849
26.4.2 Data Register (USART_TDR) 852
26.4.3 Data Register (USART_RDR) 852
26.4.4 Baud Rate Register (USART _ BRR) 853
26.4.5 Control Register 1 (USART_CR1) 854
26.4.6 Control Register 2 (USART_CR2) 857
26.4.7 Control Register 3 (USART_CR3) 859
26.4.8 Pre-divider Register (USART _ PR) 861
26.4.9 LIN Baud Rate Measurement Count Register (USART_LBMC) 862
26.4.10 USART1 Filter Control Register (USART1_NFC) 862
26.5 Precautions 863
26.5.1 UART Considerations 863
26.5.2 Clock Synchronization Mode Considerations 863
26.5.3 Other Considerations 863
27 Inter-Integrated Circuit (I2C) 864
27.1 Overview 864
27.2 Main Features 864
27.3 Functional Description 865
27.3.1 I2C System Block Diagram 865
27.3.2 I2C protocol 867
27.3.3 Address Matching 875
27.3.4 SMBus Action 879
27.3.5 Reduction 881
27.3.6 Programmable Digital Filtering 881

27.4 Interrupt and Events 883
27.5 Application Examples 884
27.5.1 I2C Initialization Process 884
27.6 Register Description 885
27.6.1 I2C Control Register 1 (I2C_CR1) 886
27.6.2 I2C Control Register 2 (I2C_CR2) 888
27.6.3 I2C Control Register 1 (I2C_CR3) 889
27.6.4 I2C Control Register 4 (I2C_CR4) 890
27.6.5 I2C Slave Address Register 0 (I2C _ SLR0) 891
27.6.6 I2C Slave Address Register1 (I2C _ SLR 1) 892
27.6.7 I2C SCL Level When Timeout Control Register (I2C_SLTR) 893
27.6.8 I2C Status Register (I2C_SR) 894
27.6.9 I2C State Reset Register (I2C _ CLR) 898
27.6.10 I2C Data Sending Register (I2C _ DTR) 899
27.6.11 I2C Data Receiving Register (I2C_DRR) 900
27.6.12 I2C Data Shift Register (I2C _ DSR) 901
27.6.13 I2C Clock Control Register (I2C_CCR) 902
27.6.14 I2C Filter Control Register (I2C_FLTR) 903
27.6.15 I2C FIFO Controller Register (I2C_FSTR) 903
27.6.16 I2C Slave Address Receiver Register (I2C_SLVADRR) 904
28 Programmable Logic Array (PLA) 905
28.1 Overview 905
28.2 Functional Description 905
28.2.1 PLA Block Diagram 905
28.2.2 Action description 906
28.2.3 PLA Input Multiplexer 907
28.2.4 Look-up Table (LUT) Control 909
28.2.5 PLA Output 910
28.3 Register Description 911
28.3.1 Global Control Register (PLA_GCTL) 912
28.3.2 Status Register (PLA_STAT) 913
28.3.3 Multiplexer Selection Register (PLAx_MUXS, x=0~15) 914
28.3.4 Lookup Table Control Register (PLAx_LUTCTL, x = 0 ~ 15) 914
28.3.5 Control Register (PLAx_CTL, x = 0 ~ 15) 915
28.4 Precautions 916
29 Serial Peripheral Interface (SPI) 917
29.1 Introduction 917
29.2 SPI Main Characteristics 917

29.3 SPI Function Description 918
29.3.1 SPI System Block Diagram 918
29.3.2 SPI Action System Description 919
29.3.3 Data Communication Description 921
29.3.4 Operating Instructions 930
29.3.5 Self-Diagnosis of Parity 938
29.3.6 Error Detection 938
29.3.7 SPI Initialization 943
29.4 Interruptand Events 943
29.4.1 Interrupt 943
29.4.2 Events 944
29.5 Register Description 945
29.5.1 SPI Data Register (SPI _ DR) 946
29.5.2 SPI Control Register (SPI_CR) 947
29.5.3 SPI Communication Configuration Register1 (SPI _ CFG1) 948
29.5.4 SPI Status Register (SPI _ SR) 950
29.5.5 SPI Communication Configuration Register2 (SPI _ CFG2) 951
30 Controller Area Network (MCAN) 952
30.1 Overview 952
30.2 Main Features 952
30.3 Functional Description 953
30.3.1 MCAN System Block Diagram 953
30.3.2 Pin Description 954
30.3.3 Operating Mode 954
30.3.4 Timestamp Generation 963
30.3.5 Timeout Counter 964
30.3.6 Receiving Processing 964
30.3.7 Send Processing 972
30.3.8 FIFO Acknowledgement Processing 978
30.3.9 Message RAM 979
30.4 Register Description 988
30.4.1 MCAN Byte Order Register (MCAN_ENDN) 990
30.4.2 MCAN Data Bit Time and Prescaler Register (MCAN_DBTP) 991
30.4.3 MCAN Test Register (MCAN_TEST) 992
30.4.4 MCAN RAM Watchdog Register (MCAN_RWD) 993
30.4.5 MCAN CC Control Register (MCAN_CCCR) 994
30.4.6 MCAN Nominal Bit time and Prescaler Register (MCAN_NBTP) 996
30.4.7 MCAN Timestamp Counter Configuration Register (MCAN_TSCC) 997

30.4.8 MCAN Timestamp Counter Value Register (MCAN_TSCV) 998
30.4.9 MCAN Timeout Counter Configuration Register (MCAN_TOCC) 999
30.4.10 MCAN Timeout Counter Value Register (MCAN_TOCV) 1000
30.4.11 MCAN Error Counter Register (MCAN_ECR) 1001
30.4.12 MCAN Protocol Status Register (MCAN_PSR) 1002
30.4.13 MCAN Send Delay Compensation Register (MCAN_TDCR) 1004
30.4.14 MCAN Interrupt Register (MCAN_IR) 1005
30.4.15 MCAN Interrupt Enable Register (MCAN_IE) 1008
30.4.16 MCAN Interrupt Line Selection Register (MCAN_ILS) 1010
30.4.17 MCAN Interrupt Line Enable Register (MCAN_ILE) 1012
30.4.18 MCAN Global Filter Configuration Register (MCAN_GFC) 1013
30.4.19 MCAN Standard ID Filter Configuration Register (MCAN_SIDFC) 1014
30.4.20 MCAN Extended ID Filter Configuration Register (MCAN_XIDFC) 1015
30.4.21 MCAN Extended ID AND Mask Register (MCAN_XIDAM) 1016
30.4.22 MCAN High Priority Message Status Register (MCAN_HPMS) 1017
30.4.23 MCAN New Data 1 Register (MCAN_NDAT1) 1018
30.4.24 MCAN New Data 2 Register (MCAN_NDAT2) 1018
30.4.25 MCAN Rx FIFO0 Configuration Register (MCAN_RXF0C) 1019
30.4.26 MCAN Rx FIFO0 Status Register (MCAN_RXF0S) 1020
30.4.27 MCAN Rx FIFO0 Acknowledge Register (MCAN_RXF0A) 1021
30.4.28 MCAN Rx Buffer Configuration Register (MCAN_RXBC) 1021
30.4.29 MCAN Rx FIFO1 Configuration Register (MCAN_RXF1C) 1022
30.4.30 MCAN Rx FIFO1 Status Register (MCAN_RXF1S) 1023
30.4.31 MCAN Rx FIFO1 Acknowledge Register (MCAN_RXF1A) 1024
30.4.32 MCAN Rx Buffer and FIFO Element Size Configuration Register (MCAN_RXESC) 1025
30.4.33 MCAN Tx Buffer Configuration Register (MCAN_TXBC) 1026
30.4.34 MCAN Tx FIFO/Queue Status Register (MCAN_TXFQS) 1027
30.4.35 MCAN Tx Buffer Element Size Configuration Register (MCAN_TXESC) 1028
30.4.36 MCAN Transmit Buffer Request Pending Register (MCAN_TXBRP) 1029
30.4.37 MCAN Tx Buffer Add Request Register (MCAN_TXBAR) 1030
30.4.38 MCAN Tx Buffer Cancel Request Register (MCAN_TXBCR) 1031
30.4.39 MCAN Tx Buffer Send Occurred Register (MCAN_TXBTO) 1032
30.4.40 MCAN Tx Buffer Cancel Completed Register (MCAN_TXBCF) 1032
30.4.41 MCAN Tx Buffer Send Interrupt Enable Register (MCAN_TXBTIE) 1033
30.4.42 MCAN Tx Buffer Cancel Completed Interrupt Enable Register (MCAN_TXBCIE) 1033
30.4.43 MCAN Tx Event FIFO Configuration Register (MCAN_TXEFC) 1034
30.4.44 MCAN Tx Event FIFO Status Register (MCAN_TXEFS) 1035
30.4.45 MCAN Tx Event FIFO Acknowledge Register (MCAN_TXEFA) 1036

30.4.46 MCAN Message RAM Check Control Register (MCANRAM_CKCR) 1037
30.4.47 MCAN Message RAM Check Status Register (MCANRAM_CKSR) 1038
30.4.48 MCAN Message RAM Error Injection Enable Register (MCANRAM_EIEN) 1038
30.4.49 MCAN Message RAM Error Injection Bit Enable Register 0 (MCANRA_EIBIT0) 1038
30.4.50 MCAN Message RAM Error Injection Bit Enable Register 1 (MCANRAM_EIBIT1) 1039
30.4.51 MCAN Message RAM Check Error Address Register (MCANRAM_ECCERRADDR) 1039
30.5 Precautions 1040
30.5.1 CAN Usage Notes 1040
30.5.2 CAN Bus Anti-Jamming Measures 1040
30.5.3 CAN Controller Noise Constraints 1040
31 CRC Operation (CRC) 1041
31.1 Main Features 1041
31.2 Functional Description 1041
31.2.1 Functional Block Diagram 1041
31.2.2 CRC16 Encoding Mode 1041
31.2.3 CRC 16 Check Mode 1042
31.2.4 CRC32 Encoding Mode 1042
31.2.5 CRC 32 check mode 1042
31.3 Register Description 1043
31.3.1 Control Register (CRC _ CR) 1044
31.3.2 Results Register (CRC _ RESLT) 1044
31.3.3 Data Register (CRC _ DAT) 1044
32 Debug Controller (DBGC) 1045
32.1 Overview 1045
32.2 DBGC System Block Diagram 1045
32.3 SWJ-DP Debug Port (SWD and JTAG) 1046
32.3.1 Switching Mechanism of JTAG-DP or SW-DP 1047
32.4 Pinout and Debug Port Pinout 1047
32.4.1 SWJ Debug Port Pin 1047
32.4.2 Flexible SWJ-DP Pin Assignment 1047
32.4.3 Internal Pull-ups on JTAG Pins 1048
32.4.4 Use the Serial Interface and Release Unused Debug Pins for GPIO 1048
32.5 Register Description 1049
32.5.1 Chip Specific Identification Register (CHIPID) 1050
32.5.2 DBG State Register (MCUDBGSTAT) 1050
32.5.3 Peripheral Debugging Pause Register (MCUSTPCTL) 1051
32.5.4 Debug Component Configuration Register (MCUTRACECTL) 1052
32.5.5 Peripheral Debugging Pause Register 2 (MCUSTPCTL2) 1052

32.6 SW Debug Port 1053
32.6.1 Introduction to SW Agreement 1053
32.7 TPIU (Trace Port Interface Unit) 1053
32.7.1 Introduction 1053
32.7.2 TRACE Pin Allocation 1053
32.7.3 MCU Internal TRACECLKIN Connection 1055
32.7.4 TPIU Register 1055
32.7.5 TPIU Configuration Example 1055
Version Revision History 1056

Table Index

Table 1-1 Memory-Mapping 46
Table 1-2 Sample Destination Address Configuration 53
Table 1-3 MMF Base Address 54
Table 1-4 Register List 54
Table 3-1 Reset Methods and Generation Conditions 60
Table 3-2 Reset Methods and Reset Flags 61
Table 3-3 Reset Conditions for Each Module 70
Table 3-4 RMU Base Address 71
Table 3-5 RMU Register List 71
Table 4-1 Clock Source Main Features 82
Table 4-2 Specifications of Each Internal Clock 83
Table 4-3 CMU Register List 98
Table 5-1 BOR Configuration 124
Table 5-2 PVD1/PVD2 Features 125
Table 5-3 Operation Mode 130
Table 5-4 Low Power Modes 130
Table 5-5 Operating Conditions of Low Power Consumption Mode and Status of Each Module in Low Power Consumption Mode 131
Table 5-6 Power-down Mode Submodes 136
Table 5-7 RAM Module and RAM Power-down Control Bits 139
Table 5-8 Register Protection List 140
Table 5-9 PWC Register List 141
Table 6-1 ICG Base Address 162
Table 6-2 ICG Rgister List 162
Table 7-1 CPU Clock Frequency and Flash Read Wait Cycle Comparison Table 168
Table 7-2 Flash Actual Read Cycle Number 169
Table 7-3 OTP Address Distribution Table 176
Table 7-4 EFM Base Address 179
Table 7-5 EFM Register List 179
Table 8-1 SRAM Space Allocation 189
Table 8-2 SRAM Base Address 191
Table 8-3 SRAM Register List 191
Table 9-1 GPIO Base Address 206
Table 9-2 GPIO Register 206
Table 9-3 Event Port Base Address 206
Table 9-4 Event Port Register 206

Table 9-5 List of PORT Registers for 32bit Access 222
Table 10-1 External Interrupt Input Pin 223
Table 10-2 Interrupt Vector Table 225
Table 10-3 Interrupt Event Table 230
Table 10-4 INTC Base Address 250
Table 10-5 INTC Register List 250
Table 11-1 AOS Target List 265
Table 11-2 AOS Base Address 267
Table 11-3 AOS Register List 267
Table 12-1 MPU Base Address 282
Table 12-2 MPU Register List 282
Table 13-1 Measurement Error when HRC Target Frequency of 20MHz 293
Table 13-2 Measurement Error when HRC Target Frequency of 16MHz 296
Table 13-3 CTC Base Address 302
Table 13-4 CTC Register List 302
Table 14-1 Channel Reset Description 314
Table 14-2 DMA Base Address 322
Table 14-3 DMA Register List 322
Table 15-1 CMP Pin List 347
Table 15-2 CMP Analog Input List 348
Table 15-3 CMP Base Address 356
Table 15-4 CMP Register List 356
Table 15-5 Timer Window PWM List 361
Table 16-1 Specifications of each ADC unit 367
Table 16-2 Sequences A and B of Various Competitions 375
Table 16-3 AD Conversion Time 378
Table 16-4 ADC Base Address 388
Table 16-5 ADC Register List 388
Table 17-1 DAC Pinout 407
Table 17-2 DAC Base Address 410
Table 17-3 DAC Register List 410
Table 17-4 DAC Conversion and Analog Output Control 413
Table 18-1 Basic Functions and Features of Timer6 417
Table 18-2 Timer6 Port List 418
Table 18-3 Comparison of Functions in Different Modes 448
Table 18-4 Timer6 Base Address 460
Table 18-5 Timer6 Register List 460
Table 18-6 Timer6 Common Register List 461

Table 18-7 Counter (CNTER) Control Priority....491
Table 18-8 PWMA Port Output Control Priority....491
Table 18-9 PWMB Port Output Control Priority....492
Table 19-1 HRPWM Basic Functions and Features....495
Table 19-2 HRPWM Port List....497
Table 19-3 HRPWM Resolution....498
Table 19-4 Channel B Follow Function Exits and Enters Idle Time....515
Table 19-5 External Event Mapping Table....519
Table 19-6 External Event EEysrc2 Selection....520
Table 19-7 HRPWM<t>_EEFLTCR1<2>.EEyFM Corresponding Blanking Time Setting....523
Table 19-8 Filter Signal Mapping in Window Mode....524
Table 19-9 HRPWM<t>_EEFLTCR1<2>.EEyFM Corresponding Window Time Setting....525
Table 19-10 Supports Shadow Registers....529
Table 19-11 Shadow Transfer Points Supported by Each Unit....530
Table 19-12 Global ShadowStatus Flag Setting Conditions for Each Unit....535
Table 19-13 Comparison of Functions in Different Modes....552
Table 19-14 HRPWN Base Address....565
Table 19-15 HRPWM Register List....565
Table 19-16 HRPWM Common Register List....568
Table 19-17 Counter (CNTER) Control Priority....661
Table 19-18 PWMA, B Port Output Control Priority....661
Table 20-1 Basic Functions And Features Of Timer4....663
Table 20-2 Timer4 port list....665
Table 20-3 Timer4 Base Address....688
Table 20-4 Timer4 Register List....688
Table 20-5 OCCR*h Count Match List....697
Table 20-6 OCCR*h / Count Match List....698
Table 20-7 PWM port output status and register setting value....707
Table 21-1 Port Allocation Table....715
Table 21-2 Port Allocation Table....715
Table 21-3 EMB Group Comparison Table....717
Table 21-4 EMB Base Address....720
Table 21-5 EMB Register List....720
Table 22-1 Basic functions and features of TimerA....740
Table 22-2 TimerA Port List....741
Table 22-3 Internal Trigger Event TRGSEL Selection Relationship Correspondence Table....744
Table 22-4 TimerA Base Address....756
Table 22-5 TimerA Register List....756

Table 23-1 Timer0/TMR_SYNENR Base Address 778
Table 23-2 Timer0 Register List 778
Table 23-3 TMR_SYNENR Register Llist 778
Table 24-1 Basic Specifications of RTC 786
Table 24-2 RTC Base Address 792
Table 24-3 RTC Register List 792
Table 25-1 Basic Characteristics Of Watchdog Counter 803
Table 25-2 WDT/SWDT Base Address 809
Table 25-3 WDT Register List 809
Table 25-4 SWDT Register List 809
Table 26-1 USART Pin Descriptions 816
Table 26-2 Tolerance of the UART Receiver when DIV_Fraction is 0 822
Table 26-3 Tolerance of UART Receiver when DIV_Fraction is not 0 822
Table 26-4 UART Interrupt/Event Table 829
Table 26-5 Multi-Processor Mode Interrupt/Event 833
Table 26-6 LIN interrupt/Event Table 836
Table 26-7 Smart Card Mode Interrupt/Event 840
Table 26-8 Clock Synchronization Mode Interrupt/Event Table 846
Table 26-9 USART Overall Interrupt List 847
Table 26-10 USART Base Address 848
Table 26-11 USART Register List 848
Table 26-12 USART1_NFC Register List 848
Table 26-13 Baud Rate Calculation Formula (Invalid Fractional Baud Rate FBME=0) 853
Table 26-14 Baud Rate Calculation Formula Valid Fractional Baud Rate FBME=1) 853
Table 27-1 Input/Output Pin 866
Table 27-2 Interruption List 883
Table 27-3 Event Signal Output List 883
Table 27-4 I2C Base Address 885
Table 27-5 I2C Register List 885
Table 28-1 PLAxMUX0 Input Selection 907
Table 28-2 PLAxMUX1 Input Selection 908
Table 28-3 LUT Control 909
Table 28-4 PLA Output Port Correspondence Table 910
Table 28-5 PLA Base Address 911
Table 28-6 Register List 911
Table 29-1 Characteristics of SPI 917
Table 29-2 Pin Description 918
Table 29-3 SPI Pin Status Description in Master Mode 919

Table 29-4 SPI Pin Status Description in Slave Mode 919
Table 29-5 Some Setting Values Bit Speed 921
Table 29-6 SPI Pattern and Register Setting Relationship 930
Table 29-7 Correspondence Table for Error Detection 939
Table 29-8 SPI interrupt Source Description 944
Table 29-9 SPI Base Address 945
Table 29-10 SPI Register List 945
Table 30-1 MCAN Pin Description 954
Table 30-2 MCAN Configuration Register List 955
Table 30-3 DLC Encoding in MCAN 957
Table 30-4 Rx Buffer and FIFO Element Size 970
Table 30-5 Rx Buffer Filter Configuration Example 971
Table 30-6 Debug Message Filter Configuration Example 972
Table 30-7 Possible Configurations for Sending Frames 973
Table 30-8 Tx Buffer, FIFO (queue) Element Size 974
Table 30-9 Rx Buffer and FIFO Elements 980
Table 30-10 Rx Buffer and FIFO Elements Description 980
Table 30-11 Buffer Elements 982
Table 30-12 Tx Buffer Element Description 982
Table 30-13 Tx Event FIFO Elements 984
Table 30-14 Tx Event FIFO Element Description 984
Table 30-15 Standard Message ID Filter Elements 986
Table 30-16 Standard Message ID Filter Element Description 986
Table 30-17 Extended Message ID Filter Elements 987
Table 30-18 4Extended Message ID Filter Element Description 987
Table 30-19 MCAN/MCANRAM Base Address 988
Table 30-20 MCAN Register List 988
Table 30-21 MCANRAM Register List 989
Table 31-1 CRC Base Address 1043
Table 31-2 CRC Register List 1043
Table 32-1 SWJ Debug Port Pins 1047
Table 32-2 Flexible SWJ-DP Pin Assignment 1048
Table 32-3 DBGC Base Address 1049
Table 32-4 DBGC Register List 1049

Figure Index

Figure 2-1 Bus Architecture Diagram 57
Figure 3-1 Power-on Reset 62
Figure 3-2 NRST Reset Timing 63
Figure 3-3 Brownout Reset 63
Figure 3-4 Programmable Voltage Detection 1 Reset 64
Figure 3-5 Programmable Voltage Detection 2 Reset 64
Figure 3-6 Watchdog and Special Watchdog Reset 65
Figure 3-7 Power-down Wake-up Reset 66
Figure 3-8 Software Reset 66
Figure 3-9 MPU Error Reset 66
Figure 3-10 RAM Parity Reset 67
Figure 3-11 RAMECC Reset 67
Figure 3-12 Clock Frequency Abnormal Reset 68
Figure 3-13 External High-speed Oscillation Abnormal Reset 68
Figure 3-14 Cortex-M4 Lockup Reset 68
Figure 3-15 Module Function Reset 69
Figure 4-1 Clock System Block Diagram 80
Figure 4-2 Clock Frequency Measurement Block Diagram 81
Figure 4-3 External High-speed Oscillator Connection Example 84
Figure 4-4 Example of Connection Diagram of External Clock Input 85
Figure 4-5 Example of External High-speed Oscillator Fault Detection 86
Figure 4-6 XTAL is Selected as the System Clock, and XTAL Oscillation Fault is Detected 87
Figure 4-7 External Low-speed Oscillator Connection Example 89
Figure 4-8 Clock Source Switching Procedure 93
Figure 4-9 Clock Divider Switching 94
Figure 4-10 Clock Frequency Measurement Timing Diagram 96
Figure 5-1 Power Supply Diagram 123
Figure 5-2 Power-on Reset, Power-off Reset Waveforms 124
Figure 5-3 Brownout Reset Waveform 125
Figure 5-4 PVD1 Interrupt/Reset Block Diagram 126
Figure 5-5 PVD2 Interrupt/Reset Block Diagram 126
Figure 5-6 Voltage Detection 1 Interrupt Timing Diagram 127
Figure 5-7 Voltage Detection 1 Reset Timing Diagram 127
Figure 5-8 Voltage Detection 2 Interrupt Operation Sequence Diagram 128
Figure 5-9 Voltage Detection 2 Reset Operation Sequence Diagram 128
Figure 5-10 Internal Voltage Sampling Diagram 129

Figure 5-11 PTWKn Structure Block Diagram....138
Figure 7-1 Flash AddressAllocation....167
Figure 7-2 Start sector swap function 1....178
Figure 7-3 Start Swap Function 2....178
Figure 8-1 Schematic Diagram of Error Injection Function....190
Figure 9-1 Port Basic Structure Diagram....201
Figure 10-1 Interrupt System Block Diagram....224
Figure 10-2 Interrupt Request/Event Input Selection....247
Figure 10-3 Schematic diagram of digital filter operation....248
Figure 11-1 AOS Module Block Diagram....264
Figure 12-1 Only Allow Partial Space Access to Schematic Diagram....280
Figure 12-2 Schematic Diagram of Only Prohibiting Access to Part of Space....281
Figure 13-1 Basic Block Diagram of CTC....292
Figure 13-2 Schematic Diagram of CTC Calibration....299
Figure 14-1 DMA Structure Diagram....308
Figure 14-2 Chain Transmission diagram....312
Figure 14-3 Schematic Diagram of Discontinuous Address Transmission (Taking Source Address as an Example)....313
Figure 14-4 Schematic Diagram of Discontinuous Reset....315
Figure 14-5 Application Example 1: Memory-to-Memory Transfer....318
Figure 14-6 Application example 2: Transfer from memory to peripheral circuits....319
Figure 15-1 CMP Functional Block Diagram....347
Figure 15-2 Common Comparison Schematic Diagram....349
Figure 15-3 Diagram of Window Comparison....350
Figure 15-4 Input Scanning Schematic Diagram....351
Figure 15-5 Window Output Schematic 1....353
Figure 15-6 Window Output Schematic Diagram II....353
Figure 15-7 Window Output Schematic Diagram 3....354
Figure 16-1 ADC Block Diagram....367
Figure 16-2 Channel Mapping Diagram....369
Figure 16-3 Internal Analog Channel Selection....370
Figure 16-4 Sequence A Single Sweep Mode....371
Figure 16-5 Continuous Scanning....372
Figure 16-6 Data Buffering Mode....373
Figure 16-7 Double Sequence Scan Mode (Sequence A restarts from interrupted channel)....375
Figure 16-8 Double Sequence Scan Mode (Sequence A Restarts from the First Channel)....376
Figure 16-9 Analog Watchdog Protection Area (Compare Mode)....377
Figure 16-10 ADC Conversion Time....378

Figure 16-11 Conversion Action when Averaging Function is Valid 380
Figure 16-12 Single Parallel Trigger Mode (Three ADCs) 381
Figure 16-13 Single delay trigger mode (three ADCs) 383
Figure 16-14 Cyclic Parallel Trigger Mode (Three ADCs) 384
Figure 16-15 Cyclic Delay Trigger Mode (Two ADCs) 385
Figure 16-16 Cyclic Delay Trigger Mode (Three ADCs) 385
Figure 16-17 ADC interrupt and event output timing 387
Figure 17-1 DAC Conversion Channel Block Diagram 406
Figure 17-2 DAC Conversion Diagram 407
Figure 17-3 ADC conversion priority mode operation diagram 409
Figure 18-1 Timer6 Basic Block Diagram 418
Figure 18-2 Sawtooth Waveform (Counting Up) 419
Figure 18-3 Triangle Waveform 419
Figure 18-4 Compare Output Action 421
Figure 18-5 Capturing Input Actions 422
Figure 18-6 Hardware Refresh Action 422
Figure 18-7 Software Synchronization Action 424
Figure 18-8 Hardware Synchronous Action 426
Figure 18-9 Pulse Width Measurement 427
Figure 18-10 Cycle Measurement 427
Figure 18-11 Single-buffer Mode Comparison Output Timing 429
Figure 18-12 Double Buffer Capture Input Timing 430
Figure 18-13 Counting Buffer Action in Sawtooth Wave Mode 431
Figure 18-14 Count Buffer Action in Triangle Wave Mode 432
Figure 18-15 Count Buffer Action in Triangle Wave Mode 433
Figure 18-16 Filtering function of the capture input port 434
Figure 18-17 Single-edge Aligned Independent PWM 434
Figure 18-18 Bilateral Symmetrical Independent PWM 435
Figure 18-19 Software Setting of GCMBR Complementary PWM Wave Output 436
Figure 18-20 Hardware Setting GCMBR Complementary PWM Wave Output 437
Figure 18-21 Double-sided Asymmetric PWM Output 438
Figure 18-22 Phase Single Edge Aligned Independent PWM 439
Figure 18-23 4-phase Bilateral Symmetrical Complementary PWM with Dead Time 440
Figure 18-24 Cycle Interval Valid Request Signal Action 441
Figure 18-25 Position Mode - Basic Counting 442
Figure 18-26 Position Counting Mode - Phase Difference Counting (1 times counting) 443
Figure 18-27 Position Counting Mode - Phase Difference Counting (2 times counting) 443
Figure 18-28 Position Counting mode - Phase Difference Counting (4 times counting) 443

Figure 18-29 Position Counting Mode - Direction Counting....444
Figure 18-30 Revolution counting mode - Z phase counting....444
Figure 18-31 Revolution Counting mode - Position Overflow Counting....445
Figure 18-32 Revolution Counting Mode - Mixed Counting....445
Figure 18-33 Revolution Counting mode - Mixed Counting Z-phase Shielding Action Example 1....446
Figure 18-34 Revolution Counting Mode - Mixed Counting Z-phase Shielding Action Example 2....447
Figure 18-35 Example of Interrupt & Event Output in Sawtooth Wave Mode....451
Figure 19-1 HRPWM Basic Block Diagram....496
Figure 19-2 HRPWM Basic Block Diagram of Each Unit....497
Figure 19-3 Sawtooth Counter Mode....498
Figure 19-4 Triangle Counter Mode....498
Figure 19-5 Triangle counting direction....500
Figure 19-6 Capturing Input Actions....501
Figure 19-7 PWM Action Qualifer Module....501
Figure 19-8 Compare Output Action....502
Figure 19-9 Valid PWM OutputLegend in Half-wave Mode....503
Figure 19-10 Sawtooth Wave Duty Cycle Limit 1....503
Figure 19-11 Sawtooth Duty Cycle Limit 2....504
Figure 19-12 Triangle Wave Duty Cycle Limit 1....504
Figure 19-13 Dead Zone Module....505
Figure 19-14 Hardware Realizes Complementary PWM Wave Output....506
Figure 19-15 Waveform Output Under Dead Zone Mode when Pulse Width is Abnormal....506
Figure 19-16 Control Basic Block Diagram....507
Figure 19-17 Immediate Idle Entry and Exit....508
Figure 19-18 Delayed Idle Entry and Exit....509
Figure 19-19 Timing Diagram of Burst Mode Mode....511
Figure 19-20 Timing Diagram of Burst Mode Mode 2....511
Figure 19-21 Mode for Plot Burst Mode....512
Figure 19-22 Delay Mode Valid Burst Mode Enter Legend....513
Figure 19-23 Legend of State Switching During Delay Mode Valid Burst Mode....514
Figure 19-24 Exit of Burst Mode....515
Figure 19-25 Channel B Follow Function Exit and Enter Idle Legend....516
Figure 19-26 Triangle count mode Multi-phase PWM Output Slave Unit Startup and Clearing Action....517
Figure 19-27 Sawtooth Wave Mode Multi-phase PWMOutput Slave Unit Startup and Clearing Action....517
Figure 19-28 External Event Processing Block Diagram....518
Figure 19-29 Blanking Mode....521

Figure 19-30 Blanking Time 522
Figure 19-31 External Event Delay Function in Blanking Mode 522
Figure 19-32 Windowing Mode 524
Figure 19-33 Windowing Time 524
Figure 19-34 Software Synchronization Action 526
Figure 19-35 HRPWM_TRIGA Filtering Function 527
Figure 19-36 Abnormal Situation of Shadow Period Value of Triangle Wave Period Shadow Transmission time 532
Figure 19-37 Period Shadow Transfer Time 533
Figure 19-38 Shadow Action 1 in Triangle count mode 533
Figure 19-39 Shadow Action 2 in Triangle count mode 534
Figure 19-40 Burst Mode Shadow Action 534
Figure 19-41 Single Edge Aligned Independent High Resolution PWM 536
Figure 19-42 Bilateral Symmetrical Independent PWM 537
Figure 19-43 Software Realizes Complementary PWM Wave Output 538
Figure 19-44 Hardware Realizes Complementary PWM Wave Output 539
Figure 19-45 Double-sided Asymmetric PWM Output 540
Figure 19-46 6-Phase Single-side Aligned Independent PWM 541
Figure 19-47 3-phase Interleaved Complementary High-resolution PWM 543
Figure 19-48 Sawtooth Wave Mode Multi-phase Interleaved PWM Variable Frequency Output 544
Figure 19-49 Triangle count mode Multi-phase Interleaved PWM Variable Frequency Output 545
Figure 19-50 Cycle Interval Valid Request Signal Action 547
Figure 19-51 Output Management 549
Figure 19-52 Immediately Swap Output Waveform 549
Figure 19-53 Full period point Switching Output Waveform 550
Figure 19-54 DAC Synchronous Trigger Source 551
Figure 19-55 Unit 1~3 Special-compare Count Match Event 555
Figure 19-56 Example of Interrupt & Event Output in Sawtooth Wave Mode 556
Figure 20-1 Timer4 Basic Block Diagram 664
Figure 20-2 Timer4 Sawtooth Waveform 665
Figure 20-3 Timer4 Triangle Wave Waveform 665
Figure 20-4 Timer4 Sawtooth Wave Mode Counting Action 666
Figure 20-5 Timer4 Triangular Wave Mode Counting Action 666
Figure 20-6 Waveform Output Example in Sawtooth Mode 667
Figure 20-7 Triangle Wave Mode Waveform Output Example 668
Figure 20-8 Modify The Sawtooth Count Period When The Cache Is Invalid 669
Figure 20-9 Modify The Sawtooth Count Period When The Cache Is Enabled 670
Figure 20-10 Modify The Triangular Wave Count Period When The Cache Is Enabled 670

Figure 20-11 OCCR Buffer Data Transmission (When the Cycle Interval Response Link Disabled) ... 671
Figure 20-12 OCCR Buffer Data Transmission (Period Interval Response Link Enable) ... 672
Figure 20-13 Output Compare Buffer Data Transfer (OCMR Buffer Enabled) ... 673
Figure 20-14 SCCR Buffer Transmission Operation (Period Interval Response Link Transmission Disabled) ... 674
Figure 20-15 SCCR Buffer Transmission Operation (When Periodic Interval Response Link Transmission Is Enabled) ... 675
Figure 20-16 Example of Independent PWM Output of Sawtooth Wave ... 676
Figure 20-17 Example of Independent PWM Output of Triangle Wave ... 677
Figure 20-18 Sawtooth Wave Extended PWM Output ... 677
Figure 20-19 Software Complementary PWM Output ... 678
Figure 20-20 Complementary PWM Output In Dead-Band Timer Mode ... 679
Figure 20-21 Waveform Output In Dead-Band Timer Mode When The Pulse Width Is Abnormal ... 679
Figure 20-22 Complementary PWM Output in Dead-band Timer Filtering Mode ... 680
Figure 20-23 Cycle Interval Response Timing Diagram ... 681
Figure 20-24 Dedicated Event Output Signal Periodic Interval Response Output ... 682
Figure 20-25 PWM Period Output Monitoring Example ... 684
Figure 20-26 Dedicated Event Output TMR4_ADSM Timing in Compare Start Mode ... 685
Figure 20-27 Output Timing of Dedicated Event Output Signal in Delayed Start Mode ... 687
Figure 21-1 EMB Channel Functional Block Diagram ... 713
Figure 21-2 EMB Block Diagram ... 714
Figure 22-1 TimerA Basic Block Diagram ... 741
Figure 22-2 Sawtooth Waveform (Counting Up) ... 742
Figure 22-3 Triangle Waveform ... 742
Figure 22-4 Compare Output Action ... 743
Figure 22-5 Capturing Input Actions ... 745
Figure 22-6 Software Synchronization Action ... 746
Figure 22-7 Filtering function of the clock input port ... 747
Figure 22-8 Cache Action in Sawtooth Wave Mode ... 748
Figure 22-9 Cache and Special Cache Actions in Sawtooth Mode ... 748
Figure 22-10 32-Bit Cascade Counting Action ... 749
Figure 22-11 Single-edge Aligned PWM Output Example ... 749
Figure 22-12 Example of Bilateral Symmetrical PWM Output ... 750
Figure 22-13 Position Mode - Basic Counting ... 751
Figure 22-14 Position counting mode - phase difference counting (1 times counting) ... 751
Figure 22-15 Position Counting Mode - Phase Difference Counting (2 Times Counting) ... 752
Figure 22-16 Position Counting Mode - Phase Difference Counting (4 Times Counting) ... 752
Figure 22-17 Position Counting Mode - Direction Counting ... 752

Figure 22-18 Revolution Counting Mode - Z Phase Counting 753
Figure 22-19 Revolution Counting Mode - Position Overflow Counting 754
Figure 22-20 Revolution Counting Mode - Mixed Counting 754
Figure 23-1 Timer0 Basic Block Diagram 774
Figure 23-2 Timer0 Counting Timing Diagram 775
Figure 24-1 Basic block diagram of RTC 787
Figure 25-1 Hardware Startup Example 804
Figure 25-2 Software Startup Example 805
Figure 25-3 Examples of Refresh Action Timing (Action Confirmation, Refresh Request Signal Falling Edge, etc.) 806
Figure 25-4 Counter Underflow Action Example 807
Figure 25-5 Counter Refresh Action Example 808
Figure 26-1 USART System Block Diagram 816
Figure 26-2 UART data format 818
Figure 26-3 UART Send Data Legend 1 820
Figure 26-4 UART Send Data Legend 2 820
Figure 26-5 UART Internal Synchronization And Sampling Timing 821
Figure 26-6 UART Received Data Legend 1 823
Figure 26-7 UART Received Data Legend 2 823
Figure 26-8 Half-duplex Working Sequence 827
Figure 26-9 over8=1 DE assertion 827
Figure 26-10 over8=0 DE assertion 828
Figure 26-11 over8=1 DE De-asserts 828
Figure 26-12 over8=0 DE de-asserts 828
Figure 26-13 Multiprocessor Communication Diagrams 829
Figure 26-14 Multiprocessor Mode Data Format 830
Figure 26-15 Diagram of Multi-Processor Mode Send Data 831
Figure 26-16 Diagram of Multiprocessor Mode Receive Data 1 832
Figure 26-17 Diagram of Multiprocessor Mode Receive Data 2 833
Figure 26-18 LIN Bus Data Behavior 833
Figure 26-19 Smart card Pattern connected legend 836
Figure 26-20 Smart Card Pattern Synchronization and Sampling Timing Figure 837
Figure 26-21 Smart Card ModeData Format 838
Figure 26-22 Smart Card Pattern Sending Data Legend 839
Figure 26-23 Smart Card Pattern Receive Data Legend 840
Figure 26-24 Clock Synchronization Mode Data Format 841
Figure 26-25 Diagram of Clock Synchronization Mode Sending Data 1 843
Figure 26-26 Clock Synchronization Mode Sending Data Legend 2 843

Figure 26-27 Diagram of Clock Synchronization Mode Receive Data 1 844
Figure 26-28 Diagram of Clock Synchronization Mode Receive Data 2 845
Figure 27-1 I2C System Block Diagram 865
Figure 27-2 Example of I2C Bus Structure 866
Figure 27-3 I2C Bus Timing Diagram 867
Figure 27-4 I2C Data Format of I²C bus 868
Figure 27-5 Sequence Diagram of Master Sending Data in 7-bit Address Format (Example) 869
Figure 27-6 Sequence Diagram of Received Data for Master in 7-bit Address Format (Example) 870
Figure 27-7 Sequence Diagram of Slave Send Mode in 7-bit Address Format (Example) 871
Figure 27-8 Sequence Diagram of 7-bit Address Format Slave Receive Mode (Example) 871
Figure 27-9 SSCL Synchronization Time Series 872
Figure 27-10 Sending Sequence Diagram of Slave (1) 873
Figure 27-11 Sending Sequence Diagram of Slave (2) 874
Figure 27-12 Slave Receiving Timing Diagram 874
Figure 27-13 Fast ACK/NACK Timing Diagram 875
Figure 27-14 Timing when Selecting 7-bit Address Format 876
Figure 27-15 Timing when Selecting 10-Bit Address Format 877
Figure 27-16 Broadcast Address Matching Timing Diagram 877
Figure 27-17 SMBus Master Address Matching Timing Diagram 878
Figure 27-18 SMBus Alarm Response Address Matching Timing Diagram 879
Figure 27-19 SMBus Default Address Timing Diagram 879
Figure 27-20 Digital Filtering Circuit Block Diagram 882
Figure 28-1 PLA Module Block Diagram 905
Figure 28-2 PLA Unit Block Diagram 906
Figure 29-1 System Block Diagram 918
Figure 29-2 Master Mode Structure 920
Figure 29-3 Three-wire clock synchronization operation 921
Figure 29-4 Data Format 922
Figure 29-5 MSB sent First, Parity Check Invalid 923
Figure 29-6 LSB sent First, Parity Check Invalid 923
Figure 29-7 MSB Sent First, When Parity Check Valid 924
Figure 29-8 LSB Sent First, When Parity Check Valid 924
Figure 29-9 Data Transfer Format Diagram (CPHA = 0) 925
Figure 29-10 Data Transfer Format (CPHA = 1) 926
Figure 29-11 Full Duplex Synchronous Serial Communication 927
Figure 29-12 Sending Communication Only 928
Figure 29-13 Universal Communication 929
Figure 29-14 Continuous Communication 929

Figure 29-15 Parity Checking Process 938
Figure 29-16 Overload Error Handling 940
Figure 29-17 Schematic diagram of the action when the clock automatic stop function is enabled (CPHA=1) 941
Figure 29-18 Schematic diagram of the action when the clock automatic stop function is enabled (CPHA=0) 941
Figure 29-19 Parity Error 942
Figure 30-1 MCAN System Block Diagram 953
Figure 30-2 Transceiver Delay Measurement 959
Figure 30-3 Pin control in bus monitoring mode 961
Figure 30-4 Pin Control of Loopback Mode 963
Figure 30-5 Standard Message ID Filter Path 967
Figure 30-6 Extended Message ID Filter Path 968
Figure 30-7 Rx FIFO Status 969
Figure 30-8 Rx FIFO Overflow Processing 971
Figure 30-9 Example of Mixed Configuration of Dedicated Tx Buffer and Tx FIFO 976
Figure 30-10 Example of Mixed Configuration of Dedicated Tx Buffer and Send Queue 977
Figure 30-11 Message RAM Configuration 979
Figure 31-1 CRC Application Diagram 1041
Figure 32-1 Commissioning Control System 1045
Figure 32-2 Commissioning Control System 1046
Figure 32-3 JTAG-DP to SW-DP Switching Timing 1047
Figure 32-4 TPIU Block Diagram 1053

Document Conventions

Register-Related Abbreviations

Abbreviations used in Register Descriptions in this manual.

Abbreviation	Chinese Interpretation
RW	Read and write, software can read and write this bit.
R	Read only, , the software can only read this bit.
W	Write only, software can only write to this bit. The reading is indeterminate.
RW0	Software can read and write this bit, Writing 0 is valid and writing 1 has no effect.
RW1	Software can read and write this bit, Writing 1 is valid and writing 0 has no effect.
RC	Read only, this bit is automatically cleared after a read operation.
RWA	Readable, writable in unlocked or safe mode.
RW1A	Readable, write 1 in unlocked or secure mode, write 0 without restriction.
RWC	Readable, write any value to clear this bit.
RW1C	Readable, write 1 to clear this bit.
RS	Read only, this bit is automatically set after a read operation.
Res	Reserved Bit, must remain at reset value.

Glossary

Definitions of Technical Terms and Abbreviations used in this Manual.

Technical Terms/Abbreviations	Full Name in English	Chinese Interpretation
Byte	-	Byte, 8 -bit data length.
Half word	-	Halfword, 16 -bit data length.
Word	-	Word, 32-bit data length.
Double word	-	Double word , 64 -bit data length.
AHB	Advanced High-performance Bus	Advanced High Performance Bus.
APB	Advanced Peripheral Bus	High-speed Peripheral Bus.
DMA	Direct Memory Access	Direct Memory Access.
.....	.....	.....

About this Manual

This manual mainly introduces the function, operation and usage of the chip. For information on the availability and quantity of peripherals for each chip model, as well as specifications such as peripheral interfaces, electrical features, and pin packaging, please refer to the corresponding "Datasheet".

Overview

The HC32F336 series are high-performance MCU based on ARM® Cortex®-M4 32-bit RISC CPU with a maximum operating frequency of 150 MHz. The Cortex-M4 core integrates a floating-point arithmetic unit (FPU) and a DSP to implement single-precision floating-point arithmetic operations, supports all ARM single-precision data processing instructions and data types, and supports the full DSP instruction set. The kernel integrates the MPU unit and the DMAC dedicated MPU unit to ensure the safety of system operation.

The HC32F336 series integrates high-speed on-chip memory, including up to 128KB of Flash and up to 36KB of SRAM. Integrated Flash access acceleration unit to achieve single cycle program execution of the CPU on Flash. The polled bus matrix supports multiple bus hosts to access memory and peripherals simultaneously, improving performance. The bus hosts include CPU, DMA dedicated DMA, etc. In addition to the bus matrix, it supports data transfer between peripherals, basic arithmetic operations and mutual triggering of events, which can significantly reduce the transaction processing load of the CPU.

The HC32F336 series integrates rich peripheral functions , including 3 independent 12bit 2.5MSPS ADCs; 3 12bit 15MSPS DACs; The HC32F336 series integrates rich peripheral functions. 4 multi-function PWM Timers (Timer6), supporting orthogonal encoding input and 8-channel complementary PWM output; 6 22-bit high-precision PWM (HRPWM), supporting 12-channel 104 ps high-resolution PWM waveform output; 1 motor PWM Timer (Timer4), supporting 8-channel complementary PWM output; 4 16-bit universal timers (TimerA) and 1 32-bit universal timer (TimerA), supporting orthogonal encoding input and 20 channels of duty cycle adjustable PWM output; 6 serial communication interfaces (I2C/UART/SPI); 2-way MCAN controllers.

The HC32F336 series supports a wide voltage range (1.8~3.63V), a wide operating temperature range (-40~105℃) and various low power modes.

Typical Application

The HC32F336 series provides 64pin and 48pin LQFP packages, and can be used in AC/DC, DC/DC digital power applications, such as communications and server power supplies, brick power supplies, micro-inverters, charging pile DC/DC, high-performance frequency conversion control and other fields.

1 Memory Mapping

1.1 Memory mapping

The MCU supports 4 GB of linear address space from 0x0000 0000 to 0xFFFF FFFF, which contains programs and data. Refer to the following table for more memory mapping.

Table 1-1 Memory-Mapping

Memory Classification * 2		Start Address*1	End Address*1	Space Size	Module * 3	Protection * 4	Description
System	-	0xE010 0000	0xFFFF FFFF	511MB	Reserved	-	Custom space
	Private Peripheral External Bus	0xE00F F000	0xE00F FFFF	4KB	ROMTABLE	-	-
		0xE004 2400	0xE00F EFFF	755KB	BLANK	-	-
		0xE004 2000	0xE004 23FF	1KB	DBGC	-	Debug Control Register Area
		0xE004 1000	0xE004 1FFF	4KB	BLANK	-	-
		0xE004 0000	0xE004 0FFF	4KB	TPIU	-	-
	Private Peripheral Internal Bus	0xE000 F000	0xE003 FFFF	196KB	BLANK	-	-
		0xE000 E000	0xE000 EFFF	4KB	SCS	-	System control space NVIC/MPU, etc.
		0xE000 3000	0xE000 DFFF	44KB	BLANK	-	-
		0xE000 2000	0xE000 2FFF	4KB	FPB	-	-
		0xE000 1000	0xE000 1FFF	4KB	DWT	-	-
		0xE000 0000	0xE000 0FFF	4KB	ITM	-	-
External equipment	-	0xA000 0000	0xDFFF FFFF	1024MB	Reserved	-	-
External memory	AHB5 Clock: HCLK	0x9800 0000	0x9FFF FFFF	128MB	Reserved	-	-
		0x8820 0000	0x97FF FFFF	254MB	Reserved	-	-
	AHB5	0x8810 0000	0x881F FFFF	1MB	Reserved	-	-
		0x8800 0800	0x880F FFFF	1022KB	Reserved	-	-
		0x8800 0400	0x8800 07FF	1KB	Reserved	-	-
		0x8800 0000	0x8800 03FF	1KB	Reserved	-	-
		0x8000 0000	0x87FF FFFF	128MB	Reserved	-	-
		0x6000 0000	0x7FFF FFFF	512MB	Reserved	-	-
Peripherals	-	0x4400 0000	0x5FFF FFFF	448MB	Reserved	-	-
		0x4200 0000	0x43FF FFFF	32MB	PeriBitBand	-	-

Memory Classification * 2		Start Address*1	End Address*1	Space Size	Module * 3	Protection * 4	Description
		0x4010 0000	0x41FF FFFF	31MB	Reserved	-	-
		0x400C 0000	0x400F FFFF	256KB	Reserved	-	-
		0x4008 0000	0x400B FFFF	256KB	Reserved	-	-
	AHB2 Clock: PCLK0	0x4007 8800	0x4007 FFFF	30KB	Reserved	-	-
		0x4007 8400	0x4007 87FF	1KB	Reserved	-	-
		0x4007 8000	0x4007 83FF	1KB	Reserved	-	-
		0x4007 3840	0x4007 7FFF	18KB	BLANK	-	-
		0x4007 3800	0x4007 383F	64B	CANRAM_ECC	-	-
		0x4007 3000	0x4007 37FF	2KB	BLANK	-	-
		0x4007 2800	0x4007 2FFF	2KB	CAN_RAM	-	-
	AHB3 Clock: PCLK0	0x4007 1800	0x4007 27FF	4KB	BLANK	-	-
		0x4007 1600	0x4007 17FF	512B	BLANK		
		0x4007 1400	0x4007 15FF	512B	MCAN_2	-	-
		0x4007 1200	0x4007 13FF	512B	BLANK		
		0x4007 1000	0x4007 11FF	512B	MCAN_1	-	-
		0x4006 0000	0x4007 0FFF	68KB	BLANK	-	-
Peripherals	AHB1 Clock: HCLK	0x4005 9000	0x4005 FFFF	28KB	BLANK	-	-
		0x4005 8C00	0x4005 8FFF	1KB	BLANK	-	-
		0x4005 8800	0x4005 8BFF	1KB	BLANK	-	-
		0x4005 8400	0x4005 87FF	1KB	BLANK	-	-
		0x4005 8000	0x4005 83FF	1KB	BLANK	-	-
		0x4005 7C00	0x4005 7FFF	1KB	BLANK	-	-
		0x4005 7800	0x4005 7BFF	1KB	BLANK	-	-
		0x4005 7400	0x4005 77FF	1KB	BLANK	-	-
		0x4005 7000	0x4005 73FF	1KB	BLANK	-	-
		0x4005 6C00	0x4005 6FFF	1KB	BLANK	-	-
		0x4005 6800	0x4005 6BFF	1KB	BLANK	-	-
		0x4005 6400	0x4005 67FF	1KB	BLANK	-	-
		0x4005 6000	0x4005 63FF	1KB	BLANK	-	-
		0x4005 5C00	0x4005 5FFF	1KB	BLANK	-	-
		0x4005 5800	0x4005 5BFF	1KB	BLANK	-	-

Memory Classification * 2		Start Address*1	End Address*1	Space Size	Module * 3	Protection * 4	Description
		0x4005 5400	0x4005 57FF	1KB	PERIC	-	Peripheral Module Control Register
		0x4005 5000	0x4005 53FF	1KB	BLANK	-	-
		0x4005 4400	0x4005 4FFF	3KB	BLANK	-	Access Denied
		0x4005 4000	0x4005 43FF	1KB	CMU	With protection	-
		0x4005 3800	0x4005 3FFF	2KB	GPIO	-	-
		0x4005 3400	0x4005 37FF	1KB	BLANK	-	-
		0x4005 3000	0x4005 33FF	1KB	DMA	-	-
		0x4005 2000	0x4005 2FFF	4KB	BLANK	-	-
		0x4005 1000	0x4005 1FFF	4KB	INTC	With protection	-
		0x4005 0C00	0x4005 0FFF	1KB	BLANK	-	-
Peripherals		0x4005 0800	0x4005 0BFF	1KB	RAMIF	with protection	-
		0x4005 0400	0x4005 07FF	1KB	BLANK	-	-
		0x4005 0000	0x4005 03FF	1KB	DMPU	With protection	-
	APB4 Clock: PCLK3	0x4004 FC00	0x4004 FFFF	1KB	PLA	-	-
		0x4004 F800	0x4004 FBFF	1KB	BLANK	-	-
		0x4004 F400	0x4004 F7FF	1KB	BLANK	-	-
		0x4004 F000	0x4004 F3FF	1KB	BLANK	-	-
		0x4004 EC00	0x4004 EFFF	1KB	BLANK	-	-
		0x4004 E800	0x4004 EBFF	1KB	BLANK	-	-
		0x4004 E400	0x4004 E7FF	1KB	BLANK	-	-
		0x4004 E000	0x4004 E3FF	1KB	BLANK	-	-
		0x4004 D000	0x4004 DFFF	1KB	BLANK	-	-
		0x4004 CC00	0x4004 CFFF	1KB	PWC, CMU	With protection	-
		0x4004 C800	0x4004 CBFF	1KB	BLANK	-	-
		0x4004 C400	0x4004 C7FF	1KB	WKTMR	With protection	-
		0x4004 C000	0x4004 C3FF	1KB	RTC	With protection	-
		0x4004 AC00	0x4004 BFFF	5KB	BLANK	-	-
		0x4004 A800	0x4004 ABFF	1KB	BLANK	-	-
		0x4004 A400	0x4004 A7FF	1KB	BLANK	-	-
		0x4004 A000	0x4004 A3FF	1KB	BLANK	-	-

Memory Classification * 2		Start Address*1	End Address*1	Space Size	Module * 3	Protection * 4	Description
		0x4004 9C00	0x4004 9FFF	1KB	CTC	-	-
		0x4004 9800	0x4004 9BFF	1KB	BLANK	-	-
		0x4004 9400	0x4004 97FF	1KB	SWDT	With protection	-
		0x4004 9000	0x4004 93FF	1KB	WDT	With protection	-
		0x4004 8800	0x4004 8FFF	2KB	BLANK	-	-
		0x4004 8400	0x4004 87FF	1KB	FCM	-	-
		0x4004 8000	0x4004 83FF	1KB	FCG, CMU	With protection	-
Peripherals	APB3 Clock: PCLK4	0x4004 2400	0x4004 7FFF	23KB	BLANK	-	-
		0x4004 2000	0x4004 23FF	1KB	BLANK	-	-
		0x4004 1C00	0x4004 1FFF	1KB	BLANK	-	-
		0x4004 1800	0x4004 1BFF	1KB	BLANK	-	-
		0x4004 1400	0x4004 17FF	1KB	DAC_2	-	-
		0x4004 1000	0x4004 13FF	1KB	DAC_1	-	-
		0x4004 0C00	0x4004 0FFF	1KB	BLANK	-	-
		0x4004 0800	0x4004 0BFF	1KB	ADC_3	-	-
		0x4004 0400	0x4004 07FF	1KB	ADC_2	-	-
		0x4004 0000	0x4004 03FF	1KB	ADC_1	-	-
Peripherals	APB5 Clock: PCLK0	0x4003 C000	0x4003 FFFF	16KB	HRPWM	-	-
		0x4003 BC00	0x4003 BFFF	1KB	BLANK	-	-
		0x4003 B800	0x4003 BBFF	1KB	BLANK	-	-
		0x4003 B400	0x4003 B7FF	1KB	I2C	-	-
		0x4003 B000	0x4003 B3FF	1KB	CMP_3	-	-
		0x4003 AC00	0x4003 AFFF	1KB	TimerA_4	-	-
		0x4003 A800	0x4003 ABFF	1KB	TimerA_3	-	-
		0x4003 A400	0x4003 A7FF	1KB	TimerA_2	-	-
		0x4003 A000	0x4003 A3FF	1KB	TimerA_1	-	-
		0x4003 9000	0x4003 9FFF	4KB	Timer6	-	-
		0x4003 8800	0x4003 8EFF	2KB	BLANK	-	-
		0x4003 8600	0x4003 87FF	512B	EMB	-	-
		0x4003 8400	0x4003 85FF	1KB	CMP_12	-	-
		0x4003 8000	0x4003 83FF	1KB	Timer4	-	-

Memory Classification * 2		Start Address*1	End Address*1	Space Size	Module * 3	Protection * 4	Description
	APB2 Clock: PCLK1	0x4002 B800	0x4003 7FFF	50KB	BLANK	-	-
		0x4002 B000	0x4002 B7FF	2KB	BLANK	-	-
		0x4002 9800	0x4002 AFFF	6KB	BLANK	-	-
		0x4002 9400	0x4002 97FF	1KB	BLANK	-	-
		0x4002 9000	0x4002 93FF	1KB	BLANK	-	-
		0x4002 8800	0x4002 8FFF	2KB	BLANK	-	-
		0x4002 8400	0x4002 87FF	1KB	BLANK	-	-
		0x4002 8000	0x4002 83FF	1KB	BLANK	-	-
		0x4002 7C00	0x4002 7FFF	1KB	BLANK	-	-
		0x4002 7800	0x4002 7BFF	1KB	BLANK	-	-
		0x4002 7400	0x4002 77FF	1KB	BLANK	-	-
		0x4002 7000	0x4002 73FF	1KB	BLANK	-	-
		0x4002 6C00	0x4002 6FFF	1KB	BLANK	-	-
		0x4002 6800	0x4002 6BFF	1KB	BLANK	-	-
		0x4002 6400	0x4002 67FF	1KB	BLANK	-	-
		0x4002 6000	0x4002 63FF	1KB	TimerA_5	-	-
		0x4002 5800	0x4002 5FFF	2KB	BLANK	-	-
		0x4002 5400	0x4002 57FF	1KB	BLANK	-	-
		0x4002 5000	0x4002 53FF	1KB	BLANK	-	-
		0x4002 4C00	0x4002 4FFF	1KB	BLANK	-	-
		0x4002 4800	0x4002 4BFF	1KB	BLANK	-	-
		0x4002 4400	0x4002 47FF	1KB	Timer0_2	-	-
		0x4002 4000	0x4002 43FF	1KB	Timer0_1	-	-
		0x4002 2800	0x4002 3FFF	6KB	BLANK	-	-
		0x4002 2400	0x4002 27FF	1KB	BLANK	-	-
Peripherals		0x4002 2000	0x4002 23FF	1KB	BLANK	-	-
		0x4002 1C00	0x4002 1FFF	1KB	BLANK	-	-
		0x4002 1800	0x4002 1BFF	1KB	BLANK	-	-
		0x4002 1400	0x4002 17FF	1KB	BLANK	-	-
		0x4002 1000	0x4002 13FF	1KB	BLANK	-	-
		0x4002 0C00	0x4002 0FFF	1KB	BLANK	-	-
		0x4002 0800	0x4002 0BFF	1KB	BLANK	-	-

Memory Classification * 2		Start Address*1	End Address*1	Space Size	Module * 3	Protection * 4	Description
		0x40020400	0x400207FF	1KB	BLANK	-	-
		0x40020000	0x400203FF	1KB	BLANK	-	-
	APB1 Clock: PCLK1	0x4001E800	0x4001 FFFF	6KB	BLANK	-	-
		0x4001E400	0x4001E7FF	1KB	BLANK	-	-
		0x4001E000	0x4001E3FF	1KB	BLANK	-	-
		0x4001DC00	0x4001DFFF	1KB	BLANK	-	-
		0x4001D800	0x4001DBFF	1KB	USART_4	-	-
		0x4001D400	0x4001D7FF	1KB	USART_3	-	-
		0x4001D000	0x4001D3FF	1KB	USART_2	-	-
		0x4001CC00	0x4001CFFF	1KB	USART_1	-	-
		0x4001C800	0x4001CBFF	1KB	BLANK	-	-
		0x4001C400	0x4001C7FF	1KB	BLANK	-	-
		0x4001C000	0x4001C3FF	1KB	SPI	-	-
		0x40018000	0x40018FFF	16KB	BLANK	-	-
		0x40017C00	0x40017FFF	1KB	EMB	-	-
		0x40010C00	0x40017BFF	28KB	BLANK	-	-
		0x40010800	0x40010BFF	1KB	AOS	-	Internal trigger event register area
		0x40010400	0x400107FF	1KB	EFM	with protection	-
		0x40010000	0x400103FF	1KB	BLANK	-	-
	AHB4 Clock: PCLK1	0x40009400	0x4000 FFFF	27KB	BLANK	-	-
		0x40009000	0x400093FF	1KB	BLANK	-	-
		0x40008C00	0x40008FFF	1KB	CRC	-	-
		0x40008800	0x40008BFF	1KB	BLANK	-	-
		0x40008400	0x400087FF	1KB	BLANK	-	-
		0x40008000	0x400083FF	1KB	BLANK	-	-
		0x40000000	0x40007FFF	32KB	Reserved	-	-
SRAM	SRAM Clock: HCLK	0x24000000	0x3FFF FFFF	448MB	Reserved	-	-
		0x22000000	0x23FF FFFF	32MB	SRAMBitBand	-	-
		0x20100000	0x21FF FFFF	31MB	Reserved	-	-
		0x200F1000	0x200F FFFF	60KB	Reserved	-	-
		0x200F0000	0x200F 0FFF	4KB	Ret SRAM	With protection	-

Memory Classification * 2		Start Address*1	End Address*1	Space Size	Module * 3	Protection * 4	Description
		0x2006 0000	0x200E FFFF	576KB	Reserved	-	-
		0x2004 0000	0x2005 FFFF	128KB	Reserved	-	-
		0x2000 3000	0x2003 FFFF	240KB	Reserved	-	-
		0x2000 0000	0x2000 3FFF	16KB	SRAM0	-	ECC RAM
CODE	SRAM Clock: HCLK	0x1FFF C000	0x1FFF FFFF	16KB	SRAMH	-	ECC RAM
		0x0300 4000	0x1FFF BFFF	471.94MB	BLANK	-	-
	OTP, Flash Clock: HCLK	0x0300 204C	0x0300 3FFF	8116B	BLANK	-	-
		0x0300 2040	0x0300 204B	12B	Data Security Protection	-	Used to configure data security protection
		0x0300 2004	0x0300 203F	60B	BLANK	-	-
		0x0300 2000	0x0300 2003	4B	Boot swap	-	-
		0x0300 1000	0x0300 1FFF	4KB	BLANK	-	-
		0x0300 0A00	0x0300 0FFF	1.5KB	OTP	-	-
		0x0300 0000	0x0300 09FF	2.5KB	BLANK	-	-
	-	0x0210 0000	0x02FF FFFF	15M	BLANK	-	-
	REMAP Clock: HCLK	0x020A 0000	0x020F FFFF	384KB	BLANK	-	-
		0x0208 0000	0x0209 FFFF	128KB	REMAP1	-	Address Remapping Area 1
		0x0202 0000	0x0207 FFFF	384KB	BLANK	-	-
		0x0200 0000	0x0201 FFFF	512KB	REMAP0	-	Address Remapping Area 0
	-	0x0008 0000	0x01FF FFFF	31.5M	BLANK	-	-
	Flash Clock: HCLK	0x0002 0000	0x0007 FFFF	384KB	BLANK	-	-
		0x0000 0000	0x0001 FFFF	128KB	Embedded Flash 0	-	-

Note:

1. *Please refer to ARM Cortex-M4 Processor Technical Reference Manual Revision:r0p1。*
2. Please refer to the bus section *[Bus Architecture (BUS)]* for bus description.
3. *Reserved: Access to the bus will cause a bus error; BLANK: Write access is invalid, and read 0 during read access.*
4. *For modules with protection function, only CPU privileged mode access is supported when the protection function is valid.* For specific registers and descriptions, refer to the *[Memory Protection Unit (MPU)]* chapter.

1.2 Bit band region

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

In this MCU, both the peripheral register and the SRAM are mapped to a bit band area, which enables read and write operations of a single bit band. These operations are only available for Cortex-M4 access and are not valid for other bus interfaces such as DMA.

1.3 Address Remapping

This MCU provides 2 remapping addresses, and the memory address remapping function can be configured. The source address can be set as the main flash memory Flash address and the high-speed SRAM address.

Remap address 0:

0x0200 0000~0x0208 0000 (depending on the remapping space MMF_REMCR0/1.RMSIZE[4:0])

Remap address 1:

0x0208 0000~0x0210 0000 (depending on the remapping space MMF_REMCR0/1.RMSIZE[4:0])

When the remapping function is valid, the address correspondence is as:

Table 1-2 Sample Destination Address Configuration

Register Setting	Remap Address (CPU Address - CPUADDR[31:0])	Source Address		
		High 3-bit Address	Middle Address	Low Address
RMSIZE[4:0]=0b01110 case (remap space: 16KB)	0x0200 0000~0x0200 3FFF	All 0	RMTADDR[16:2]	CPUADDR[13:0]
RMSIZE[4:0]=0b01111 case (remap space: 32KB)	0x0200 0000~0x0200 7FFF	All 0	RMTADDR[16:3]	CPUADDR[14:0]
RMSIZE[4:0]=0b10000 case (remap space: 64KB)	0x0208 0000~0x0208 FFFF	All 0	RMTADDR[16:4]	CPUADDR[15:0]
RMSIZE[4:0]=0b10001 case (remap space: 128KB)	0x0208 0000~0x0209 FFFF	All 0	RMTADDR[16:5]	CPUADDR[16:0]

Example:

1. *Use the function of remapping address 0, set the source address as the main flash memory FLASH address 0x0000 8000, and the remapping space is 32K, and the register MMF_REMCR0 needs to be set to 0x8000 800F.*
2. *Use the remapping address 1 function, set the source address to the high-speed SRAM address 0x1FFF 8000, the remapping space to 16K, and set the register MMF_REMCR1 to 0x9FFF 800E.*

Note:
The starting address of the source address should be set as an integer multiple of the remapping space.

1.4 Register Description

Table 1-3 MMF Base Address

Name	Base Address	Description
MMF	0x4001 0500	MMF Base Address

Table 1-4 Register List

Register	Description	Offset Address	Bit Width	Reset Value
MMF_REMPRT	access protection register	0x0000	32	0x0000 0000
MMF_REMCR0	Remap Control Register 0	0x0004	32	0x0000 0000
MMF_REMCR1	Remap Control Register 1	0x0008	32	0x0000 0000

1.4.1 Access Protection Register (MMF_REMPRT)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
MMF_REMPRT[15:0]															

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b16	Res	Reserved Bit	Read at “0”, write at “0”	R
b15~b0	MMF_REMPRT[15:0]	protection register	Registers MMF_REMCR0 and MMF_REMCR1 are write-protected: First write 0x0123 to MMF_REMPRT[15:0] and then write 0x3210 to release the protection; When the registers MMF_REMCR0 and MMF_REMCR1 are in the write protection state, the read register is 0 When the registers MMF_REMCR0 and MMF_REMCR1 are released from the write protection state, the read register is 1	RW

1.4.2 Remap Control Register (MMF_REMCRx, x=0, 1)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
EN	Res			RMTADDR[16:4]											
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
RMTADDR[3:0]				Res							RMSIZE[4:0]				

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31	EN	Remap Valid Bit	0: no remapping 1: remapping is valid	RW
b30~b29	Res	Reserved Bit	Read at “0”, write at “0”	R
b28~b12	RMTADDR[16:0]	Source address	The effective number of digits is related to the setting of RMSIZE[4:0]. Settings can be referred to Table 1-2.	RW
b11~b5	Res	Reserved Bit	Read at “0”, write at “0”	R
b4~b0	RMSIZE[4:0]	remap space	Set remapping space 00000~ 01011: Reserved, setting is prohibited 01100：4KB 01101：8KB 01110：16KB 01111：32KB 10000：64KB 10001：128KB (64KB products cannot be set) 10010~ 11111: Reserved, setting is prohibited	RW

2 Bus Architecture (BUS)

2.1 Overview

The main system is composed of 32-bit multilayer AHB bus matrix, which can interconnect the following host bus and slave bus.

- host bus
  - Cortex-M4 core CPU-I bus, CPU-D bus, CPU-S bus
  - System DMA Bus
- Slave bus
  - Flash ICODE bus
  - Flash DCODE bus
  - Flash MCODE bus (the bus used by host other than the CPU to access Flash)
  - High-speed SRAMH (SRAMH 16KB) bus
  - System SRAM (SRAM0 16KB) bus
  - System SRAM (Ret SRAM 4KB) bus
  - APB1 peripheral bus (SPI/ USART/ EFM/ AOS)
  - APB2 peripheral bus (TimerA/Timer0)
  - APB3 peripheral bus (ADC/DAC)
  - APB4 peripheral bus (PLA/FCM/ WDT/ SWDT/ PWC/ CTC/ RTC/ WKTM)
  - APB5 peripheral bus (TimerA/ Timer4/ Timer6/ CMP/ I2C/EMB/HRPWM)
  - AHB1 peripheral bus (CMU/ GPIO/ DMA/ INTC/ DMPU)
  - AHB3 peripheral bus (MCAN)
  - AHB4 peripheral bus (CRC)

With the help of the bus matrix, high -efficiency concurrent access from the Master bus to the slave bus can be realized.

2.2 Bus Architecture

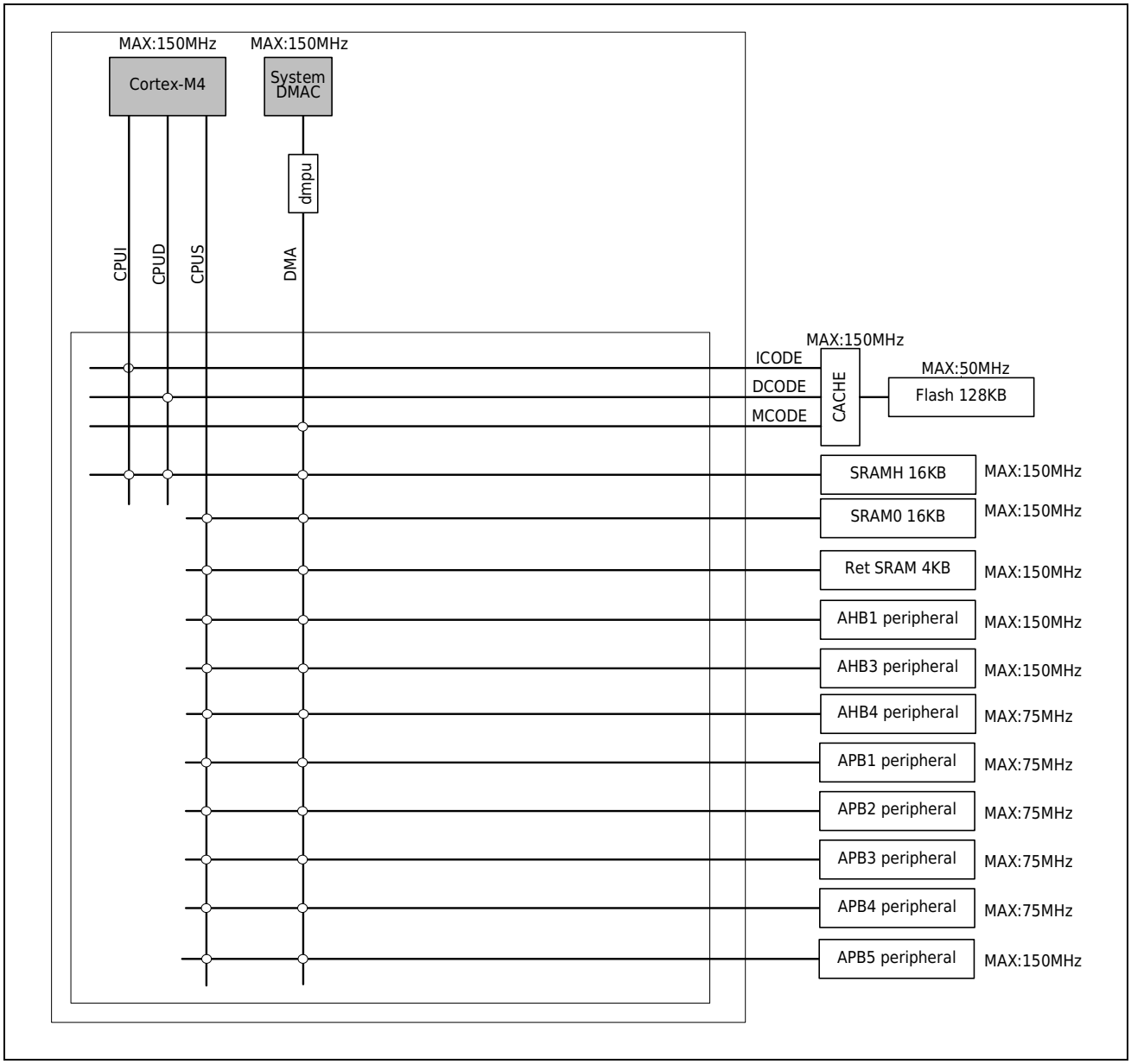


Figure 2-1 Bus Architecture Diagram

The bus matrix is used for access arbitration management among the master buses. Arbitration uses a round-robin scheduling algorithm.

- CPU-I bus

  The instruction bus of the Cortex-M4 core, through which the CPU obtains instructions. The access objects are Flash and SRAMH containing codes.

- CPU-D bus

  The data bus of the Cortex-M4 core, through which the CPU performs immediate load and debug access. The access objects are Flash and SRAMH that contain code or data.

- CPU-S bus

The system bus of the Cortex-M4 core, the CPU accesses peripherals or system SRAM through this bus, and can also obtain instructions and immediate data through this bus (the efficiency is lower than that through the CPU-I bus and CPU-D bus). The access objects are SRAM0/ Ret SRAM peripherals and external expansion space.

- DMA bus

  System DMA dedicated bus, DMA access data memory and peripherals through this bus, and the access objects are Flash/ SRAMH/ SRAM0/ Ret SRAM peripherals and external expansion space.

2.3 Bus Function

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

Through the bus matrix, when the access targets of different Master buses do not conflict, each access can be carried out simultaneously. For example, CPU-I accesses Flash, CPU-D accesses SRAMH, CPUS accesses APB peripherals, and DMA accesses SRAM0, and these accesses can be performed simultaneously.

3 Reset Management Unit (RMU)

3.1 Introduction

The chip is configured with 15 reset modes.

- Power-on Reset (POR)
- NRST pin reset (NRST)
- Brown-out Reset (BOR)
- Programmable Voltage Detect 1 Reset (PVD1R)
- Programmable Voltage Detect 2 Reset (PVD2R)
- Watchdog Reset (WDTR)
- Special watchdog reset (SWDTR)
- Power-down wake-up reset (PDRST)
- Software Reset (SRST)
- MPU Error Reset (MPUR)
- RAM Parity Reset (RAMPR)
- RAMECC reset (RAMECCR)
- Clock exception reset (CKFER)
- External High Speed Oscillator Abnormal Shock Reset (XTALER)
- Cortex-M4 Lockup Reset (LKUPR)

3.2 Reset Mode and Reset Flag Bit

The reset method and generation condition are as shown Table 3-1.

Table 3-1 Reset Methods and Generation Conditions

Reset Mode	Generating Condition
Power-on reset	VCC powered on
Reset pin reset Pin reset is a reset caused by the drive of the NRST pin to a low level.	NRST pin input low level
Brown-out reset	VCC voltage drops below VBOR voltage
Programmable voltage detection 1 reset	VCC voltage drops below PVD1 voltage
Programmable voltage detection 2 reset	VCC voltage drops below PVD2 voltage
watchdog reset	Watchdog timer generates a refresh error or an overflow error
Special watchdog resetting	Refresh error or overflow error occurred in special watchdog
Power-down wake-up reset	By setting the reset generated by the power-down mode, the core wakes up from the reset state after the power-down wake-up event occurs
Software reset	Set the reset register bit (ARM register AIRCR.SYSRESETREQ bit)
MPU error reset	Reset generated by MPU access error
RAM parity reset	Reset generated when a parity error occurs in RAM
RAM ECC error reset	Reset generated when ECC error occurs in RAM
Clock frequency exception reset	When the clock frequency monitoring function (FCM) detects a clock cycle error
External High Speed Oscillator Abnormal Shock Reset	Resetting of External High-speed Oscillator Abnormal Shock
Cortex-M4 Lockup Reset (LKUPR)	When Cortex-M4 encounters a serious abnormality, it stops its PC pointer at the current address and locks itself to reset the entire chip after several clock cycles delay

When a reset occurs, the chip will set the corresponding reset flag according to the reset method, and the reset flag is as shown Table 3-2. 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 3-2 Reset Methods and Reset Flags

Reset Flag	Reset Mode														
	Power-on reset	Pin reduction	Brown-out reset	Voltage detection 1 reset	Voltage detection 2 reset	watchdog reset	Special watchdog	Power-down wake-up	Software reset	MPU error reset	RAM parity error reset	RAM ECC error reset	Clock frequency exception	External High Speed	Cortex-M4 Lockup Reset
Power-on reset flag (RMU_RSTF0.PORF)	✓	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Pin Reset Flag (RMU_RSTF0.PINRF)	×	✓	—	—	—	—	—	—	—	—	—	—	—	—	—
Brown-out reset flag (RMU_RSTF0.BORF)	×	×	✓	—	—	—	—	—	—	—	—	—	—	—	—
Programmable voltage sense 1 reset flag (RMU_RSTF0.PVD1RF)	×	×	×	✓	—	—	—	—	—	—	—	—	—	—	—
Programmable voltage sense 2 reset flag (RMU_RSTF0.PVD2RF)	×	×	×	—	✓	—	—	—	—	—	—	—	—	—	—
Watchdog Reset Flag (RMU_RSTF0.WDRF)	×	×	×	—	—	✓	—	×	—	—	—	—	—	—	—
Special watchdog reset flag (RMU_RSTF0.SWDRF)	×	×	×	—	—	—	✓	×	—	—	—	—	—	—	—
Power-down wake-up reset flag (RMU_RSTF0.PDRF)	×	—	—	—	—	—	—	✓	—	—	—	—	—	—	—
Software reset flag (RMU_RSTF0.SWRF)	×	×	×	—	—	—	—	×	✓	—	—	—	—	—	—
MPU error reset (RMU_RSTF0.MPUERF)	×	×	×	—	—	—	—	×	—	✓	—	—	—	—	—
RAM parity error reset (RMU_RSTF0.RAPERF)	×	×	×	—	—	—	—	×	—	—	✓	—	—	—	—
RAM ECC reset (RMU_RSTF0.RAECRF)	×	×	×	—	—	—	—	×	—	—	—	✓	—	—	—
Clock Frequency Abnormal Reset (RMU_RSTF0.CKFERF)	×	×	×	—	—	—	—	×	—	—	—	—	✓	—	—
External high-speed oscillator abnormal stop reset (RMU_RSTF0.XTALERF)	×	×	×	—	—	—	—	×	—	—	—	—	—	✓	—
Cortex-M4 Lockup Reset (RMU_RSTF0. LKUPRF)	×	×	×	—	—	—	—	×	—	—	—	—	—	—	✓

✓: set ×: clear —: unchanged

3.3 Reset Timing

3.3.1 Power-on reset

The power-on reset is an internal reset caused by the power-on reset circuit, and the timing is as follows Figure 3-1. 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. Please refer to [Description of Action of Power-on/Power-off Reset] for details on power-on reset.

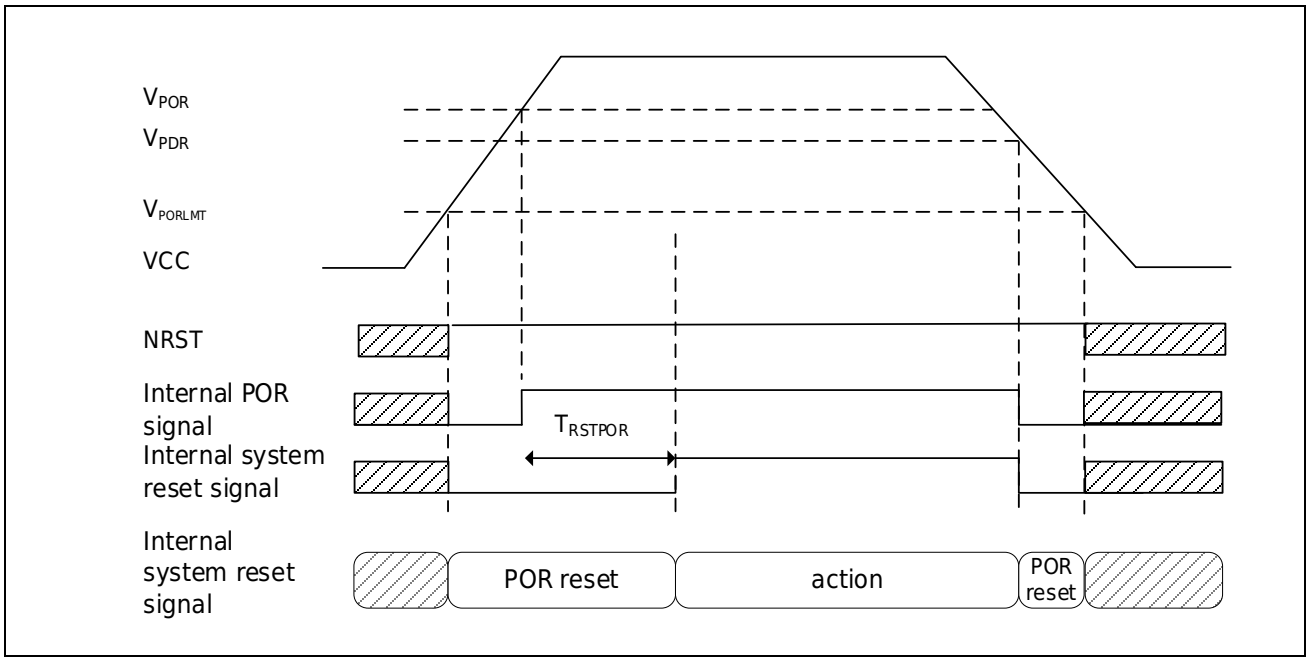


Figure 3-1 Power-on Reset

3.3.2 NRST Pin Reset

Pin reset is a reset caused by the NRST pin being driven low, and the reset timing is as followsFigure 3-2. 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.

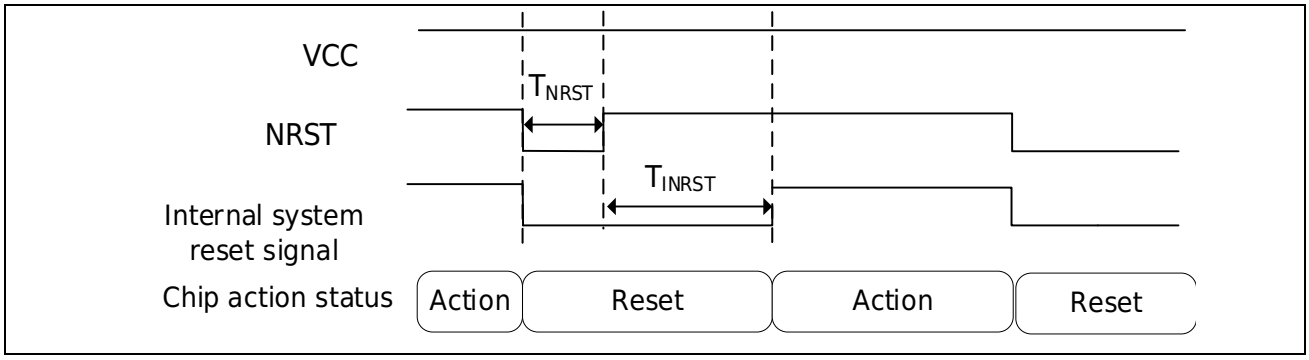


Figure 3-2 NRST Reset Timing

3.3.3 Brown-out Reset

Brown-out reset is an internal reset caused by the voltage monitoring circuit, the timing is as follows Figure 3-3. After the brown-out is set as reset enable through the ICG register, if the VCC voltage is lower than the monitoring voltage V_{BOR}, RMU_RSTF0.VBORF will be set. When the VCC voltage is higher than the monitoring voltage V_{BOR}, the reset will be released after the V_{BOR} reset time (T_{RSTBOR}).

For the reset setting of brown-out, please refer to [Brown-out Reset (BOR) Description].

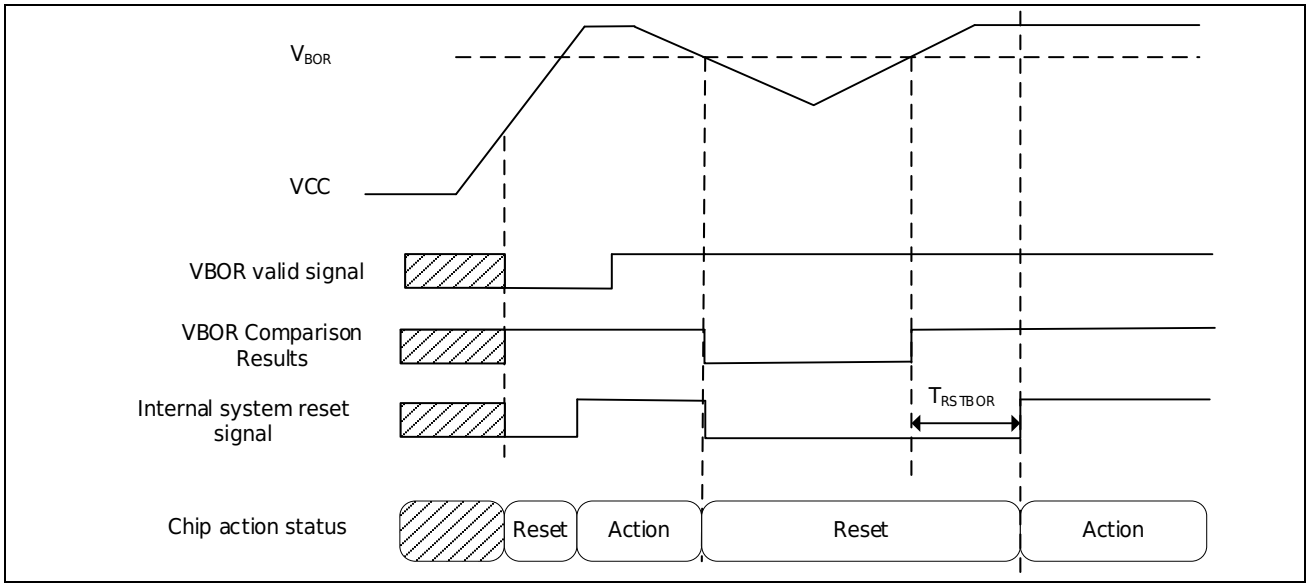


Figure 3-3 Brownout Reset

3.3.4 Programmable Voltage Detection 1 Reset, Programmable Voltage Detection 2 Reset

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

When the VCC voltage is lower than the monitoring voltage of the programmable voltage detection 1, according to the setting, the programmable voltage detection 1 reset can be generated and RMU_RSTF0.PVD1F is set. When the VCC voltage is higher than the monitoring voltage of programmable voltage detection 1, the reset will be released after the reset time of PVD1 (T_{IPVD1}).

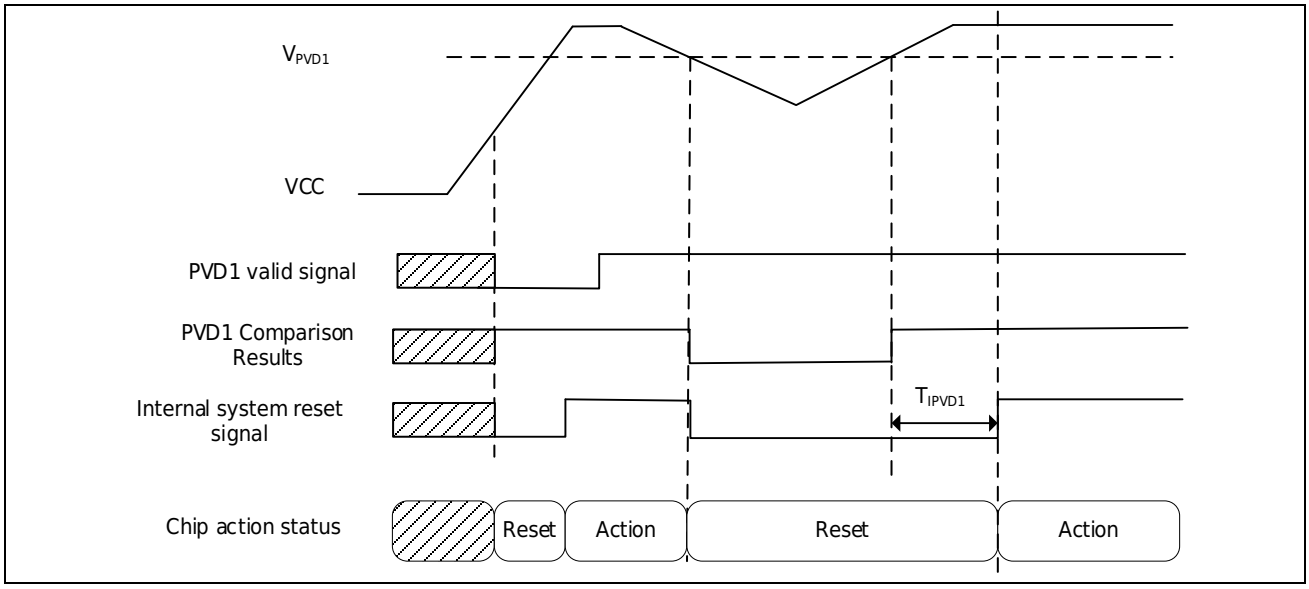


Figure 3-4 Programmable Voltage Detection 1 Reset

When the VCC voltage is lower than the monitoring voltage of the programmable voltage detection 2, according to the setting, a programmable voltage detection 2 reset may be generated and RMU_RSTF0.PVD2F is set. When the VCC voltage is higher than the monitoring voltage of programmable voltage detection 2, the reset will be released after the reset time of PVD2 (T_{IPVD2}).

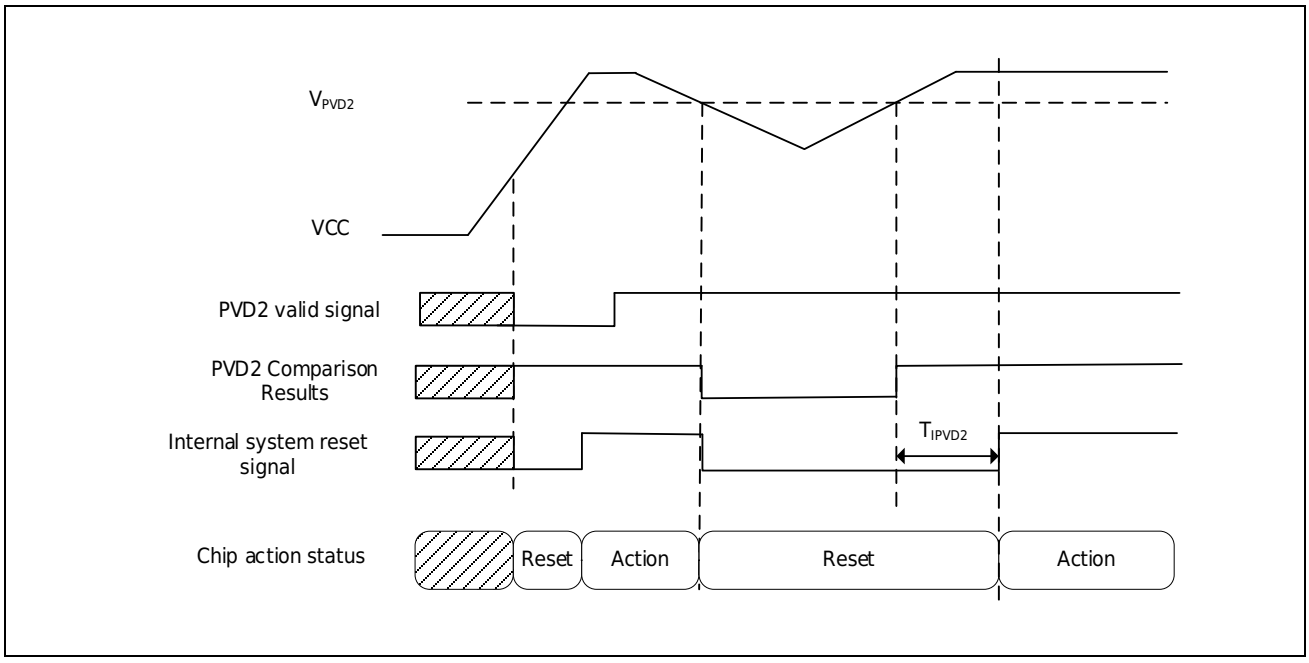


Figure 3-5 Programmable Voltage Detection 2 Reset

For reset settings of programmable voltage detection 1 and programmable voltage detection 2, please refer to [Power Voltage Detection Unit (PVD) Description].

3.3.5 Watchdog Reset, Special Watchdog Reset

The watchdog reset is an internal reset caused by the watchdog timer, and the special watchdog reset is an internal reset caused by the special watchdog timer. The reset timing is as follows Figure 3-6.

After the watchdog reset is set to be valid, the watchdog reset will be generated when the watchdog timer underflows or the write operation is not performed during the refresh period. A watchdog reset sets RMU_RSTF0.WDRF. After the watchdog reset is generated, the chip is released from reset after the internal reset time T_{RIPT}.

After the special watchdog reset is set to be valid, the watchdog reset will be generated when the special watchdog timer underflows or the write operation is not performed during the refresh period. Special watchdog reset RMU_RSTF0.SWDRF setup. After generating special watchdog reset, after internal reset time T_{RIPT}, the chip is released.

For details about watchdog reset and special watchdog reset, please refer to [Watchdog Counter (WDT/SWDT)].

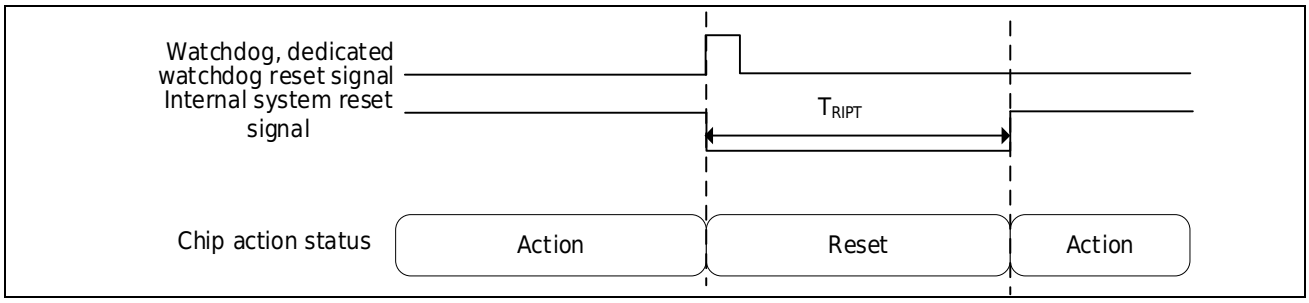


Figure 3-6 Watchdog and Special Watchdog Reset

3.3.6 Power-down Wake-up Reset

Power-down wake-up reset is an internal reset generated when the chip executes the WFI command when PWC_PWRC0.PWDN is set to 1, enters power-down mode, and releases power-down mode through a power-down mode wake-up event. The timing is as follows Figure 3-7. After the power-down mode is released and the return time (T_{IPDx}, x=1, 2, 3, 4) has elapsed, the power-down wake-up reset is released. The return time varies according to the specific power-down mode set, the minimum is in power-down mode 1, and the maximum is in power-down mode 3.

For details about power-down wake-up reset, please refer to [Power Down Mode].

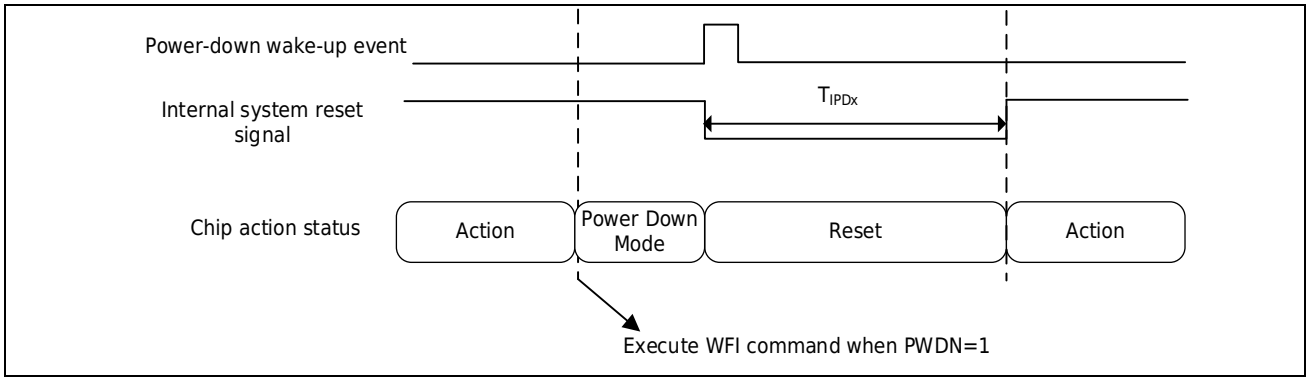


Figure 3-7 Power-down Wake-up Reset

3.3.7 Software Reset

A software reset is generated by writing the SYSRESETREQ bit of ARM register AIRCR. When generating software reset, RMU _ RSTF0.The SWRF bit is set. After the internal reset time T_{RIPT}, the chip is de-reset.

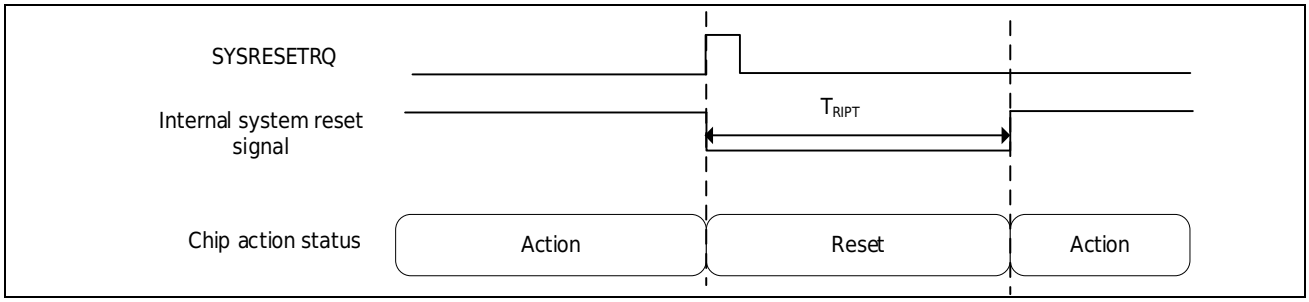


Figure 3-8 Software Reset

3.3.8 MPU Error Reset

MPU error reset sets RMU_RSTF0.MPUERF, the timing is as follows Figure 3-9. After the internal reset time T_{RIPT}, the chip is de-reset.

For the setting of MPU error reset, please refer to [Memory Protection Unit (MPU)].

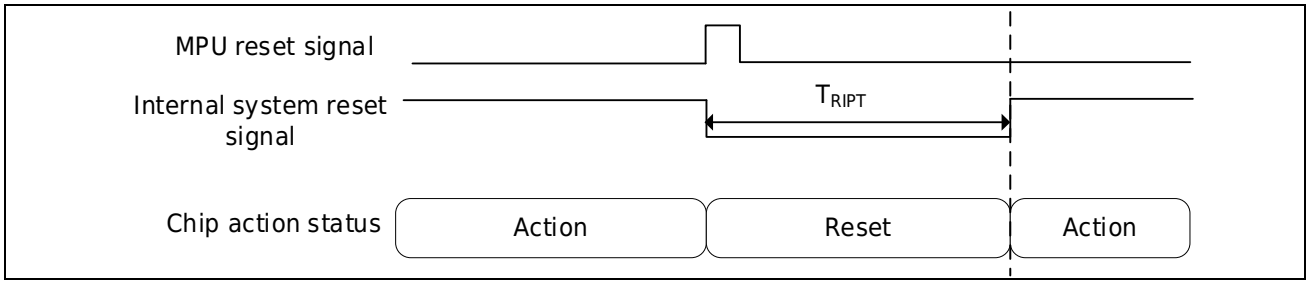


Figure 3-9 MPU Error Reset

3.3.9 RAM Parity Reset

When a RAM parity error occurs, a RAM parity reset may occur according to the settings. The reset timing is shown in the Figure 3-10. A RAM parity error sets RMU_RSTF0.RAPERF. After the internal reset time T_{RIPT}, the chip is de-reset.

For the setting of RAM parity error reset, please refer to [Built-in SRAM (SRAM)].

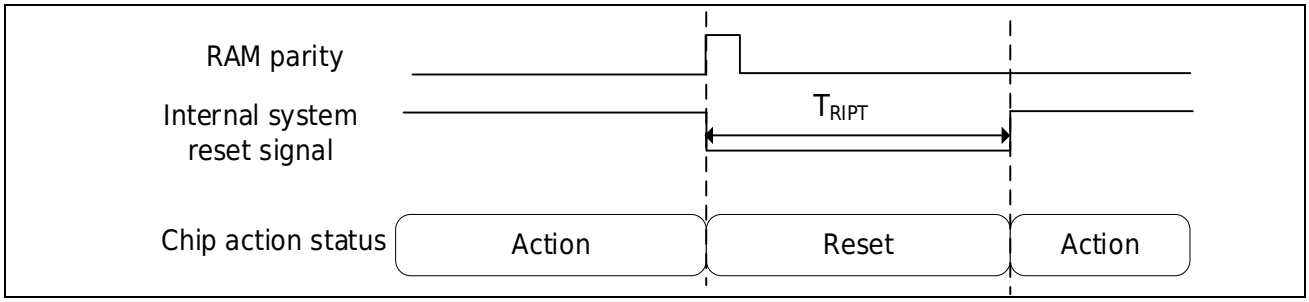


Figure 3-10 RAM Parity Reset

3.3.10 RAMECC Reset

When an error occurs in RAMECC verification, a RAMECC reset may be generated according to the settings, and its reset timing is shown in the Figure 3-11. A RAMECC reset sets RMU_RSTF0.RAECRF. After the internal reset time T_{RIPT}, the chip is de-reset.

For setting of RAMECC reset, please refer to [Built-in SRAM (SRAM)].

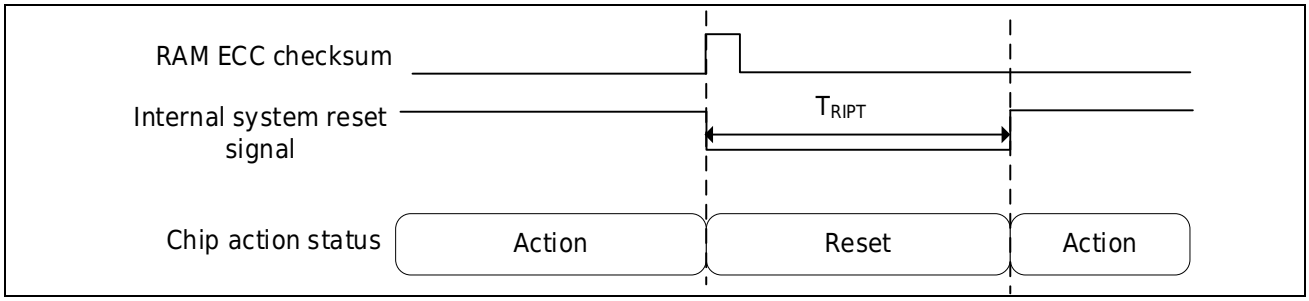


Figure 3-11 RAMECC Reset

3.3.11 Clock Frequency Exception Reset

When the FCM module built into the chip detects that the clock frequency is abnormal, it can generate a clock frequency abnormal reset according to the setting. The reset timing is shown in the Figure 3-12. Clock frequency exception reset sets RMU_RSTF0.CKFERF. After the internal reset time T_{RIPT}, the chip is de-reset.

Please refer to [Clock Frequency Measurement] for setting of clock frequency abnormal reset.

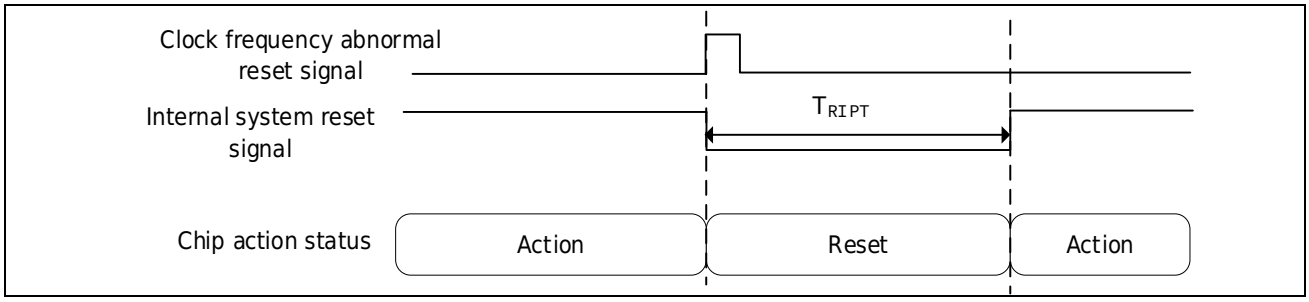


Figure 3-12 Clock Frequency Abnormal Reset

3.3.12 External High Speed Oscillator Abnormal Shock Reset

RMU _ RSTF0 is generated when the external high-speed oscillator abnormally stops vibration when the shock stop detection module is valid and the reset is enabled.XTALERF is set. After the internal reset time T_{RIPT}, the chip is de-reset.

Please refer to [External High-speed Oscillator Fault Detection] for the setting of external high-speed oscillator abnormal stop vibration reset.

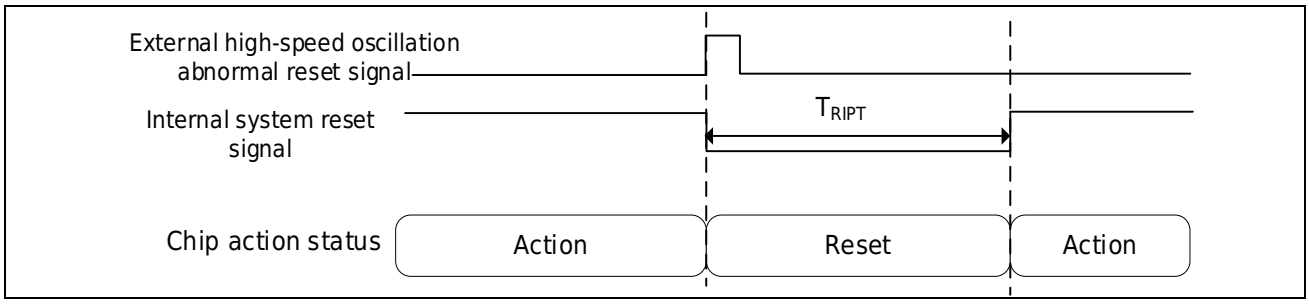


Figure 3-13 External High-speed Oscillation Abnormal Reset

3.3.13 Cortex-M4 Lockup Reset

When Cortex-M4 encounters a serious abnormality, it stops its PC pointer at the current address and locks itself to reset the entire chip after several clock cycles delay, the timing is as followsFigure 3-14. The Cortex-M4 Lockup reset sets RSTF0.LKUPRF. After the internal reset time T_{RIPT}, the chip is de-reset.

The Cortex-M4 Lockup reset must be used after RMU_PRSTCR0.LKUPREN is set.

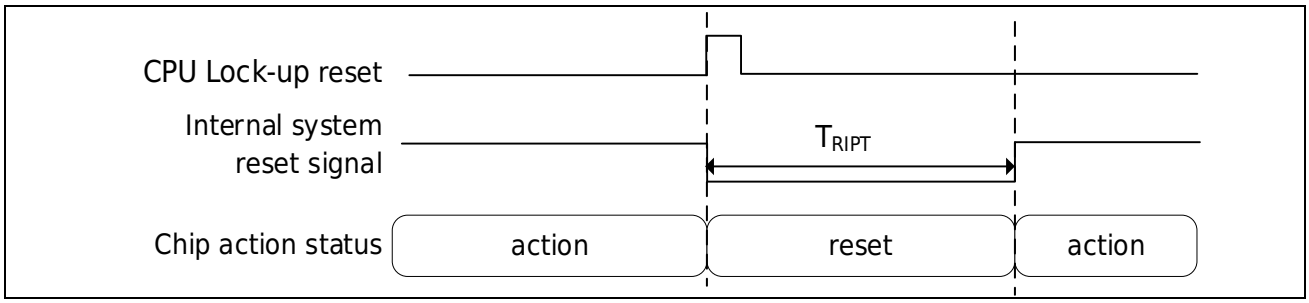


Figure 3-14 Cortex-M4 Lockup Reset

3.3.14 Judgment of Reset Mode

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 MULTIRF bit in RMU _ RSTF0 is 1. After reading RMU _ RSTF0, all reset flags can be cleared to 0 by setting the CLRF bit. After setting and clearing RMU_RSTF0, wait at least 6 CPU clock cycles before reading the RMU_RSTF0 register again.

3.3.15 Module Function Reset

In addition to the above 15 chip resets, the product also configures functional reset registers RMU_FRST0~3 for each peripheral module. Writing 0 to the corresponding bit of RMU_FRST can reset the function of the corresponding module. After RMU_FRST is written to 0, it will be automatically set to 1 after the rising edge of the clock of two peripheral modules.

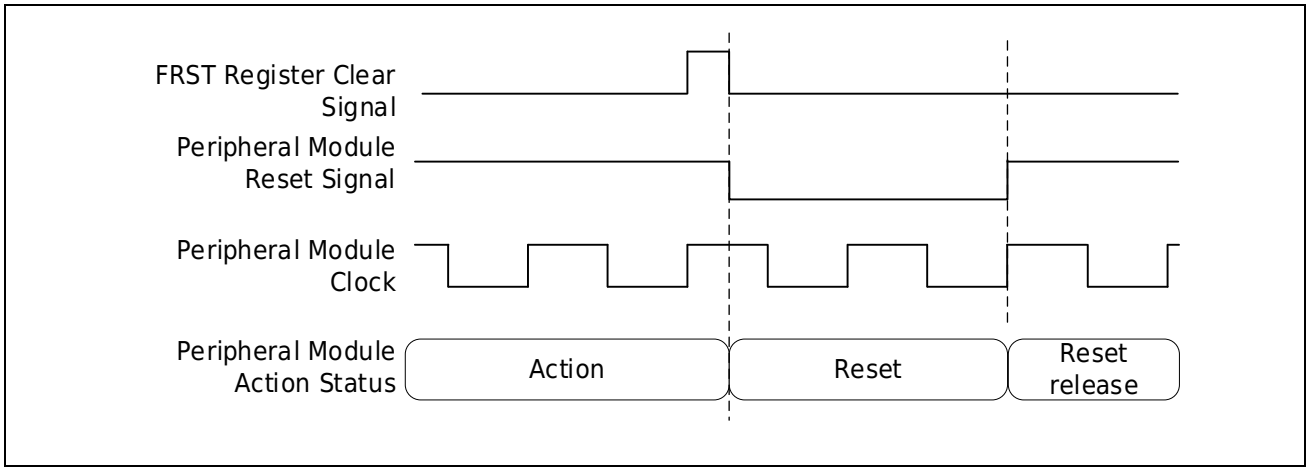


Figure 3-15 Module Function Reset

3.3.16 Reset Conditions for Each Module

All reset sources in the table refer to the 15 reset sources described in the introduction of this chapter.

Table 3-3 Reset Conditions for Each Module

Modules	Register	Reset Source
Debug Controller (DBGC)	MCUSTPCTL, MCUSTPCTL2, MCUTRACECTL, MCUDBGSTAT	1. Power-on reset 2. Power-down wake-up reset
Real Time Clock (RTC)	RTC internal registers	Module software reset control bit: RTC_CR0.RESET
Built-in SRAM (SRAM)	SRAMC_CKSR	1. Power-on reset 2. Power-down mode 3 wake-up reset
	Other registers besides SRAMC_CKSR	All reset sources
Power control (PWC) Clock control (CMU)	PWC_PWRC0, PWC_PWRC1, PWC_PDWKE0, PWC_PDWKE1, PWC_PDWKE2, PWC_PDWKES, CMU_XTALCFGR	All reset sources other than the following: 1. Power-down mode 1 wake-up reset 2. Power-down mode 2 wake-up reset 3. Power-down mode 4 wake-up reset
	PWC_PDWKF0, PWC_PDWKF1	All reset sources except power-down wake-up reset
	PWC_PVDLCR, PWC_PVDCR1, PWC_PVDFCR, PWC_PVDCR0, PWR_PWRC2, PWR_PWRC4	1. Power-on reset 2. Pin reduction 3. Brown-out reset 4. watchdog reset 5. Special watchdog resetting 6. Power-down mode 3 wake-up reset
	PWC_PVDICR. PVD1EDGS[1:0], PWC_PVDDSR.PVD1DETFLG, PWC_PVDICR. PVD2EDGS[1:0], PWC_PVDDSR.PVD2DETFLG	1. Power-on reset 2. Pin reduction 3. Brown-out reset 4. watchdog reset 5. Special watchdog resetting 6. Power-down wake-up reset
	PWC_WKTC	Power-on reset
	Other than the above	All reset sources
Registers other than the above modules		All reset sources

3.4 Register Description

Table 3-4 RMU Base Address

Name	Base Address	Description
RMU	0x4004 CCE0	RMU Base Address

Table 3-5 RMU Register List

Register	Description	Offset Address	Bit Width	Reset Value
RMU_FRST0	Function reset control register 0	0x00	32	0xFFFF FFFF
RMU_FRST1	Function reset control register 1	0x04	32	0xFFFF FFFX
RMU_FRST2	Function reset control register 2	0x08	32	0xFFFF FFFF
RMU_FRST3	Function reset control register 3	0x0C	32	0xFFFF FFFF
RMU_PRSTCR0	Reset control register	0x10	8	0x40
RMU_RSTF0	Reset Flag Register	0x14	32	Different Reset Values According to Different Reset Methods

3.4.1 Function Reset Control 0 (RMU_FRST0)

Reset value: 0xFFFF FFFF

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res								CRC	Res				CTC	AOS	FCM
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res	DMA	Res		PLA	Res										

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b24	Res	Reserved Bit	When reading, all bits are "1", when writing, please write "1" to all bits	RW
b23	CRC	CRC Reset control	0：CRC function reset 1: Invalid CRC function	RW
b22~b19	Res	Reserved Bit	Read at “1”, write at “1”	RW
b18	CTC	CTC Reset control	0: Internal clock calibrator CTC function reset valid 1: Internal clock calibrator CTC function reset release	RW
b17	AOS	AOS Reset control	0: Automatic operation system AOS function reset valid 1: Automatic operation system AOS function reset release	RW
b16	FCM	FCM Reset control	0: The clock frequency measurement module FCM function in the clock controller CMU is reset and valid 1: The clock frequency measurement module FCM function in the clock controller CMU reset and released	RW
b15	Res	Reserved Bit	Read at “1”, write at “1”	RW
b14	DMA	DMA Reset control	0: DMA Function Reset Valid 1: DMA Function Reset Release	RW
b13~b12	Res	Reserved Bit	Read at “1”, write at “1”	RW
b11	PLA	PLA Reset control	0: PLA Function Reset Valid 1: PLA Function Reset Release	RW
b10~b0	Res	Reserved Bit	When reading, all bits are "1", when writing, please write "1" to all bits	RW

3.4.2 Function Reset Control 1 (RMU_FRST1)

Reset value: 0xFFFF FFFX

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															SPI
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res											I2C	Res		MCAN2	MCAN1

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b17	Res	Reserved Bit	When reading, all bits are "1", when writing, please write "1" to all bits	RW
b16	SPI	SPI Reset control	0: Serial peripheral interface SPI function reset valid 1: Serial peripheral interface SPI function reset release	RW
b15~b5	Res	Reserved Bit	When reading, all bits are "1", when writing, please write "1" to all bits	RW
b4	I2C	I2C Reset control	0: I2C Function Reset Valid 1: I2C Function Reset Release	RW
b3	Res	Reserved Bit	When reading, all bits are "1", when writing, please write "1"	RW
b2	Res	Reserved Bit	It is indeterminate when reading, please write "1" when writing	RW
b1	MCAN2	MCAN2 Reset control	0: MCAN2 Function Reset Valid 1: MCAN2 Function Reset Release	RW
b0	MCAN1	MCAN1 Reset control	0: MCAN1 Function Reset Valid 1: MCAN1 function reset invalid	RW

Note:

The register (CMU_CANCKCFGR) selects XTAL as the CAN clock by default. If XTAL is not configured to oscillate, bit2-0 is 0; if XTAL is configured to oscillate, bit2-0 is 1.

3.4.3 Function Reset Control 2 (RMU_FRST2)

Reset value: 0xFFFF FFFF

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res							TMRA_5	TMRA_4	TMRA_3	TMRA_2	TMRA_1	Res			
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
EMB	Res	TMR0_2	TMR0_1	HRPWM	Res		TMR4	Res							TMR6

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b25	Res	Reserved Bit	When reading, all bits are "1", when writing, please write "1" to all bits	RW
b24	TMRA_5	TimerA_5 Reset control	0: TimerA Unit 5 Function Reset Valid 1: TimerA Unit 5 Function Reset Release	RW
b23	TMRA_4	TimerA_4 Reset control	0: TimerA Unit 4 Function Reset Valid 1: TimerA Unit 4 Function Reset Release	RW
b22	TMRA_3	TimerA_3 Reset control	0: TimerA Unit 3 Function Reset Valid 1: TimerA Unit 3 Function Reset Release	RW
b21	TMRA_2	TimerA_2 Reset control	0: TimerA Unit 2 Function Reset Valid 1: TimerA Unit 2 Function Reset Release	RW
b20	TMRA_1	TimerA_1 Reset control	0: TimerA Unit 1 Function Reset Valid 1: TimerA Unit 1 Function Reset Release	RW
b19~b16	Res	Reserved Bit	When reading, all bits are "1", when writing, please write "1" to all bits	RW
b15	EMB	EMB Reset control	0: Emergency brake module EMB Function Reset Valid 1: Emergency brake module EMB Function Reset Release	RW
b14~b13	Res	Reserved Bit	When reading, all bits are "1", when writing, please write "1" to all bits	RW
b13	TMR0_2	Timer0_2 Reset control	0: Timer0 Unit 2 Function Reset Valid 1: Timer0 Unit 2 Function Reset Release	RW
b12	TMR0_1	Timer0_1 Reset control	0: Timer0 Unit 1 Function Reset Valid 1: Timer0 Unit 1 Function Reset Release	RW
b11	HRPWM	HRPWM Reset control	0: HRPWM Unit 1 Function Reset Valid 1: HRPWM unit 1 Function Reset Release	RW
b10~b9	Res	Reserved Bit	Read at “1”, write at “1”	RW
b8	TMR4	Timer4 Reset control	0: Timer4 Function Reset Valid 1: Timer4 Function Reset Release	RW
b7~b1	Res	Reserved Bit	When reading, all bits are "1", when writing, please write "1" to all bits	RW
b0	TMR6	Timer6 Reset control	0: Timer6 unit 1234 Function Reset Valid 1: Timer6 unit 1234 Function Reset Release	RW

Note:

TMR_SYNENR is not controlled by the RMU_FRST register.

3.4.4 Function Reset Control 3 (RMU_FRST3)

Reset value: 0xFFFF FFFF

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res								USART4	USART3	USART2	USART1	Res			
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res						CMP3	CMP12	Res		DAC2	DAC1	Res	ADC3	ADC2	ADC1

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31-b24	Res	Reserved Bit	When reading, all bits are "1", when writing, please write "1" to all bits	RW
b23	USART4	USART4 Reset control	0: Universal Synchronous Asynchronous Receiver USART Unit 4 Function Reset Valid 1: Universal Synchronous Asynchronous Receiver USART Unit 4 Function Reset Release	RW
b22	USART3	USART3 Reset control	0: Universal Synchronous Asynchronous Receiver sendter USART Unit 3 Function Reset Valid 1: Universal Synchronous Asynchronous Receiver USART Unit 3 Function Reset Release	RW
b21	USART2	USART2 Reset control	0: Universal Synchronous Asynchronous Receiver USART Unit 2 Function Reset Valid 1: Universal Synchronous Asynchronous Receiver USART Unit 2 Function Reset Release	RW
b20	USART1	USART1 Reset control	0: Universal Synchronous Asynchronous Receiver USART Unit 1 Function Reset Valid 1: Universal Synchronous Asynchronous Receiver USART Unit 1 Function Reset Release	RW
b19-b10	Res	Reserved Bit	When reading, all bits are "1", when writing, please write "1" to all bits	RW
b9	CMP3	CMP3 Reset control	0: Voltage comparator CMP Unit 3 Function Reset Valid 1: Voltage comparator CMP Unit 3 Function Reset Release	RW
b8	CMP12	CMP12 Reset control	0: Voltage comparator CMP Unit 1 and Unit 2 Function Reset Valid 1: Voltage comparator CMP Unit 1 and Unit 2 Function Reset Release	RW
b7-b6	Res	Reserved Bit	When reading, all bits are "1", when writing, please write "1" to all bits	RW
b5	DAC2	DAC2 Reset control	0: Digital-to-Analog Converter DAC2 Function Reset Valid 1: Digital-to-Analog Converter DAC2 Function Reset Release	RW
b4	DAC1	DAC1 Reset control	0: Digital-to-Analog Converter DAC1 Function Reset Valid 1: Digital-to-Analog Converter DAC1 Function Reset Release	RW
b3	Res	Reserved Bit	Read at “1”, write at “1”	RW
b2	ADC3	ADC3 Reset control	0: Analog-to-Digital Conversion Module ADC Unit 3 Function Reset Valid 1: Analog-to-Digital Conversion Module ADC Unit 3 Function Reset Release	RW
b1	ADC2	ADC2 Reset control	0: Analog-to-Digital Conversion Module ADC Unit 2 Function Reset Valid 1: Analog-to-Digital Conversion Module ADC Unit 2 Function Reset Release	RW
b0	ADC1	ADC1 Reset control	0: Analog-to-Digital Conversion Module ADC Unit 1 Function Reset Valid 1: Analog-to-Digital Conversion Module ADC Unit 1 Function Reset Release	RW

3.4.5 Reset Control Register (RMU_PRSTCR0)

Reset Value: 0x40

b7	b6	b5	b4	b3	b2	b1	b0
Res		LKUPREN	Res				

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b7	Res	Reserved Bit	Read at “0”, write at “0”	RW
b6	Res	Reserved Bit	Read at “1”, write at “1”	RW
b5	LKUPREN	Lockup Reset Enable	0: Lockup reset is invalid 1: Lockup reset enable	RW
b4-b0	Res	Reserved Bit	Read at “0”, write at “0”	RW

3.4.6 Reset Flag Register (RMU _ RSTF0)

Reset value: 0xXXXX XXXX (depending on the reset mode, the reset value is different)

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
CLRF	MULTIRF	Res													
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res	LKUPRF	XTALERF	CKFERF	RAECRF	RAPERF	MPUERF	SWRF	PDRF	SWDRF	WDRF	PVD2RF	PVD1RF	BORF	PINRF	PORF

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31	CLRF	Clear reset flag	The software is set to 1 to clear the reset flag bit. Read at 0. The set action must be performed after RMU _ RSTF0 is read. 0: No operation 1: Reset mark	RW
b30	MULTIRF	More than 2 reset occurrence flags	When two or more resets occur, the hardware is set. Reset by setting CLRF. 0: No reduction of two or more 1: When two or more resets occur	RW
b29~b15	Res	Reserved Bit	Read at “0”, write at “0”	RW
b14	LKUPRF	Cortex-M4 Lockup Reset	When the Cortex-M4 Lockup reset occurs, the hardware is set. Reset by setting CLRF. 0: Cortex-M4 Lockup reset not occurring 0: Cortex-M4 Lockup reset occurred	RW
b13	XTALERF	External high-speed oscillator exception Stop-and-reset flag	When an external high-speed oscillator abnormally stops and resets, the hardware is set. Reset by setting CLRF. 0: No External High-Speed Oscillator Abnormal Shock Reset 1: External high-speed oscillator abnormal shuffling reset	RW
b12	CKFERF	abnormal clock frequency Reset flag	Set by hardware when a clock frequency exception reset occurs. Reset by setting CLRF. 0: No clock frequency exception reset occurs 1: A clock frequency abnormal reset occurs	RW
b11	RAECRF	RAMECC reset flag	Set by hardware when a RAMECC reset occurs. Reset by setting CLRF. 0: No RAMECC reset occurred 1: RAMECC reset occurs	RW
b10	RAPERF	RAM parity check error reset flag	When a RAM parity error is reset, the hardware is set. Reset by setting CLRF. 0: No RAM parity error reset 1: RAM parity error reset	RW
b9	MPUERF	MPU error reset flag	Set by hardware when an MPU error reset occurs. Reset by setting CLRF. 0: No MPU error reset occurred 1: MPU error reset occurred	RW
b8	SWRF	Software reset flag	When software reset occurs, the hardware is set. Reset by setting CLRF. 0: No software reset occurred 1: Software reset	RW
b7	PDRF	Power-down mode reset	Set by hardware when a power-down reset occurs. Reset by setting. 0: Power-down mode reset has not occurred 1: Power-down mode reset occurred	RW
b6	SWDRF	Special watchdog reset flag	When a special watchdog is reset, the hardware is set. Reset by setting. 0: No special watchdog reset 1: Special watchdog resetting occurs	RW
b5	WDRF	watchdog reset flag	Set by hardware when a watchdog reset occurs. Reset by setting. 0: A watchdog reset has not occurred 1: Watchdog reset occurred	RW
b4	PVD2RF	Programmable voltage detection 2 Reset flag	Set by hardware when a programmable voltage sense 2 reset occurs. Reset by setting. 0: Programmable voltage detection 2 reset does not occur 1: Programmable voltage detection 2 reset occurs	RW
b3	PVD1RF	Programmable voltage detection 1 Reset flag	Set by hardware when a Programmable Voltage Detect 1 reset occurs. Reset by setting. 0: Programmable voltage detection 1 reset does not occur	RW

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
			1: Programmable voltage detection 1 reset occurs	
b2	BORF	Brown-out reset flag	Set by hardware when a brown-out reset occurs. Reset by setting. 0: Brown-out reset has not occurred 1: Brown-out reset occurred	RW
b1	PINRF	NRST pin reset flag	When pin reset occurs, it is set by hardware. Reset by setting. 0: No NRST reset occurred 1: NRST reset occurs	RW
b0	PORF	Power-on reset flag	When a power-on reset occurs, the hardware is set. Reset by setting. 0: No power-on reset 1: Power-on reset	RW

4 Clock Management Unit (CMU)

4.1 Overview

The clock control unit provides a series of frequency clock functions, including an external high-speed oscillator, an external low-speed oscillator, a PLL clock, an internal high-speed oscillator, an internal medium-speed oscillator, an internal low-speed oscillator, an SWDT dedicated internal low-speed oscillator, clock prescaler, clock multiplexing and clock gating circuits.

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

AHB, APB and Cortex-M4 clocks are all derived from the system clock. The maximum operating clock frequency of the system clock can reach 150MHz, and there are 6 selectable clock sources:

1) External high-speed oscillator (XTAL)
2) External low-speed oscillator (XTAL32)
3) PLLH Clock (PLLH)
4) Internal high-speed oscillator (HRC)
5) Internal medium speed oscillator (MRC)
6) Internal low-speed oscillator (LRC)

For each clock source, you can turn it on and off separately when not in use to reduce Power Consumption. SWDT has an independent clock source: SWDT dedicated internal low-speed oscillator (SWDTLRC). The real-time clock (RTC) uses an external low-speed oscillator, an internal low-speed oscillator, or an XTAL fractional clock as its clock source.

For each clock source, you can turn it on and off separately when not in use to reduce Power Consumption.

4.2 System Block Diagram

4.2.1 System Block Diagram

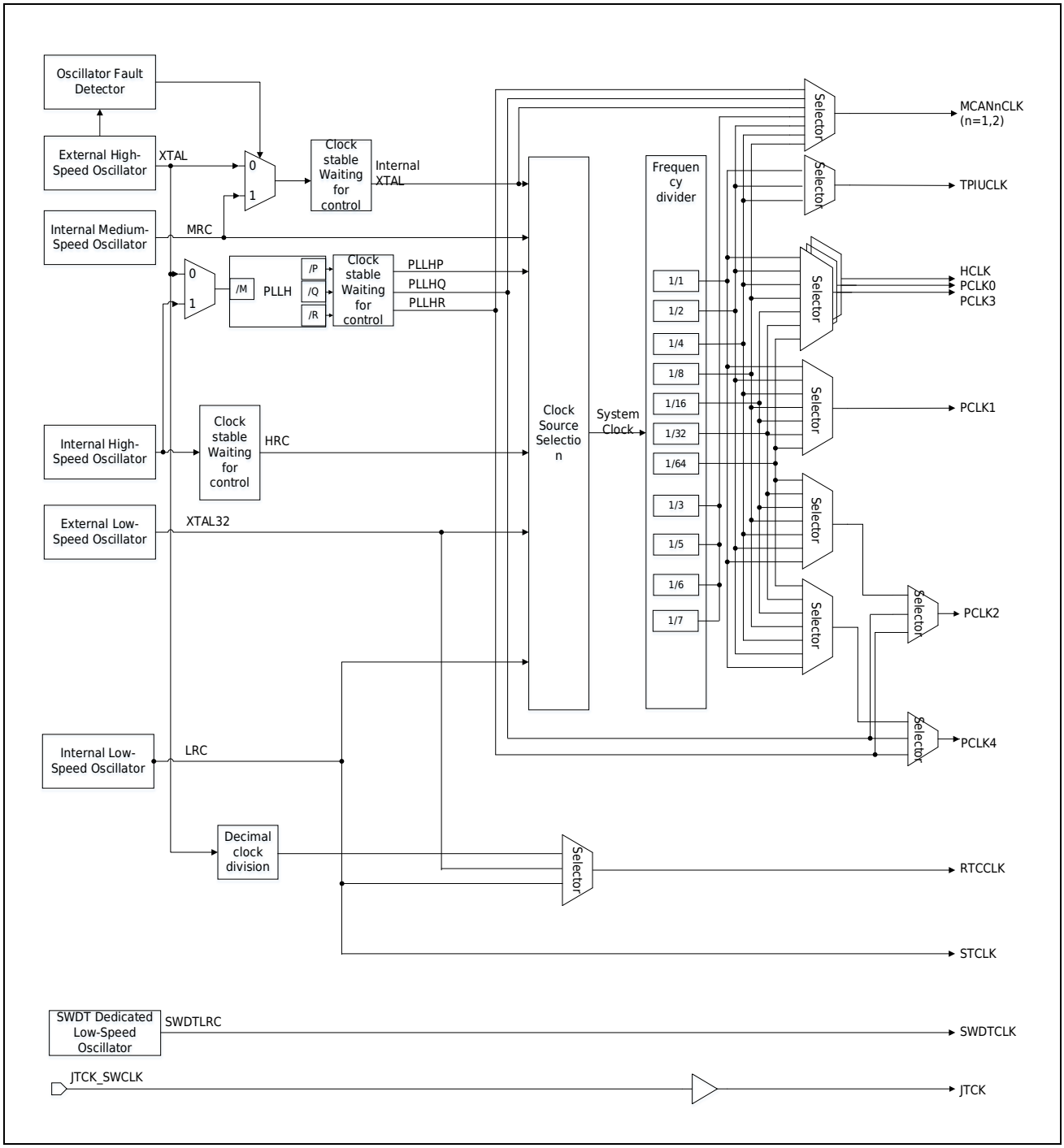


Figure 4-1 Clock System Block Diagram

4.2.2 Clock Frequency Measurement Block Diagram

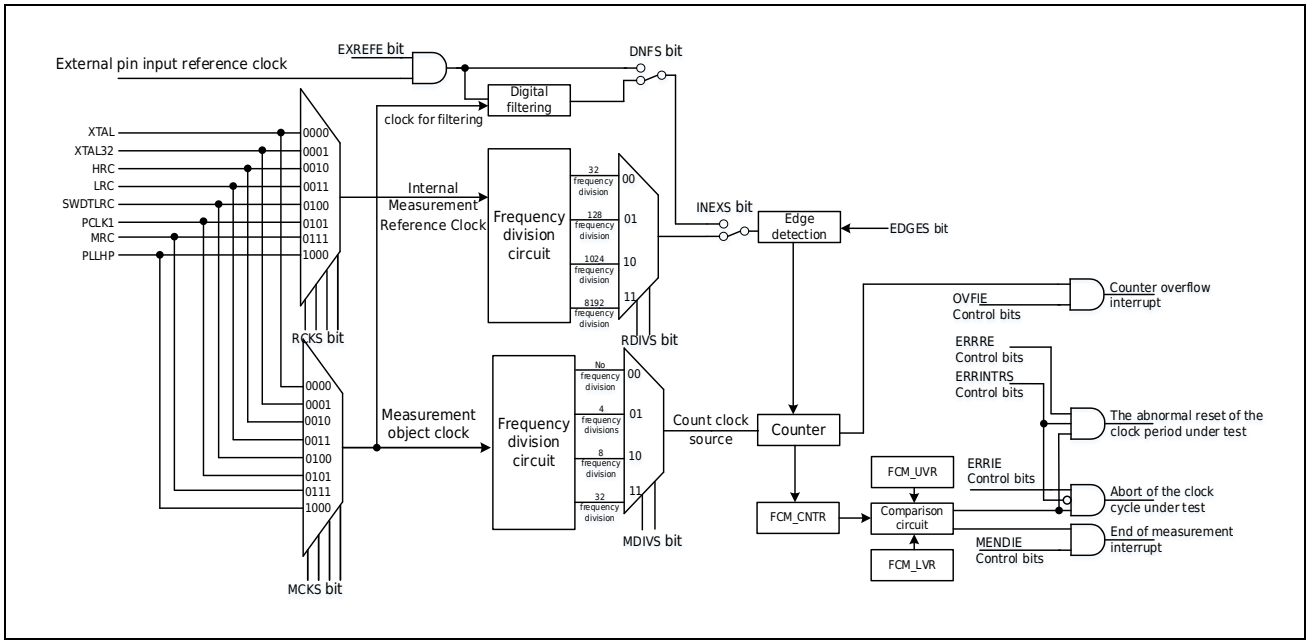


Figure 4-2 Clock Frequency Measurement Block Diagram

4.3 Clock Source Specification

The main characteristics of each clock source are shown in the following table.

Table 4-1 Clock Source Main Features

Clock Source	Specifications
External high-speed oscillator (XTAL)	Frequency range of crystal oscillator: 4~25MHz External clock input: Up to 25 MHz Oscillator Fault Detection Function
External low-speed oscillator (XTAL32)	Crystal oscillator frequency: 32.768KHz
PLLH Clock (PLLH)	Input clock: external high-speed oscillator or internal high-speed oscillator PLLH input clock frequency division: 1~25 arbitrary frequency division optional PFD input clock frequency = input clock/PLLH input clock frequency division, frequency range 1MHz~25MHz PLLH multiplication factor: 20~480 times VCO oscillation frequency: 240MHz~480MHz PLLHQ output frequency division ratio: 2~16 arbitrary frequency division PLLHP output frequency division ratio: 2~16 arbitrary frequency division PLLHR output frequency division ratio: 2~16 arbitrary frequency division PLLHP output frequency = (input clock / PLLH input clock frequency division) *PLLH frequency multiplication factor / PLLHP output frequency division ratio PLLHQ output frequency = (input clock / PLLH input clock frequency division) * PLLH frequency multiplication factor / PLLHQ output frequency division ratio PLLHR output frequency = (input clock / PLLH input clock frequency division) * PLLH frequency multiplication factor / PLLHR output frequency division ratio
Internal high-speed oscillator (HRC)	Frequency: 16MHz or 20MHz User-Writable Registers for Frequency Trimming
Internal Medium Speed Oscillator (MRC)	Frequency: 8 MHz User-Writable Registers for Frequency Trimming
Internal low-speed oscillator (LRC)	Frequency: 32.768 KHz User-Writable Registers for Frequency Trimming
SWDT dedicated internal low-speed oscillator (SWDTLRC)	Frequency: 10 KHz

4.4 Working Clock Specification

Table 4-2 Specifications of Each Internal Clock

Clock	Scope of Action	Specifications
HCLK	CPU, DMA, EFM, SRAM0, SRAMH, Ret SRAM, MPU, GPIO, INTC	Up to 150 MHz. Configured by the HCLKS bit in the CMU_SCFGR register. Selectable system clock division: 1, 2, 4, 8, 16, 32, 64
PCLK0	HRPWM_n (n=1~6), Timer6_n (n=1~4), Timer4, TimerA_n (n=1~4), EMB, I2C, CMP12, CMP3, MCANn (n=1~2) control logic	Up tp 150 MHz Configured by the PCLK0S bit in the CMU_SCFGR register. Selectable system clock division: 1, 2, 4, 8, 16, 32, 64
PCLK1	USARTn (n=1~4), SPI, Timer0_n (n=1~2), TimerA_5, CRC	Up tp 75 MHz Configured by the PCLK1S bit in the CMU_SCFGR register. Selectable system clock division: 1, 2, 4, 8, 16, 32, 64
PCLK2	ADC conversion clock	Up tp 60 MHz Configured by the PCLK2S bit in the CMU_SCFGR register and the PERICKSEL bit in the CMU_PERICKSEL register. Selectable system clock division: 1, 2, 4, 8, 16, 32, 64 Optional independent clock source: PLLHQ，PLLHR
PCLK3	RTC (control logic), WDT, SWDT (control logic), WKTM, FCM, CTC, PLA	Up tp 75 MHz Configured by the PCLK3S bit in the CMU_SCFGR register. Selectable system clock division: 1, 2, 4, 8, 16, 32, 64
PCLK4	ADCn (n=1~3) (control logic), DAC (control logic)	Up tp 75 MHz Configured by the PCLK4S bit in the CMU_SCFGR register and the PERICKSEL bit in the CMU_PERICKSEL register. Selectable system clock division: 1, 2, 4, 8, 16, 32, 64 Optional independent clock source: PLLHQ，PLLHR
MCANnCLK	MCAN1/ MCAN2 communication clock	Up tp 80 MHz Configured by the MCAN2CKS/MCAN1CKS bits in the CMU_CANCKCFGR register. The system clock division can be selected as 2, 3, 4, 5, 6, 7, or 8. Optional independent clock source: PLLHQ，PLLHR，XTAL
STCLK	SYSTICK Timer External Reference Clock	Internal low speed oscillator clock. CPU SYSTICK control and status CLKSOURCE bit configuration register SYSTICK timer clock.
SWDTCLK	SWDT Mcounter clock	Frequency 10 KHz
TCK	JTAG clock	Up tp 25 MHz
TPIUCLK	Clock for Cortex-M4 debug tracer	Up tp 30 MHz Configured by the TPIUCKS bit in the CMU_TPIUCKCFGR register. Selectable system clock division: 1, 2, 4

The following rules must be observed between the clocks:

- HCLK frequency ≥ PCLK0, PCLK1, PCLK2, PCLK3, PCLK4 frequency
- PCLK0 frequency≥PCLK1 frequency, PCLK0 frequency≥PCLK3 frequency
- PCLK2 frequency: PCLK4 frequency = 1:8, 1:4, 1:2, 1:1, 2:1, 4:1, 8:1
- When SWDT is used: PCLK3 frequency > SWDTCLK frequency

4.5 Crystal Oscillator Circuit

4.5.1 External High-speed Oscillator

4.5.1.1 Oscillator Mode

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

XTAL opens and closes through the XTALSTP bit of CMU _ XTALCR.

The XTALSTBF flag bit of CMU _ OSCSTBSR 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.

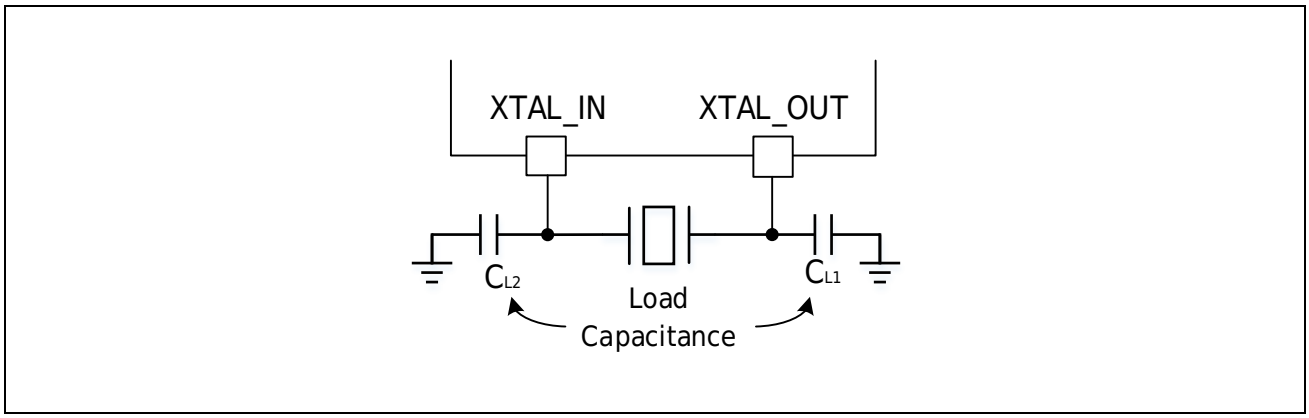


Figure 4-3 External High-speed Oscillator Connection Example

4.5.1.2 Clock Input Mode

External clock source must be provided in clock input mode. This mode is selected by XTALMS bit "1" of CMU_XTALCFGR and XTALSTP bit "0" of CMU_XTALCR. The XTAL_IN pin must be driven with an external clock signal with a duty cycle of approximately 50%. At this time, the XTAL_OUT pin can be configured as a GPIO according to the register settings.

The external clock input is connected as shown in the Figure 4-4.

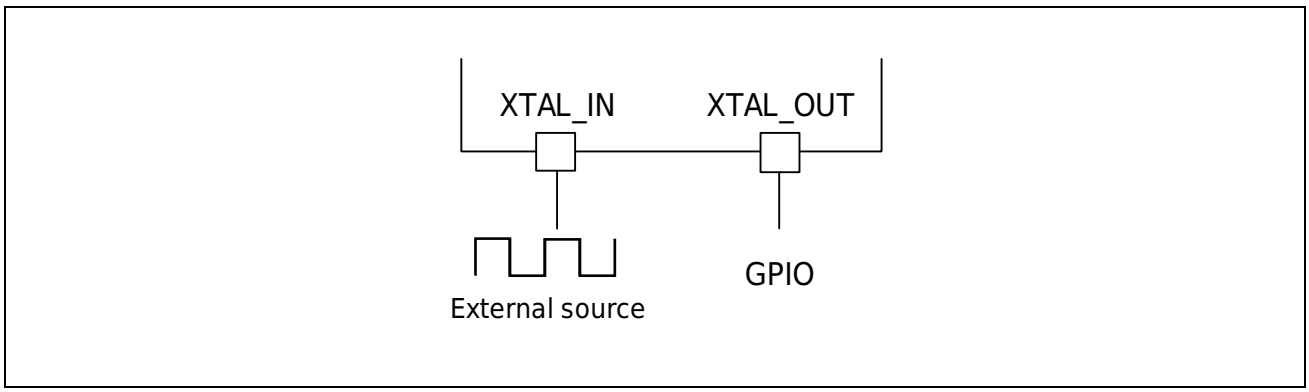


Figure 4-4 Example of Connection Diagram of External Clock Input

4.5.2 External High-speed Oscillator Fault Detection

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

Open or close by registerCMU _ XTALSTDCR XTALSTDE bit.

After the reset is removed, the external high-speed oscillator stops oscillating and the External High-speed Oscillator Fault Detection function is invalid. To enable the external high-speed oscillator failure detection function, you must enable the external high-speed oscillator to oscillate, and wait until the external high-speed oscillator is stable, that is, CMU_OSCSTBSR.XTALSTBF is 1, and turn it on through the XTALSTDE bit of the register CMU_XTALSTDCR.

When UPLL select XTAL clock as the input source, they can only select the XTAL oscillation fault to generate reset function.

Since the oscillator fault detection is to detect the oscillator abnormal oscillation caused by external factors, the oscillator fault detection function is disabled when the oscillator external high-speed oscillator is to be stopped or transferred to the Stop Mode and power-down mode by software.

If the external high-speed oscillator fails, the action waveform is shown in the Figure 4-5. Operation process reference [XTAL Fault Detection Action Detected].

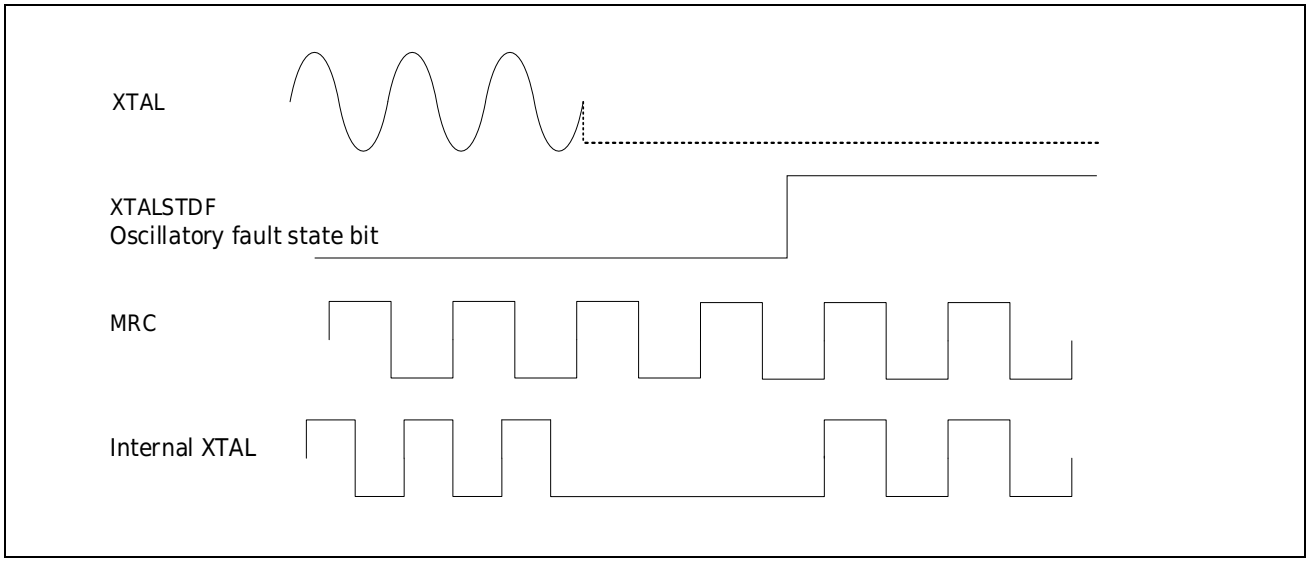


Figure 4-5 Example of External High-speed Oscillator Fault Detection

4.5.2.1 XTAL Fault Detection Action Detected

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

When an external high-speed oscillator oscillation failure is detected, EMB can be triggered to set the PWM output of Timer6/Timer4 to Hi-Z output. Refer to Chapter [Emergency Brake Module (EMB)].

The system clock is selected as XTAL. When an XTAL fault is detected, the action is shown in the Figure 4-6.

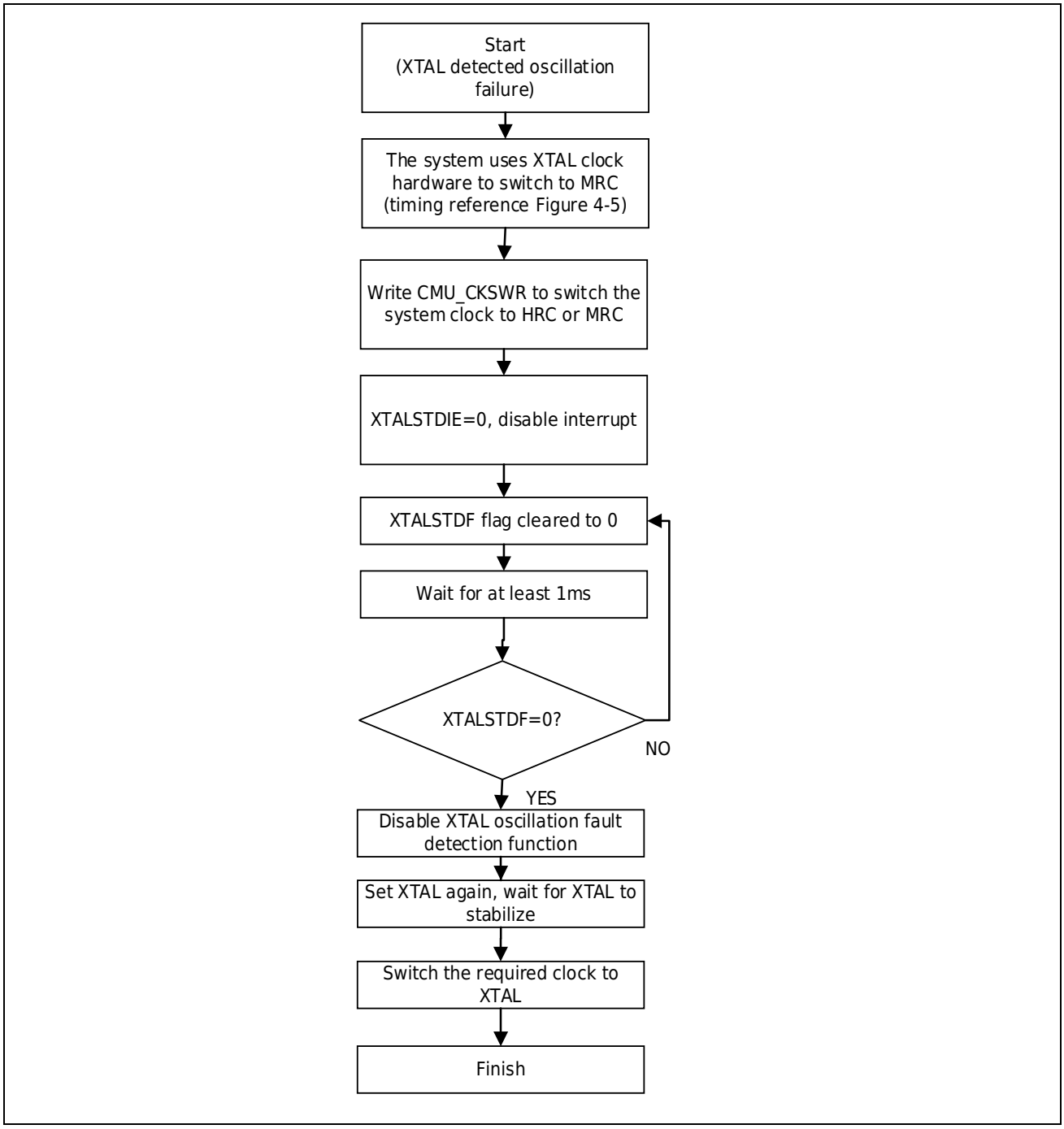


Figure 4-6 XTAL is Selected as the System Clock, and XTAL Oscillation Fault is Detected

4.5.2.2 XTAL Oscillation Fault Generated Termination Restoration Detected

XTAL oscillation fault interrupt can be configured as maskable interrupt or non-maskable interrupt, refer to chapter [Interrupt controller (INTC)].

When the XTAL oscillation fault is configured as reset, the XTAL oscillation fault is detected, and the chip resets. For the reset action, refer to Chapter [Reset Control (RMU)].

4.5.3 External High Speed Oscillator Fractional Frequency

By setting the CMU_XTALDIVR register, the external high-speed oscillator clock can be divided to 32.768KHz for use by the RTC.

Example of setting the external high-speed oscillator from 10MHz to 32.768KHz:

Frequency division number = = =

The NUMER[16:0] register is set to 0x1312D, the DENOM[10:0] register is set to 0x100, that is, the register CMU_XTALDIVR is set to 0x1312 D100, and then the CMU_XTALDIVCR setting is started.

Example of setting the external high-speed oscillator from 8MHz to 32.768KHz:

Frequency division number = = =

The NUMER[16:0] register is set to 0x07A12, the DENOM[10:0] register is set to 0x080, that is, the register CMU_XTALDIVR is set to 0x07A1 2080, and then the CMU_XTALDIVCR setting is started.

4.5.4 External Low-speed Oscillator

The 32.768KHz external low-speed oscillator can provide a more accurate clock source for the system clock and real-time clock circuit (RTC). It has the advantages of low Power Consumption and high precision.

XTAL32 is turned on and off by the XTAL32STP bit of CMU_XTAL32CR.

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.

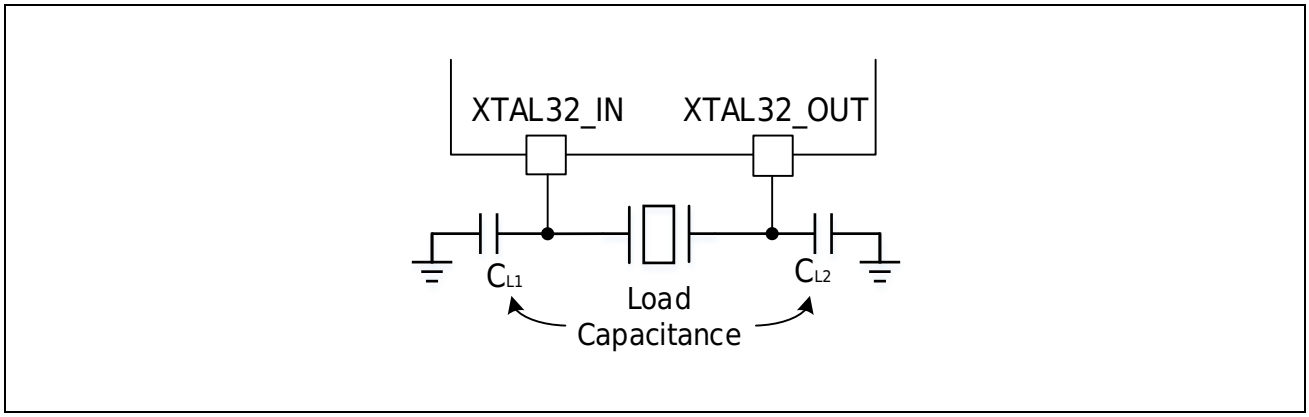


Figure 4-7 External Low-speed Oscillator Connection Example

The initialization process of XTAL32 power-on for the first time is as follows:

1. CMU_XTAL32CR.XTAL32STP bit write 1, stop XTAL32;
2. Set the matching XTAL32 drive capability through CMU_XTAL32FGR;
3. Set filter function by CMU_XTAL32NFR;
4. CMU_XTAL32CR.XTAL32STP bit write 0, XTAL32 oscillation;
5. The software waits for XTAL32 to stabilize. For the stabilization time, refer to the electrical features section of the corresponding data sheet.

If the external low-speed oscillator is not used, set the XTAL32STP bit of CMU_XTAL32CR to 1 to turn off the external low-speed oscillator.

4.6 Internal RC clock

4.6.1 HRC clock

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

HRC oscillator has the advantage of being less expensive (no external components needed). In addition, the start-up speed is higher than that of the XTAL crystal vibrator, but the accuracy is lower than that of the external crystal vibrator even after calibration.

Frequency Calibration

Because the production process is different, the RC oscillator frequency of different chips is also different, so each device will be factory calibrated to ensure accuracy. Refer to the **internal high-speed (HRC) oscillator chapter in the electrical characteristics of the Data Manual**.

If the application is subject to temperature variations, this may also affect the speed of the RC oscillator. Users can fine-tune the HRC frequency through the register.

The HRCSTBF flag in CMU _ OSCSTBSR indicates the stability of HRC. The HRC is not available until the hardware places this position at startup.

HRC can be turned on or off by the HRCSTP bit in the CMU_HRCCR register.

4.6.2 MRC Clock

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

The advantage of the MRC oscillator is that it starts up quickly.

Frequency Calibration

Because the production process is different, the RC oscillator frequency of different chips is also different, so each device will be factory calibrated to ensure accuracy. Refer to the **internal medium-speed (MRC) oscillator chapter in the electrical characteristics of the Data Manual**.

If the application is subject to temperature variations, this may also affect the speed of the RC oscillator. Users can fine-tune the MRC frequency through the register.

MRC can be turned on or off by the MRCSTP bit in the CMU_MRCCR register.

The MRC clock can also be used as a backup clock source in case the XTAL crystal fails. See [XTAL Fault Detection Action Detected].

4.6.3 LRC Clock

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

LRC oscillator can be started quickly .

Frequency Calibration

Because the production process is different, the RC oscillator frequency of different chips is also different, so each device will be factory calibrated to ensure accuracy. Refer to the **internal low-speed (LRC) oscillator chapter in the electrical characteristics of the Data Manua**l.

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

LRC can be turned on or off by the LRCSTP bits in the CMU _ LRCCR register.

4.6.4 SWDTLRC Clock

The SWDTLRC clock signal is generated by the internal 10KHz low-speed oscillator, and the SWDT exclusive clock. If SWDT has been started by ICG setting, the internal low-speed oscillator dedicated to SWDT will be forced to open and cannot be disabled.

Because the production process is different, the RC oscillator frequency of different chips is also different, so each device will be factory calibrated to ensure accuracy. Refer to the **SWDT dedicated internal low-speed (SWDTLRC) oscillator chapter in the electrical characteristics of the Data Manual**.

4.7 PLL Clock

The HC32F336 has one PLL:

The PLLH is clocked by the XTAL or HRC oscillator and has three different output clocks:

- P divider output for system clock generation (up to 150 MHz)
- All three outputs can be used to generate ADC, MCAN clocks

The PLL input source clock , and can choose HRC or XTAL oscillator as the clock source, which is configured by the CMU_PLLHCFGR.PLLSRC bit. After the HRC or XTAL oscillator is stable, configure the PLL.

The frequency division coefficients M, N, P, Q, R of PLLH can be configured independently. Since the PLL configuration parameters cannot be changed after the PLL is enabled, it is recommended to configure the PLL first and then enable it.

PLLs are disabled by hardware when entering power-down mode and stop mode.

4.8 Clock Switching Step

After the system is reset, the default system clock is MRC. Switch the clock source by setting the register CMU_CKSW, refer to the switching steps Clock Source Switch. You can switch from one clock source to another only if the target clock source is stable.

The waiting period of Flash/SRAM should be configured correctly to prevent the system clock frequency from being larger than the maximum operating frequency of Flash. Refer to the [Relationship between CPU clock and Flash read time] section for configuration.

4.8.1 Clock Source Switch

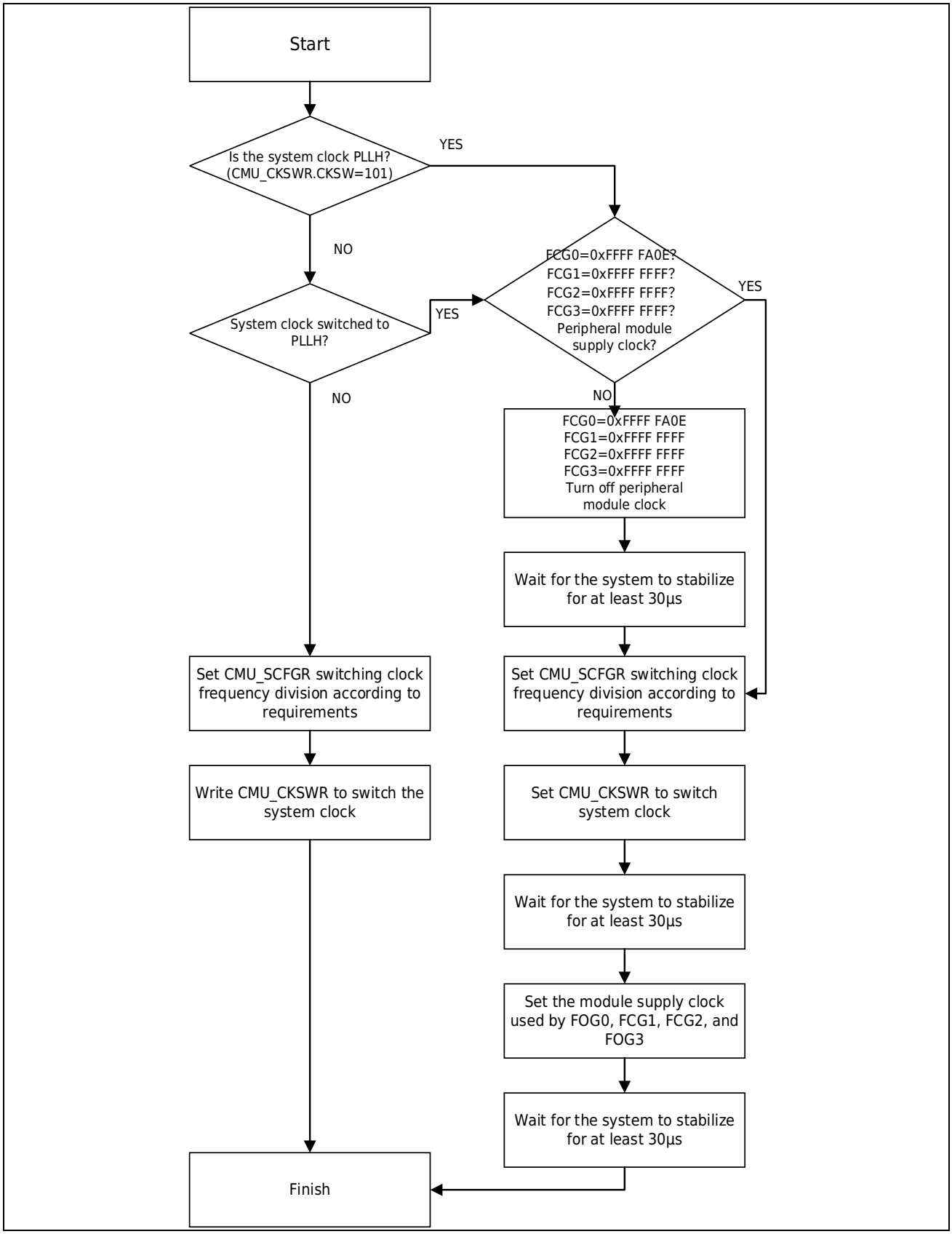


Figure 4-8 Clock Source Switching Procedure

4.8.2 Clock Divider Switching

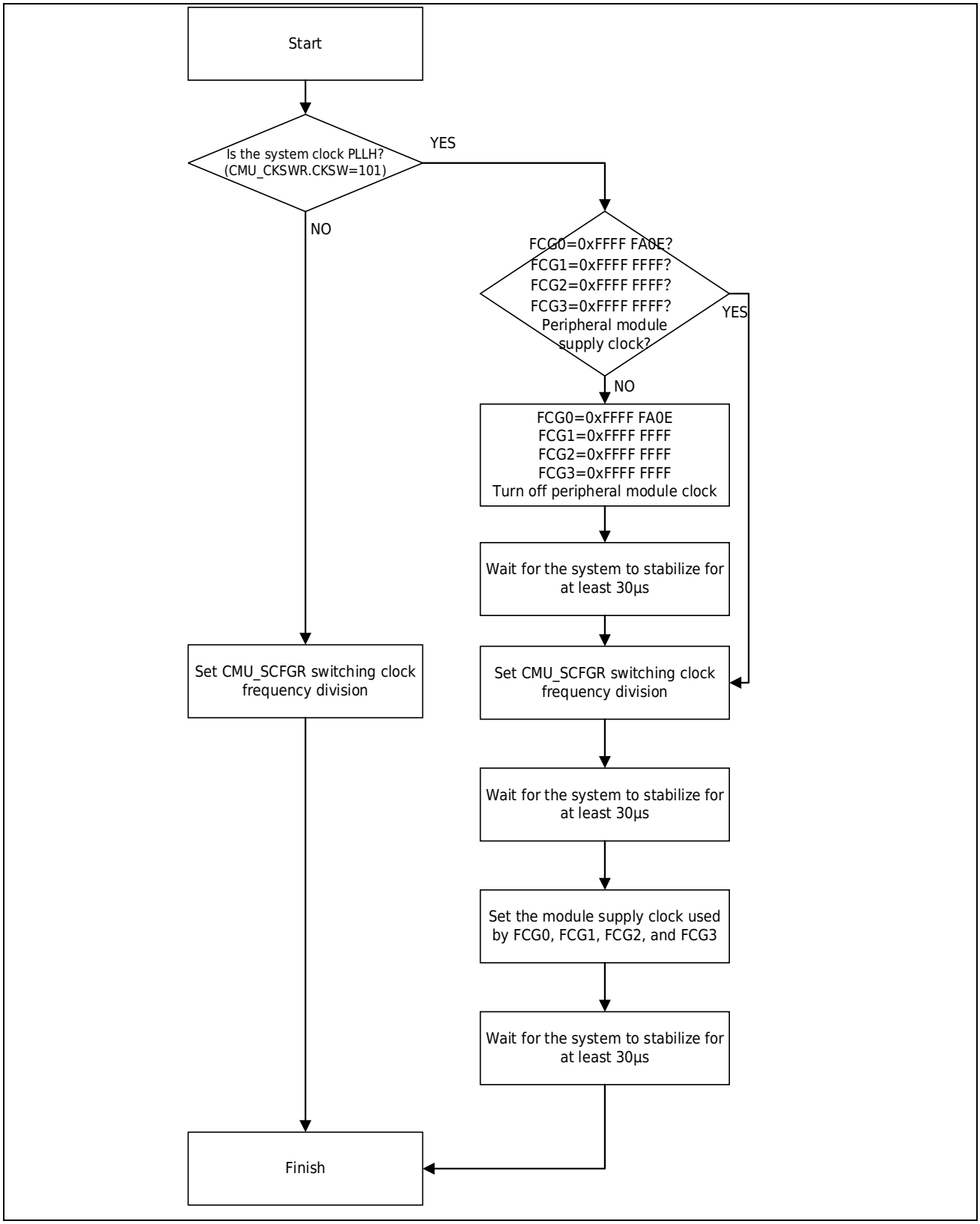


Figure 4-9 Clock Divider Switching

4.9 Clock Output Function

There are two clock outputs:

- MCO_1

The user can output different clock sources to the MCO_1 pin through a configurable pre-divider (1, 2, 4, 8, ..., 128 division):

  - HRC Clock
  - MRC Clock
  - LRC Clock
  - XTAL clock
  - XTAL32 clock
  - PLLHP/ PLLHQ clock
  - System clock

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

- MCO_2

The user can output different clock sources to the MCO_2 pin through a configurable pre-divider (1, 2, 4, 8, ..., 128 division):

  - HRC Clock
  - MRC Clock
  - LRC Clock
  - XTAL clock
  - XTAL32 clock
  - PLLHP/ PLLHQ clock
  - System clock

The desired clock source is selected by the CMU_MCO2CFGR.MCO2SEL bit.

The MCO_1/MCO_2 output clock must not exceed 100 MHz (maximum I/O speed).

4.10Clock Frequency Measurement

4.10.1 Clock Frequency Measurement

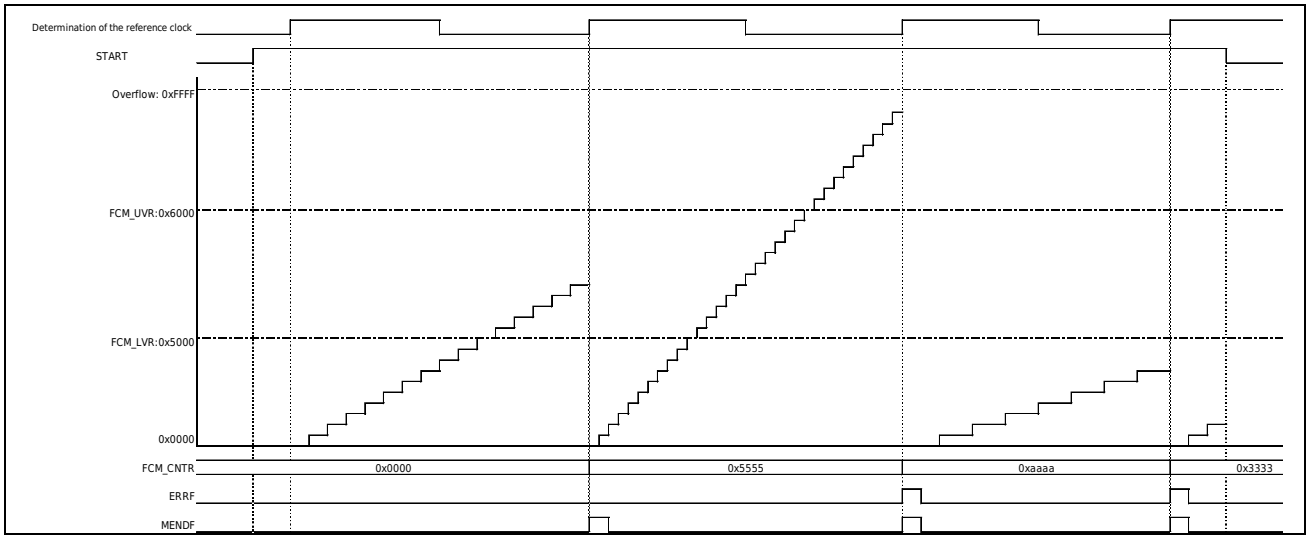


Figure 4-10 Clock Frequency Measurement Timing Diagram

1. Use FCM_MCCR/FCM_RCCR to select the reference clock to be measured, clock frequency division, and select the effective edge of the reference clock.
2. After writing 1 to the START bit of FCM_STR, the valid edge selected by the EDGES bit is detected, and the counter starts counting up.
3. When the valid 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 FCM_LVR<=FCM_CNTR<=FCM_UVR, the measured clock frequency measurement is normal. When FCM_LVR>FCM_CNTR or FCM_CNTR>FCM_UVR, the measured clock frequency is abnormal, and an interrupt or reset can occur according to the ERRINTRS/ERRRE/ERRIE settings.
4. After the START bit of FCM_STR is written to 0, the counter counting is stopped and cleared.

4.10.2 Digital Filtering Function

External pin input reference clock FCMREF has digital filtering function. The digital filter function performs three samplings according to the sampling clock selected by the DNFS bit, and when the level of the three samplings is the same, the level is sent to the inside.

The digital filtering function can set the digital filtering function to be valid or invalid and the sampling clock.

4.10.3 Interrupt/Reset Function

There are three types of interrupt requests for the clock frequency measurement circuit. They are:

1) Abnormal frequency interrupt
2) Frequency measurement end interrupt
3) Counter overflow interrupt

The clock frequency measurement circuit has a reset request:

Frequency abnormal reset

4.11Register Description

Table 4-3 CMU Register List

Base address 1: 0x4004 8400

Register	Description	Offset Address	Bit Width	Reset Value
FCM_LVR	FCM Lower Limit Compare Value Register	0x00	32	0x0000 0000
FCM_UVR	FCM Upper Limit Compare Value Register	0x04	32	0x0000 0000
FCM_CNTR	FCM Counter Value Register	0x08	32	0x0000 0000
FCM_STR	FCM start stop register	0x0C	32	0x0000 0000
FCM_MCCR	FCM Measurement Object Control Register	0x10	32	0x0000 0000
FCM_RCCR	FCM Measurement Reference Control Register	0x14	32	0x0000 0000
FCM_RIER	FCM interrupt reset control register	0x18	32	0x0000 0000
FCM_SR	FCM flags register	0x1C	32	0x0000 0000
FCM_CLR	FCM flag bit clear register	0x20	32	0x0000 0000

Base address 2: 0x4004 CC00

Register	Description	Offset Address	Bit Width	Reset Value
CMU_XTALCFGR	CMU XTAL configuration register	0x50	8	0x80
CMU_XTAL32CR	CMU XTAL32 Control Register	0x54	8	0x00
CMU_XTAL32CFGR	CMU XTAL32 Configuration Register	0x58	8	0x00
CMU_XTAL32NFR	CMU XTAL32 filter register	0x68	8	0x00
CMU_LRCCR	CMU LRC Control Register	0x6C	8	0x00
CMU_LRCTRM	CMU LRC Calibration Register	0x74	8	0x00

Base address 3: 0x4004 8000

Register	Description	Offset Address	Bit Width	Reset Value
CMU_XTALDIVR	CMU XTAL Fractional Configuration Register	0x18	32	0x03D0 9040
CMU_XTALDIVCR	CMU XTAL Fractional Control Register	0x1C	32	0x0000 0000

Base address4: 0x4005 4000

Register	Description	Offset Address	Bit Width	Reset Value
CMU_XTALSTBCR	CMU XTAL Stable Configuration Register	0x0A2	8	0x05
CMU_XTALCR	CMU XTAL Control Register	0x032	8	0x01
CMU_XTALSTDCR	CMU XTAL Oscillatory Fault Control Register	0x040	8	0x00
CMU_XTALSTDSR	CMU XTAL Oscillatory Fault States	0x041	8	0x00
CMU_HRCCR	CMU HRC Control Register	0x036	8	Determined by ICG1.HRCSTP value
CMU_HRCTRM	CMU HRC Calibration Register	0x062	8	0x00
CMU_MRCCR	CMU MRC Control Register	0x038	8	0x80
CMU_MRCTRM	CMU MRC Calibration Register	0x061	8	0x00
CMU_PLLHCFGR	CMU PLLH configuration register	0x100	32	0x1110 1300
CMU_PLLHCR	CMU PLLH Control Register	0x02A	8	0x01
CMU_OSCSTBSR	CMU Timer stability register	0x03C	8	0x00
CMU_CKSWR	CMU System timer switchover register	0x026	8	0x01
CMU_SCFGR	CMU Clock Divider Configuration Register	0x020	32	0x0000 0000
CMU_CANCKCFGR	CMU CAN Clock Configuration Register	0x018	8	0xDD
CMU_PERICKSEL	CMU CMU/ DA Clock Configuration Register	0x010	16	0x0000
CMU_TPIUCKCFGR	CMU Debug clock config register	0x03F	8	0x00
CMU_MCO1CFGR	CMU MCO1 Clock Output Configuration Register	0x03D	8	0x00
CMU_MCO2CFGR	CMU MCO2 Clock Output Configuration Register	0x03E	8	0x00

4.11.1 CMU XTAL Configuration Register (CMU _ XTALCFGR)

Reset value: 0x80

b7	b6	b5	b4	b3	b2	b1	b0
Res		XTALDRV[1:0]		XTALMS	Res		

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b7	Res	Reserved Bit	Write "1" on write	W
b6	Res	Reserved Bit	Write "0" on write	RW
b5~b4	XTALDRV[1:0]	XTAL Driving Ability Selection	00: High drive capability (recommended 20~ 25 MHz crystal oscillator) 01: Medium-drive capability (recommended 16 ~ 20 MHz crystal oscillator) 10: Low drive capability (recommended 8-16 MHz crystal) 11: Ultra-low drive capability (recommended 4~ 8 MHz crystal oscillator)	RW
b3	XTALMS	XTAL mode selection bit	0: Oscillator mode 1: External clock input mode	RW
b2~b0	Res	Reserved Bit	Read at “0”, write at “0”	RW

4.11.2 CMU XTAL Stability Configuration Register (CMU_XTALSTBCR)

Reset Value: 0x05

b7	b6	b5	b4	b3	b2	b1	b0
Res				XTALSTB[3:0]			

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b7~b4	Res	Reserved Bit	Read at “0”, write at “0”	RW
b3~b0	XTALSTB[3:0]	XTAL Stability Time Selection	0001: Stable counter 35 cycles 0010: Stable counter 67 cycles 0011: Stable counter 131 cycles 0100: Stable counter 259 cycles 0101: Stable counter 547 cycles 0110: Stable counter 1059 cycles 0111: Stable counter 2147 cycles 1000: Stable counter 4291 cycles 1001: Stable counter 8163 cycles A counting period of a stable counter = LRC period / 8 This register is configured with CMU _ XTALCR.XTALSTP bit 1 and CMU _ OSCSTBSR.XTALSTBF bit 0.	RW

4.11.3 CMU XTAL Fractional Divider Configuration Register (CMU_XTALDIVR)

Reset value: 0x03D0 9040

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res				NUMER[16:4]											
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
NUMER[3:0]				Res	DENOM[10:0]										

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b29	Res	Reserved Bit	Read at “0”, write at “0”	RW
b28~b12	NUMER[16:0]	XTAL Fractional Frequency Numerator Setting Value	XTAL fractional frequency numerator setting value, for application method, refer to the chapter [External High Speed Oscillator Fractional Frequency]	RW
b11	Res	Reserved Bit	Read at “0”, write at “0”	RW
b10~b0	DENOM[10:0]	XTAL fractional frequency denominator setting value	XTAL decimal frequency denominator setting value, for application method, please refer to the [External High Speed Oscillator Fractional Frequency] section. Please set this value to 2^N. For example, 0x100 means the denominator is 256. 0x80 means the denominator is 128.	RW

4.11.4 CMU XTAL Fractional Divider Control Register (CMU_XTALDIVCR)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res															FRADIVEN

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b1	Res	Reserved Bit	Read at “0”, write at “0”	RW
b0	FRADIVEN	XTAL decimal division starts	0: XTAL fractional frequency stop 1: XTAL fractional division starts	RW

4.11.5 CMU XTAL Control Register (CMU _ XTALCR)

Reset value: 0x01

b7	b6	b5	b4	b3	b2	b1	b0
Res							XTALSTP

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b7~b1	Res	Reserved Bit	Read at “0”, write at “0”	RW
b0	XTALSTP	XTAL oscillator open stop bit	0: XTAL oscillator starts 1: XTAL oscillator stops	RW

Note:

1. *When XTAL is selected as system clock or PLLH clock source, XTALSTP is prohibited to write 1 to stop XTAL oscillator.*
2. *The XTAL oscillator is set by software to oscillate. After confirming that the XTAL oscillator is stable through the XTALSTBF bit, it can enter Stop Mode, power-down mode or stop the XTAL oscillator by software setting.*
3. *The XTAL oscillator is set to stop by software, and the XTAL oscillator can be entered into Stop Mode, power-down mode or restarted after confirming that the XTAL oscillator is stopped through the XTALSTBF bit.*

4.11.6 CMU XTAL Oscillatory Fault Control Register (CMU _ XTALSTDCR)

Reset value: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
XTALSTDE	Res				XTALSTDRIS	XTALSTDRE	XTALSTDIE

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b7	XTALSTDE	XTAL Oscillatory Fault Detection Function Allowed	0: Disable XTAL Oscillation Fault Detection 1: Allow XTAL Oscillation Fault Detection **Note:** Oscillator fault detection is to detect the oscillator abnormal oscillation caused by external factors. Please disable the oscillator oscillation fault detection function before entering Stop Mode or power-down mode.	RW
b6~b3	Res	Reserved Bit	Read at “0”, write at “0”	RW
b2	XTALSTDRIS	XTAL Oscillatory Fault Reset Interrupt Selection	0: XTAL OSCILLATION FAULT GENERATES TERATE 1: XTAL Oscillation Fault Reset **Note:** When UPLL select XTAL clock as the input source, they can only select the XTAL oscillation fault to generate reset function.	RW
b1	XTALSTDRE	XTAL Oscillatory Fault Reset Allowance	0: Disable XTAL Oscillatory Fault Reset 1: Allow XTAL Oscillation Fault Reset	RW
b0	XTALSTDIE	XTAL Oscillation Fault Interrupt Allowance	0: Disable XTAL OSCILLATION FAULT TERMINATION 1: Allow XTAL OSCILLATION FAULT TERMINATION Set the PWM output of Timer6/Timer4 to Hi-Z output through EMB, and the XTALSTDIE bit needs to be set to 1.	RW

Note:

– *When XTAL is selected as system clock or PLLH clock source, XTALSTP is prohibited to write 1 to stop XTAL oscillator.*

4.11.7 CMU XTAL Oscillatory Fault State Register (CMU _ XTALSTDSR)

Reset value: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
Res							XTALSTDF

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b7~b1	Res	Reserved Bit	Read at “0”, write at “0”	RW
b0	XTALSTDF	XTAL Oscillatory Fault State Bit	0: No XTAL oscillation fault detected 1: XTAL Oscillation Fault Detected Setting conditions: XTAL OSCILLATION FAULT UNDER XTALSTDE = 1 Reset condition: When the system clock selects a clock other than XTAL, read 1 and write 0.	RW

4.11.8 CMU XTAL32 Configuration Register (CMU_XTAL32CFGR)

Reset value: 0xXX

b7	b6	b5	b4	b3	b2	b1	b0
Res					XTAL32DRV[2:0]		

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b7~b3	Res	Reserved Bit	Read at “0”, write at “0”	RW
b2~b0	XTAL32DRV[2:0]	XTAL32 drive capability selection	000: Medium drive capability 001: Large drive capacity Other: prohibitions **Note:** For application method, refer to the **Electrical Features Section of Data Sheet** [Low-speed external clock generated by crystal oscillator/ceramic resonator]	RW

4.11.9 CMU XTAL32 Filter Register (CMU_XTAL32NFR)

Reset value: 0xXX

b7	b6	b5	b4	b3	b2	b1	b0
Res						XTAL32NF[1:0]	

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b7~b2	Res	Reserved Bit	Read at “0”, write at “0”	RW
b1~b0	XTAL32NF[1:0]	XTAL32 Oscillator Filter Selection	00: RUN mode/Stop Mode/power down mode, 3μs filter of XTAL32 is valid 01: The 3μs filter of XTAL32 is valid in RUN mode, and the 3μs filter of XTAL32 in Stop Mode or power-down mode is invalid 10: Prohibit setting 11: RUN mode/Stop Mode/power down mode, the 3μs filter of XTAL32 is invalid	RW

4.11.10 CMU XTAL32 Control Register (CMU_XTAL32CR)

Reset value: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
Res							XTAL32STP

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b7~b1	Res	Reserved Bit	Read at “0”, write at “0”	RW
b0	XTAL32STP	XTAL32 oscillator start stop bit	0: XTAL32 oscillator oscillates 1: XTAL32 oscillator stopped	RW

Note:

1. *When XTAL32 is selected as the system clock source, XTAL32STP is prohibited from writing 1 to stop the XTAL32 oscillator.*
2. *The software sets the XTAL32 action to start, and waits for 5 XTAL32 cycles before stopping the XTAL32 again.*
3. *The software sets the XTAL32 to stop, and waits for 5 XTAL32 cycles before starting the XTAL32 again.*

4.11.11 CMU HRC Calibration Register (CMU_HRCTRM)

Reset value: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
HRCTRIM[7:0]							

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b7~b0	HRCTRIM[7:0]	HRC frequency calibration bit	Frequency calibration needs to be within the guaranteed range of HRC frequency. 10000000：-128 10000001：-127 ..... 11111111：-1 00000000: Center Code 00000001：+1 ..... 01111110：+126 01111111：+127	RW

Note:

– *The frequency calibration needs to be within the HRC frequency guaranteed range.*

4.11.12 CMU HRC Control Register (CMU _ HRCCR)

Reset value: determined by ICG1.HRCSTP value

b7	b6	b5	b4	b3	b2	b1	b0
Res							HRCSTP

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b1	Res	Reserved Bit	Read at “0”, write at “0”	RW
b0	HRCSTP	HRCoscillator open stop bit	0: HRCoscillator Oscillation 1: HRCoscillator stops According to ICG1.HRCSTOP configuration, HRC starts to stop after reset.	RW

Note:

1. *When HRC is selected as the system clock source or PLLH clock source, it is forbidden to write 1 in CMU_HRCCR.HRCSTP to stop the HRC clock.*
2. *The HRC oscillator is set by software, and the HRC can be entered into Stop Mode, power-down mode or stop HRC only after confirming that the HRC is stable through the HRCSTBF bit.*
3. *The software sets the HRC to stop, and the HRC can be stopped after confirming the MPLL stop through the HRCSTBF bit before entering the Stop Mode, power-down mode or restarting the HRC.*

4.11.13 CMU MRC Calibration Register (CMU_MRCTRM)

Reset value: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
MRCTRIM[7:0]							

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b7~b0	MRCTRIM[7:0]	MRC Frequency Calibration Bits	10000000: -128 10000001: -127 ..... 11111111: -1 00000000: Center Code 00000001: +1 ..... 01111110: +126 01111111: +127	RW

Note:

– *The frequency calibration needs to be within the MRC frequency guaranteed range.*

4.11.14 CMU MRC Control Register (CMU_MRCCR)

Reset value: 0x80

b7	b6	b5	b4	b3	b2	b1	b0
Res							MRCSTP

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b7	Res	Reserved Bit	Read at “1”, write at “1”	RW
b6~b1	Res	Reserved Bit	Read at “0”, write at “0”	RW
b0	MRCSTP	MRC oscillator open stop bit	0: MRC oscillator starts 1: MRC oscillator stops **Note:** 1) When the XTAL oscillation fault function is valid, this bit is cleared to 0 at the same time, and MRC oscillates. 2) The Stop Mode wake-up action when the PWC_STPMCR.CKSMRC bit is 1 is set when the MRC oscillator is in the oscillating state.	RW

Note:

1. *When MRC is selected as system timer, MRCSTP writes 1 to stop MRC clock.*
2. *The software sets the MRC oscillation and waits for 5 MRC cycles before entering Stop Mode, power-down mode or stopping MRC.*
3. *The software sets the MRC to stop, and waits for 5 MRC cycles before entering the Stop Mode, power-down mode or restarting the MRC.*
4. *MRC is used as RTC calibration clock. Possibility of MRC oscillation when RTC is not initialized.* When the *RTC calibration function is enabled,* the *MRC oscillates regardless of the MRCSTP bit setting.*

4.11.15 CMU LRC Calibration Register (CMU_LRCTRM)

Reset value: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
LRCTRIM[7:0]							

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b7~b0	LRCTRIM[7:0]	LRC Frequency Calibration Bits	10000000：-128 10000001：-127 ..... 11111111：-1 00000000: Center Code 00000001：+1 ..... 01111110：+126 01111111：+127	RW

Note:

– *The frequency calibration needs to be within the LRC frequency guaranteed range.*

4.11.16 CMU LRC Control Register (CMU _ LRCCR)

Reset value: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
Res							LRCSTP

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b7~b1	Res	Reserved Bit	Read at “0”, write at “0”	RW
b0	LRCSTP	LRCoscillator open stop bit	0: LRCoscillator Oscillation 1: LRCoscillator stops	RW

Note:

1. *When LRC is selected as system timer, LRCSTP writes 1 to stop LRC clock.*
2. *The software sets the LRC oscillation and waits for 5 LRC cycles before entering Stop Mode, power-down mode or stopping LRC.*
3. *The software sets the LRC to stop, and waits for 5 LRC cycles before entering the Stop Mode, power-down mode or restarting the LRC.*
4. *When waiting for the XTAL oscillator, HRC, MPLL clocks to be stable, the LRCSTP bit setting is ignored, and the LRC is forced to oscillate.*

4.11.17 CMU PLLH Configuration Register (CMU_PLLHCFGR)

Reset value: 0x1110 1300

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
PLLHP[3:0]				PLLHQ[3:0]				PLLHR[3:0]				Res			PLLHN[8]
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
PLLHN[7:0]								PLLSRC	Res		PLLHM[1:0]				

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b28	PLLHP[3:0]	PLLH frequency division factor for system clock	Frequency used for PLLHP clock, write to PLLHP during PLLH stop condition. PLLH output clock frequency = PLLH VCO frequency / PLLHP 0000: Prohibit setting 0001: 2 frequency division 0010: 3 frequency division 0011: 4 frequency division ..... 1101: 14 frequency division 1110: 15 frequency division 1111: 16 frequency division	RW
b27~b24	PLLHQ[3:0]	PLLH frequency division factor for system clock	Frequency used for PLLHQ clock, write to PLLHQ during PLLH stop condition. PLLH output clock frequency = PLLH VCO frequency / PLLHQ 0000: Prohibit setting 0001: 2 frequency division 0010: 3 frequency division 0011: 4 frequency division ..... 1101: 14 frequency division 1110: 15 frequency division 1111: 16 frequency division	RW
b23~b20	PLLHR[3:0]	PLLH frequency division factor for system clock	Frequency used for PLLHR clock, write to PLLHR during PLLH stop condition. PLLH output clock frequency = PLLH VCO frequency / PLLHR 0000: Prohibit setting 0001: 2 frequency division 0010: 3 frequency division 0011: 4 frequency division ..... 1101: 14 frequency division 1110: 15 frequency division 1111:16 frequency division	RW
b19~b17	Res	Reserved Bit	Read at “0”, write at “0”	RW
b16~b8	PLLHN[8:0]	PLLH multiplication factor	It is used to control the VCO multiplier coefficient of PLLH, write PLLHN in PLLH stop condition. Make sure the VCO frequency of the PLLH is between 240MHz and 480MHz. VCO frequency of PLLH = VCO input frequency of PLLH * PLLHN 000010011：20 000010100：21 000010101：22 000010110：23 ..... 111011101：478 111011110：479 111011111：480	RW
b7	PLLSRC	PLLH input clock source selection	0: Select the external high-speed oscillator as the input clock of PLLH 1: Select the internal high-speed oscillator as the input clock of PLLH	RW
b6~b5	Res	Reserved Bit	Read at “0”, write at “0”	RW
b4~b0	PLLHM[4:0]	PLLH input clock frequency division factor	Used to divide the PLLH input clock before the PLLH's VCO. Write PLLHM in PLLH STOP condition. Make sure the PFD input clock frequency to the PLLH is between 1MHz and 25MHz. 00000: 1 frequency division	RW

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
			00001: 2 frequency division 00010: 3 frequency division ..... 10111: 24 frequency division Other: prohibitions	

4.11.18 CMU MPLL Control Register (CMU_PLLHCR)

Reset value: 0x01

b7	b6	b5	b4	b3	b2	b1	b0
Res							PLLHOFF

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b7~b1	Res	Reserved Bit	Read at “0”, write at “0”	RW
b0	PLLHOFF	PLLH enable	Used to start and stop PLLH. Do not set this bit to 1 if PLLH clock is used as system clock. 0: PLLH operation starts 1: PLLH stop	RW

Note:

1. *When PLLH is selected as the system clock source, it is forbidden to write 1 to PLLHOFF to stop the PLLH clock.*
2. *The software sets the PLLH action to start, and after the PLLHSTBF bit confirms that the PLLH is stable, it can enter the Stop Mode, power-down mode or software setting to stop the PLLH.*
3. *The software sets the PLLH to stop, and the PLLH can be entered into the Stop Mode, power-down mode or start the PLLH again after confirming that the MPLL is stopped through the PLLHSTBF bit.*
4. *When PLLH selects the XTAL oscillator as the clock source, the PLLH operation can be set to start after confirming that the XTAL oscillator is stable through the XTALSTBF bit. When PLLH selects HRC as the clock source, the PLLH action can be set to start after confirming that HRC is stable through the HRCSTBF bit.*

4.11.19 CMU Timer Stabilizer (CMU _ OSCSTBSR)

Reset value: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
Res		PLLHSTBF	Res	XTALSTBF	Res		HRCSTBF

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b7~b6	Res	Reserved Bit	Read at “0”, write at “0”	R
b5	PLLHSTBF	PLLH stability sign bit	0: PLLH stopped or unstabilized 1: PLLH stabilization	R
b4	Res	Reserved Bit	Read at “0”, write at “0”	R
b3	XTALSTBF	XTAL stability sign bit	0: XTAL stopped or unstabilized 1: XTAL Stabilization	R
b2~b1	Res	Reserved Bit	Read at “0”, write at “0”	R
b0	HRCSTBF	HRC stable sign bit	0: HRC stopped or unstabilized 1: HRC stability	R

4.11.20 Timer Switch Register of CMU System (CMU _ CKSWR)

Reset value: 0x01

b7	b6	b5	b4	b3	b2	b1	b0
Res					CKSW[2:0]		

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b7~b3	Res	Reserved Bit	Read at “0”, write at “0”	RW
b2~b0	CKSW[2:0]	System timer switchover	000: Select HRC clock as system clock 001: Select the MRC clock as the system clock 010: Select the LRC clock as the system clock 011: Select the XTAL clock as the system clock 100: Select the XTAL32 clock as the system clock 101: Select PLLH as the system clock 110: Setting prohibited 111: Setting prohibited **Note:** 1. Switch the target clock to ensure the clock is stable. 2. Refer to the [Clock Source Switch] chapter for the process 3. When the PWC_STPMCR.CKSMRCbit is 1, after the Stop Mode wakes up, this register initializes and the system timer selects the MRC clock.	RW

4.11.21 CMU Clock Divider Configuration Register (CMU_SCFGR)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res					HCLKS[2:0]			Res					PCLK4S[2:0]		
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res	PCLK3S[2:0]			Res	PCLK2S[2:0]			Res	PCLK1S[2:0]			Res	PCLK0S[2:0]		

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b27	Res	Reserved Bit	Read at “0”, write at “0”	RW
b26~b24	HCLKS[2:0]	HCLK clock frequency division selection bit	000: 1 frequency division of system clock 001: 2-Division Frequency of System Clock 010: 4-division frequency of the system clock 011: 8 frequency division of system clock 100: 16-division frequency of system clock 101: 32 frequency division of system clock 110: 64-division frequency of system clock 111: Setting prohibited **Note:** When the PWC_STPMCR.CKSMRC bit is 1, after wake-up from Stop Mode, this register is initialized, and HCLK is divided by 1 of the system clock.	RW
b23~b19	Res	Reserved Bit	Read at “0”, write at “0”	RW
b18~b16	PCLK4S[2:0]	PCLK4 clock frequency division selection bit	000: 1 frequency division of system clock 001: 2-Division Frequency of System Clock 010: 4-division frequency of the system clock 011: 8 frequency division of system clock 100: 16-division frequency of system clock 101: 32 frequency division of system clock 110: 64-division frequency of system clock 111: Setting prohibited **Note:** When the PWC_STPMCR.CKSMRC bit is 1, this register is initialized after wake-up from Stop Mode, and PCLK4 is divided by 1 of the system clock.	RW
b15	Res	Reserved Bit	Read at “0”, write at “0”	RW
b14~b12	PCLK3S[2:0]	PCLK3 clock frequency division selection bit	000: 1 frequency division of system clock 001: 2-Division Frequency of System Clock 010: 4-division frequency of the system clock 011: 8 frequency division of system clock 100: 16-division frequency of system clock 101: 32 frequency division of system clock 110: 64-division frequency of system clock 111: Setting prohibited **Note:** When the PWC_STPMCR.CKSMRC bit is 1, this register is initialized after wake-up from Stop Mode, and PCLK3 is divided by 1 of the system clock.	RW
b11	Res	Reserved Bit	Read at “0”, write at “0”	RW
b10~b8	PCLK2S[2:0]	PCLK2 clock frequency division selection bit	000: 1 frequency division of system clock 001: 2-Division Frequency of System Clock 010: 4-division frequency of the system clock 011: 8 frequency division of system clock 100: 16-division frequency of system clock 101: 32 frequency division of system clock 110: 64-division frequency of system clock 111: Setting prohibited **Note:** When the PWC_STPMCR.CKSMRC bit is 1, this register is initialized after wake-up from Stop Mode, and PCLK2 is divided by 1 of the system clock.	RW
b7	Res	Reserved Bit	Read at “0”, write at “0”	RW
b6~b4	PCLK1S[2:0]	PCLK1 clock frequency division selection bit	000: 1 frequency division of system clock 001: 2-Division Frequency of System Clock 010: 4-division frequency of the system clock 011: 8 frequency division of system clock	RW

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
			100: 16-division frequency of system clock 101: 32 frequency division of system clock 110: 64-division frequency of system clock 111: Setting prohibited **Note:** When the PWC_STPMCR.CKSMRC bit is 1, this register is initialized after wake-up from Stop Mode, and PCLK1 is divided by 1 of the system clock.	
b3	Res	Reserved Bit	Read at “0”, write at “0”	RW
b2~b0	PCLK0S[2:0]	PCLK0 clock frequency division selection bit	000: 1 frequency division of system clock 001: 2-Division Frequency of System Clock 010: 4-division frequency of the system clock 011: 8 frequency division of system clock 100: 16-division frequency of system clock 101: 32 frequency division of system clock 110: 64-division frequency of system clock 111: Setting prohibited **Note:** When the PWC_STPMCR.CKSMRC bit is 1, this register is initialized after wake-up from Stop Mode, and PCLK0 is divided by 1 of the system clock.	RW

4.11.22 CMU CAN Clock Configuration Register (CMU_CANCKCFGR)

Reset value: 0xDD

b7	b6	b5	b4	b3	b2	b1	b0
MCAN2CKS[3:0]				MCAN1CKS[3:0]			

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b7~b4	MCAN2CKS[3:0]	MCAN2 Communication Clock Selection	0001: system clock divided by 2 0010: system clock divided by 3 0011: System clock divided by 4 0100: System clock divided by 5 0101: System clock divided by 6 0110: System clock divided by 7 0111: System clock divided by 8 1000：PLLH/Q 1001：PLLH/R 1101：XTAL Other: prohibitions **Note:** 1. When the switching target clock source is PLLH, ensure that the PLLH clock is in a stable state. 2. When the system clock selects PLLH, it is necessary to turn off the clocks of each peripheral module, and then write the CMU_SCFGR register to switch the clock division. After writing the CMU_CANCKCFGR register, the software waits for the system to stabilize for at least 30us. 3. When the PWC_STPMCR.CKSMRC bit is 1, after waking up from Stop Mode, this register is initialized and MCANCLK is the XTAL clock source.	RW
b3~b0	MCAN1CKS[3:0]	MCAN1 Communication Clock Selection	0001: system clock divided by 2 0010: system clock divided by 3 0011: System clock divided by 4 0100: System clock divided by 5 0101: System clock divided by 6 0110: System clock divided by 7 0111: System clock divided by 8 1000：PLLH/Q 1001：PLLH/R 1101：XTAL Other: prohibitions **Note:** 1. When the switching target clock source is PLLH, ensure that the PLLH clock is in a stable state. 2. When the system clock selects PLLH, it is necessary to turn off the clocks of each peripheral module, and then write the CMU_SCFGR register to switch the clock division. After writing the CMU_CANCKCFGR register, the software waits for the system to stabilize for 30us. 3. When the PWC_STPMCR.CKSMRC bit is 1, after waking up from Stop Mode, this register is initialized and MCANCLK is the XTAL clock source.	RW

4.11.23 CMU ADC/DAC Clock Configuration Register (CMU _ PERICKSEL)

Reset value: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res												PERICKSEL[3:0]			

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b15~b4	Res	Reserved Bit	Read at “0”, write at “0”	RW
b3~b0	PERICKSEL[3:0]	ADC/DAC clock source selection	0000: PCLK2/PCLK4 set by CMU_SCFGR 1000: PCLK2/ PCLK4 configured as PLLHQ 1001: PCLK2/ PCLK4 configured as PLLHR Other: prohibitions.	RW

Note:

– *When the switching target clock source is PLLH, ensure that the PLLH clock is in a stable state.*

4.11.24 CMU Debug Clock Config Register (CMU_TPIUCKCFGR)

Reset value: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
TPIUCKOE	Res					TPIUCKS[1:0]	

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b7	TPIUCKOE	TPIU clock supply enable bit	0: Prohibited 1: Permissible	RW
b6~b2	Res	Reserved Bit	Read at “0”, write at “0”	RW
b1~b0	TPIUCKS[1:0]	TPIU clock frequency division selection bit	00: 1 frequency division 01: 2 frequency division 10: 4 frequency division Other: prohibitions	RW

4.11.25 CMU MCO1 Clock Output Configuration Register (CMU_MCO1CFGR)

Reset value: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
MCO1EN	MCO1DIV[2:0]			MCO1SEL[3:0]			

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b7	MCO1EN	MCO _ 1 Output License	0: Disable MCO _ 1 Output 1: Allow MCO _ 1 Output	RW
b6~b4	MCO1DIV[2:0]	MCO _ 1 Frequency Division Selection	000: 1 frequency division 001: 2 frequency division 010: 4 frequency division 011: 8 frequency division 100:16 frequency division 101: 32 frequency division 110: 64 frequency division 111: 128 frequency division	RW
b3~b0	MCO1SEL[3:0]	MCO _ 1timer selection	0000: HRC clock 0001: MRC clock 0010: LRC clock 0011: XTAL clock 0100: XTAL32 clock 0110：PLLHP 1000：PLLHQ 1011: System clock Other: prohibitions.	RW

4.11.26 CMU MCO2 Clock Output Configuration Register (CMU_MCO2CFGR)

Reset value: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
MCO2EN	MCO2DIV[2:0]			MCO2SEL[3:0]			

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b7	MCO2EN	MCO_2 export permission	0: disable MCO_2 output 1: Enable MCO_2 output	RW
b6~b4	MCO2DIV[2:0]	MCO_2 frequency division selection	000: 1 frequency division 001: 2 frequency division 010: 4 frequency division 011: 8 frequency division 100:16 frequency division 101: 32 frequency division 110: 64 frequency division 111: 128 frequency division	RW
b3~b0	MCO2SEL[3:0]	MCO_2 clock source selection	0000: HRC clock 0001: MRC clock 0010: LRC clock 0011: XTAL clock 0100: XTAL32 clock 0110：PLLHP 1000：PLLHQ 1011: System clock Other: prohibitions.	RW

4.11.27 FCM Lower Limit Compare Value Register (FCM_LVR)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
LVR[15:0]															

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b16	Res	Reserved Bit	Read at “0”, write at “0”	RW
b15~b0	LVR[15:0]	lower limit comparison value	A 0 in the START bit configures this register.	RW

4.11.28 FCM Upper Limit Comparison Value Register (FCM_UVR)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
UVR[15:0]															

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b16	Res	Reserved Bit	Read at “0”, write at “0”	RW
b15~b0	UVR[15:0]	Upper limit comparison value	A 0 in the START bit configures this register.	RW

4.11.29 FCM Counter Value Register (FCM_CNTR)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
CNTR[15:0]															

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b16	Res	Reserved Bit	Read at “0”, write at “0”	RW
b15~b0	CNTR[15:0]	Counter value	Save the counter value to this register when a valid edge selected by the EDGES bit of the base clock is detected (except for the first valid edge after START=1)	R

4.11.30 FCM Start Stop Register (FCM_STR)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res															START

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b1	Res	Reserved Bit	Read at “0”, write at “0”	RW
b0	START	Frequency measurement start bit	0: Stop frequency measurement 1: Frequency measurement starts	RW

4.11.31 FCM Measurement Object Control Register (FCM_MCCR)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res								MCKS[3:0]				Res		MDIVS[1:0]	

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b8	Res	Reserved Bit	Read at “0”, write at “0”	RW
b7~b4	MCKS[3:0]	Measurement object clock selection bit	0000：XTAL 0001：XTAL32 0010：HRC 0011：LRC 0100：SWDTLRC 0101：PCLK1 0111：MRC 1000：PLLHP Other: prohibitions	RW
b3~b2	Res	Reserved Bit	Read at “0”, write at “0”	RW
b1~b0	MDIVS[1:0]	Measurement object frequency division selection	00: no frequency division 01：4 frequency division 10: 8 frequency division 11: 32 frequency division	RW

4.11.32 FCM Measurement Reference Control Register (FCM_RCCR)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
EXREFE	Res	EDGES[1:0]		Res		DNFS[1:0]		INES	RCKS[3:0]				Res	RDIVS[1:0]	

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b16	Res	Reserved Bit	Read at “0”, write at “0”	RW
b15	EXREFE	External pin input reference clock FCMREF enable bit	0: Disable external pin input reference clock FCMREF 1: Allow external pins to input reference clock FCMREF	RW
b14	Res	Reserved Bit	Read at “0”, write at “0”	RW
b13~b12	EDGES[1:0]	Measurement reference edge selection bit	00: Rising edge 01: Falling edge 10: rising and falling edge 11: Setting prohibited	RW
b11~b10	Res	Reserved Bit	Read at “0”, write at “0”	RW
b9~b8	DNFS[1:0]	Digital filter function selection bit	00: no filter function 01: The clock selected by the MCKS bit is used as the filter clock 10: The frequency of the clock selected by the MCKS bit is divided by 4 as the filter clock 11: 16 frequency division of the clock selected by the MCKS bit as the filter clock	RW
b7	INEXS	Measurement reference, internal clock and terminal selection bits	0: External pin input reference clock FCMREF 1: The clock selected by the RCKS selection bit	RW
b6~b3	RCKS[3:0]	Measurement reference clock selection bit	0000：XTAL 0001：XTAL32 0010：HRC 0011：LRC 0100：SWDTLRC 0101：PCLK1 0111：MRC 1000：PLLHP Other: prohibitions	RW
b2	Res	Reserved Bit	Read at “0”, write at “0”	RW
b1~b0	RDIVS[1:0]	Measurement reference frequency division selection	00: 32 frequency division 01: 128 frequency division 10: 1024 frequency division 11：8192 frequency division	RW

4.11.33 FCM Interrupt Reset Control Register (FCM_RIER)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res								ERRE	Res		ERRINTRS	Res	OVFE	MENDIE	ERRIE

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b8	Res	Reserved Bit	Read at “0”, write at “0”	RW
b7	ERRE	Frequency abnormal reset enable bit	0: Prohibited 1: Permissible	RW
b6~b5	Res	Reserved Bit	Read at “0”, write at “0”	RW
b4	ERRINTRS	Frequency exception interrupt reset selection bit	0: Frequency abnormal interrupt 1: Reset when frequency abnormality occurs	RW
b3	Res	Reserved Bit	Read at “0”, write at “0”	RW
b2	OVFIE	Counter overflow interrupt enable bit	0: Disable counter overflow interrupt 1: Enable counter overflow interrupt	RW
b1	MENDIE	End of measurement interrupt enable bit	0: disable interrupt at the end of measurement 1: Enable interrupt at end of measurement	RW
b0	ERRIE	Frequency exception interrupt enable bit	0: Disable frequency exception interrupt 1: Allow frequency exception to interrupt	RW

4.11.34 FCM Flags Register (FCM_SR)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res													OVF	MENDF	ERRF

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b8	Res	Reserved Bit	Read at “0”, write at “0”	RW
b2	OVF	Counter overflow flag	0: The counter has not overflowed 1: Counter overflow	R
b1	MENDF	Measurement end flag	0: measuring 1: End of measurement	R
b0	ERRF	Frequency abnormal flag bit	0: No frequency exception occurs 1: Abnormal occurrence frequency	R

4.11.35 FCM Flag Clear Register (FCM_CLR)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res													OVFCLR	MENDFCLR	ERRFCLR

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b3	Res	Reserved Bit	Read at “0”, write at “0”	RW
b2	OVFCLR	Counter overflow flag clear bit	Write 1 to clear the counter overflow flag	W
b1	MENDFCLR	Measurement end flag clear bit	Write 1 to clear the measurement end flag	W
b0	ERRFCLR	Frequency exception flag clear bit	Write 1 to clear the measurement end flag	W

5 Power Control (PWC)

5.1 Overview

Power controller is used to control the power supply, switching and detection of the chip's multiple power domains in multiple operation modes and low power modes. The power controller is composed of low power control logic (PWCL) and power voltage detection unit (PVD).

The operating voltage (VCC) of the chip is 1.8 V to 3.63 V. A voltage regulator (LDO) supplies power to the VDD and VDDR domains, and a VDDR voltage regulator (RLDO) supplies power the VDDR domain in power-down mode. The chip provides three low Power Consumption modes: sleep, stop and power-down through the Power Consumption control logic (PWC).

The power voltage detection unit (PVD) provides functions such as power-on reset (POR), power-down reset (PDR), brown-out reset (BOR), programmable voltage detection 1 (PVD1), and programmable voltage detection 2 (PVD2). Among them, POR, PDR, and BOR control the reset action of the chip by detecting the VCC voltage. PVD1 detects the VCC voltage and resets or interrupts the chip according to the register settings. PVD2 detects VCC voltage or external input detection voltage, and generates reset or interrupt according to register selection.

After the chip enters power-down mode, the VDDR area can maintain power through RLDO to ensure that the real-time clock module (RTC) and wake-up timer (WKTM) can continue to operate, and keep the data of 4KB RetSRAM.

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

5.2 Distribution of Power

Figure 5-1 is the power distribution diagram of the chip. The chip is composed of VCC domain, VDD power domain, VDDR power domain and AVCC power.

The VCC domain is powered by the VCC/VSS pin and consists of low-power control logic (PWCL), power supply voltage detection unit (PVD), IO level holding circuit, voltage regulator (LDO), VDDR domain voltage regulator (RLDO), external high-speed oscillator (XTAL), internal low-speed oscillator (LRC), external low-speed oscillator (XTAL32) and other circuits.

The VDD domain is composed of digital logic such as CPU, digital peripherals, RAM, FLASH, etc., and is powered by VDD generated by LDO. The RAM in the VDD domain is divided into two independent modules, and the power-off of each module can be independently controlled by registers.

The VDDR domain consists of 4KB of Ret SRAM, real-time clock (RTC), and wake-up timer (WKTM). Power is supplied by RLDO in power-down mode, and by LDO in modes other than power-down mode. In power-down mode, Ret-SRAM can keep data, real-time clock RTC and wake-up timer WKTM

can continue to operate. When the function of the VDDR domain is not needed, you can cut off the power supply of the VDDR domain in power-down mode by setting PWR_PWRC5.VVDRSD to further reduce Power Consumption.

The analog power domain is mainly composed of analog-to-digital converters (ADCs), digital-to-analog converters (DACs), comparators (CMPs), and analog system input and output pins, and is powered by the AVCC/AVSS pins.

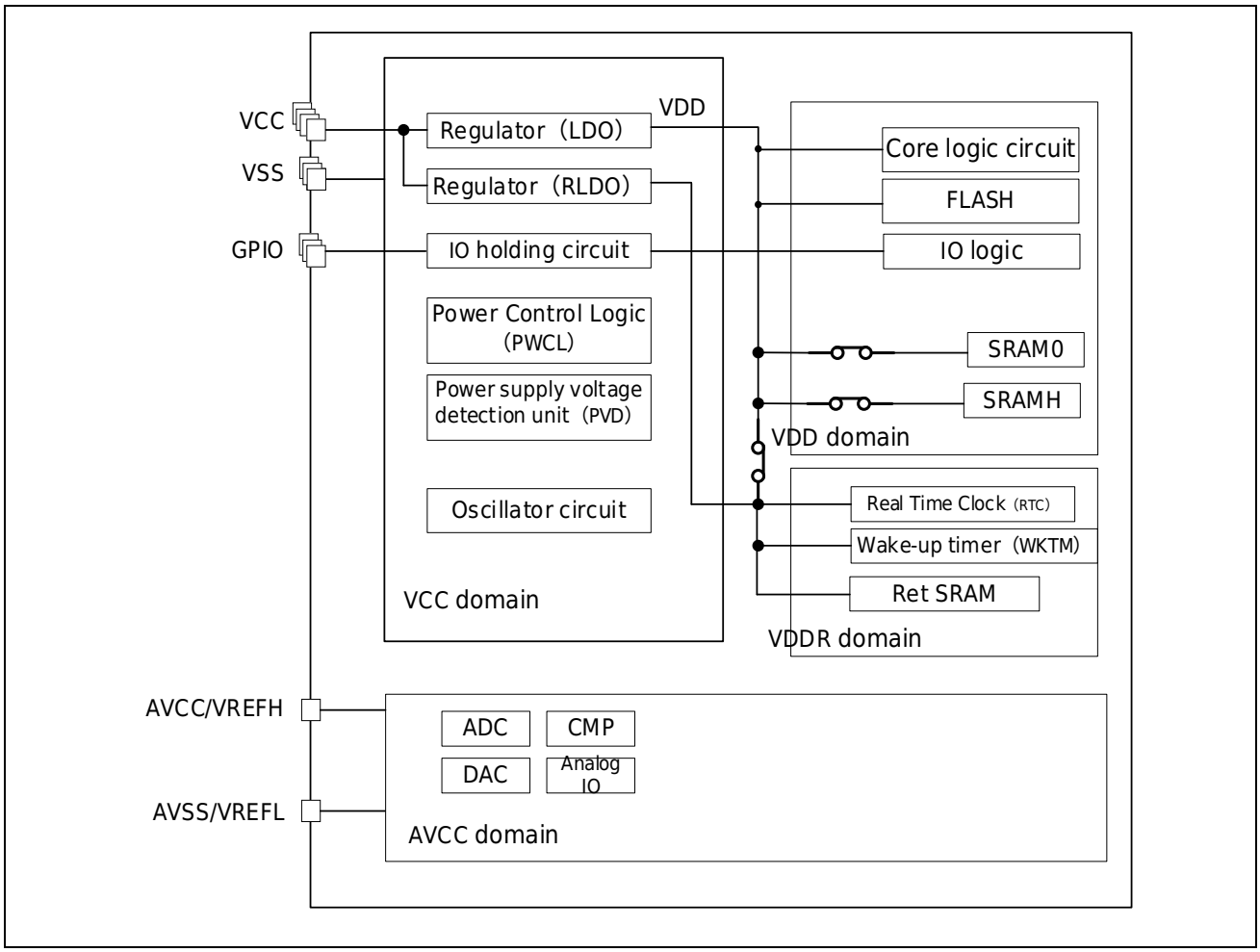


Figure 5-1 Power Supply Diagram

5.3 Power Voltage Detection Unit (PVD) Description

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

5.3.1 Description of Action of Power-on/Power-off Reset

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 Figure 5-2. 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 details of parameters such as V_{POR}, V_{PDR}, and T_{RSTPOR}, please refer to the **electrical characteristics in the Data Manual**.

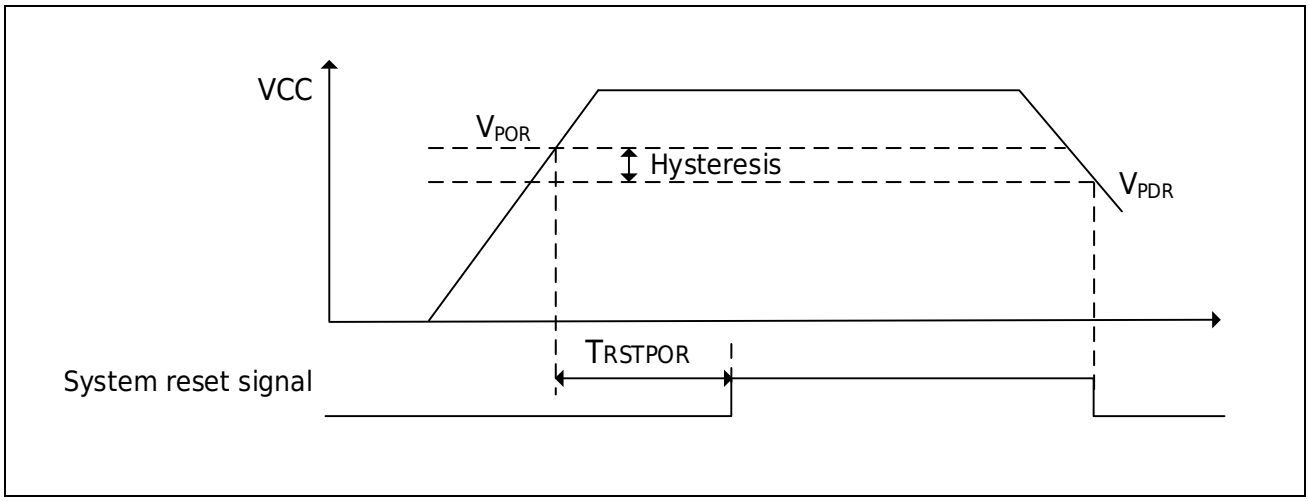


Figure 5-2 Power-on Reset, Power-off Reset Waveforms

5.3.2 Brown-out Reset (BOR) Description

During power-up, until VCC is higher than VBOR, a brown-out reset (BOR) will put the chip in reset.

The VBOR threshold is configured through the BOR_LEV, BORDIS of the initialization configuration bits (ICG). When BORDIS=0, the BOR detection voltage can be selected from 4 thresholds. When BORDIS is configured as 1, the chip performs reset control through power-on reset and power-off reset.

Table 5-1 BOR Configuration

BORDIS	BOR_LEV	Description
1	XX	BOR invalid
0	00	BOR Valid, Select BOR Threshold 0 (VBOR0)
0	01	BOR Valid, Select BOR Threshold 1 (VBOR1)
0	10	BOR Valid, Select BOR Threshold 2 (VBOR2)
0	11	BOR Valid, Select BOR Threshold 3 (VBOR3)

For the electrical features of the BOR threshold, please refer to the **Electrical Features Section of Data Sheet.**

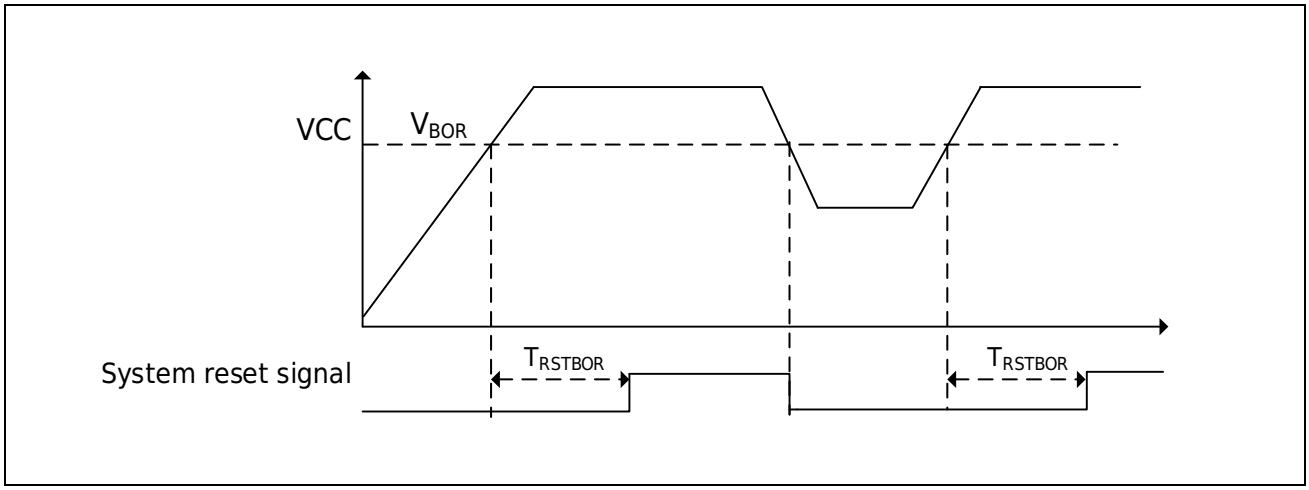


Figure 5-3 Brownout Reset Waveform

5.3.3 Programmable Voltage Detection 1 (PVD1), Programmable Voltage Detection 2 (PVD2)

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

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

The main features of the programmable voltage detection are shown Table 5-2 below.

Table 5-2 PVD1/PVD2 Features

Item	PVD1	PVD2
Test object	Whether VCC has passed the threshold voltage point (VPVD1) during the falling/rising process	Whether VCC passes through the threshold voltage point (VPVD2) during the decline/rise process, and whether the external input voltage rises/falls through the threshold voltage point (VPVD2, PWR_PVDLCR.PVD2LVL[2:0]=0b111)
Detection voltage point	Configured by PVD1LVL[2:0]	Configured by PVD2LVL[2:0]
Reduction	Reset: VCC<VPVD1; Reset release: VCC>VPVD1 after a certain reset processing time.	Reset: VCC<VPVD2; Reset release: VCC>VPVD2 after a certain reset processing time.
Interrupt	Configured as voltage detection 1 interrupt or non-maskable interrupt	Configured as voltage detection 2 interrupt or non-maskable interrupt
	VCC drops past the threshold voltage point (VPVD1) Or the VCC rising past the threshold voltage point (VPVD1) Or the VCC rising/falling past the threshold voltage point (VPVD1)	VCC drops past the threshold voltage point (VPVD2) Or the VCC rising past the threshold voltage point (VPVD2) Or the VCC rising/falling past the threshold voltage point (VPVD2)
Filtering function	Digital filtering	Digital filtering
AOS trigger function	VCC drops past the threshold voltage point (VPVD1)	VCC drops past the threshold voltage point (VPVD2)

5.3.4 PVD1, PVD2 Interrupt/Reset Block Diagram

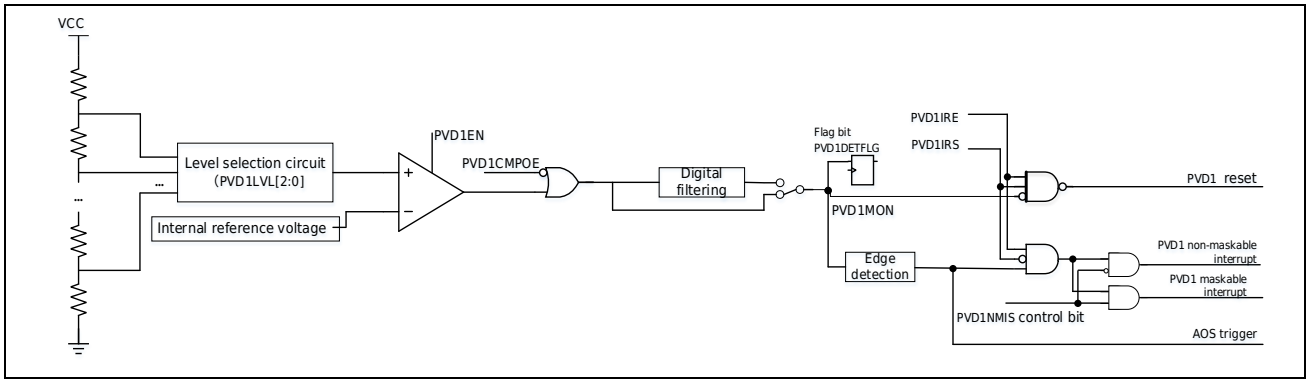


Figure 5-4 PVD1 Interrupt/Reset Block Diagram

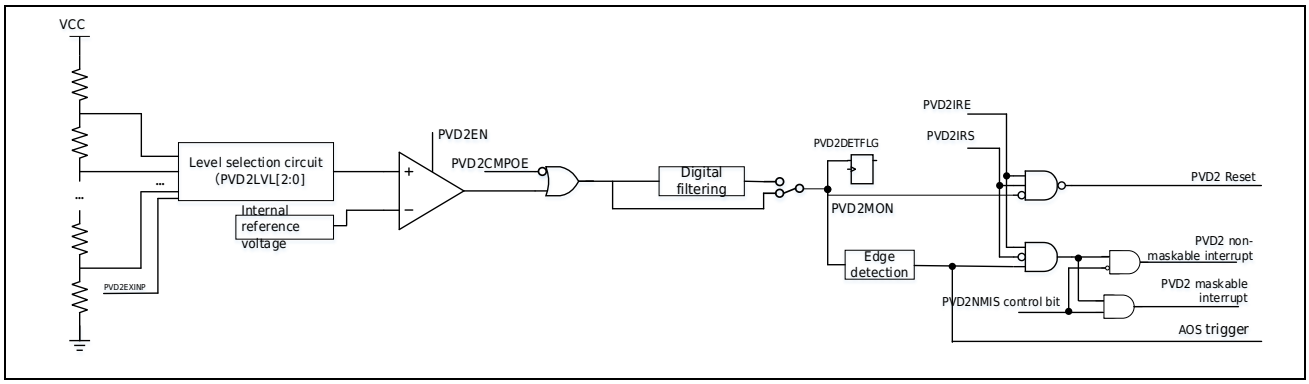


Figure 5-5 PVD2 Interrupt/Reset Block Diagram

5.3.5 Input/Output Pin

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

5.3.6 PVD1 Interrupt and Reset

When using the PVD1 circuit in Stop Mode or power-down mode, please observe the following precautions.

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

Figure 5-6 is the operating timing diagram of the voltage detection 1 interrupt. PVD1DETFLG needs to be cleared before the interrupt can occur again.

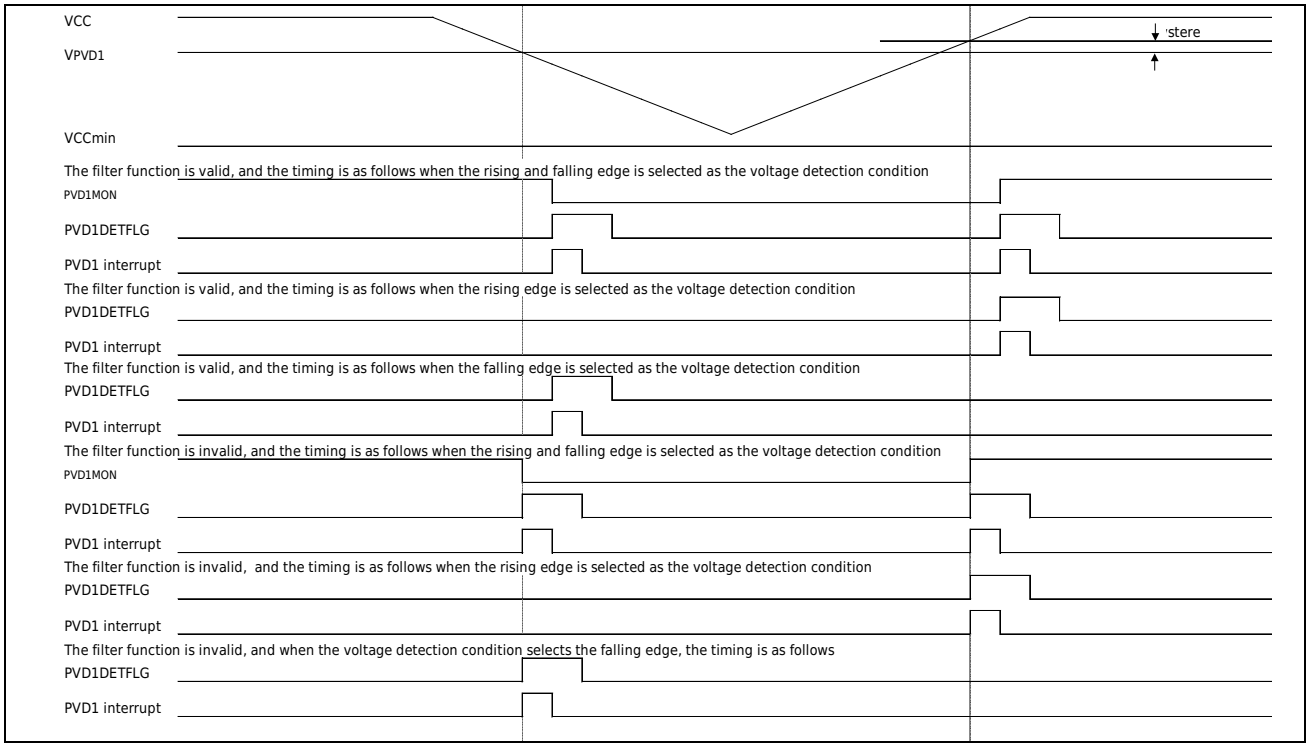


Figure 5-6 Voltage Detection 1 Interrupt Timing Diagram

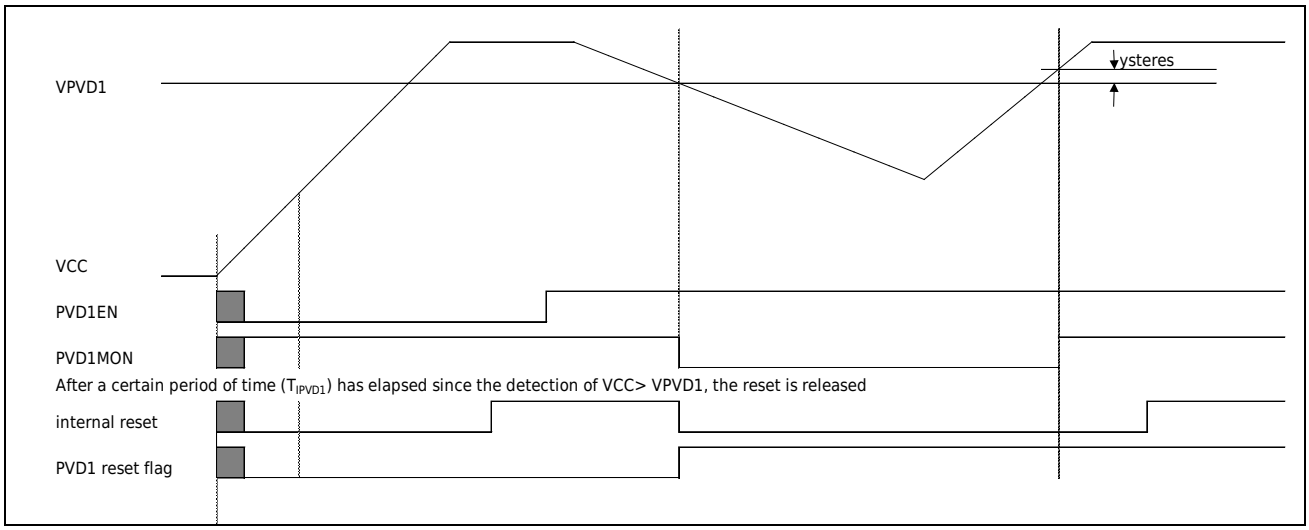


Figure 5-7 Voltage Detection 1 Reset Timing Diagram

5.3.7 PVD2 Interrupt and Reset

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

1. Stop Mode
   1) The digital filter must be disabled.
2. Power Down Mode
   1) The digital filter must be disabled.

2) PVD2INTRS is set to 0, and PVD2 is selected to generate an interrupt; when the reset function is selected, power-down mode cannot be entered.

Figure 5-8 is the operating timing diagram of the voltage detection 2 interrupt. PVD2DETFLG needs to be cleared before the interrupt can occur again.

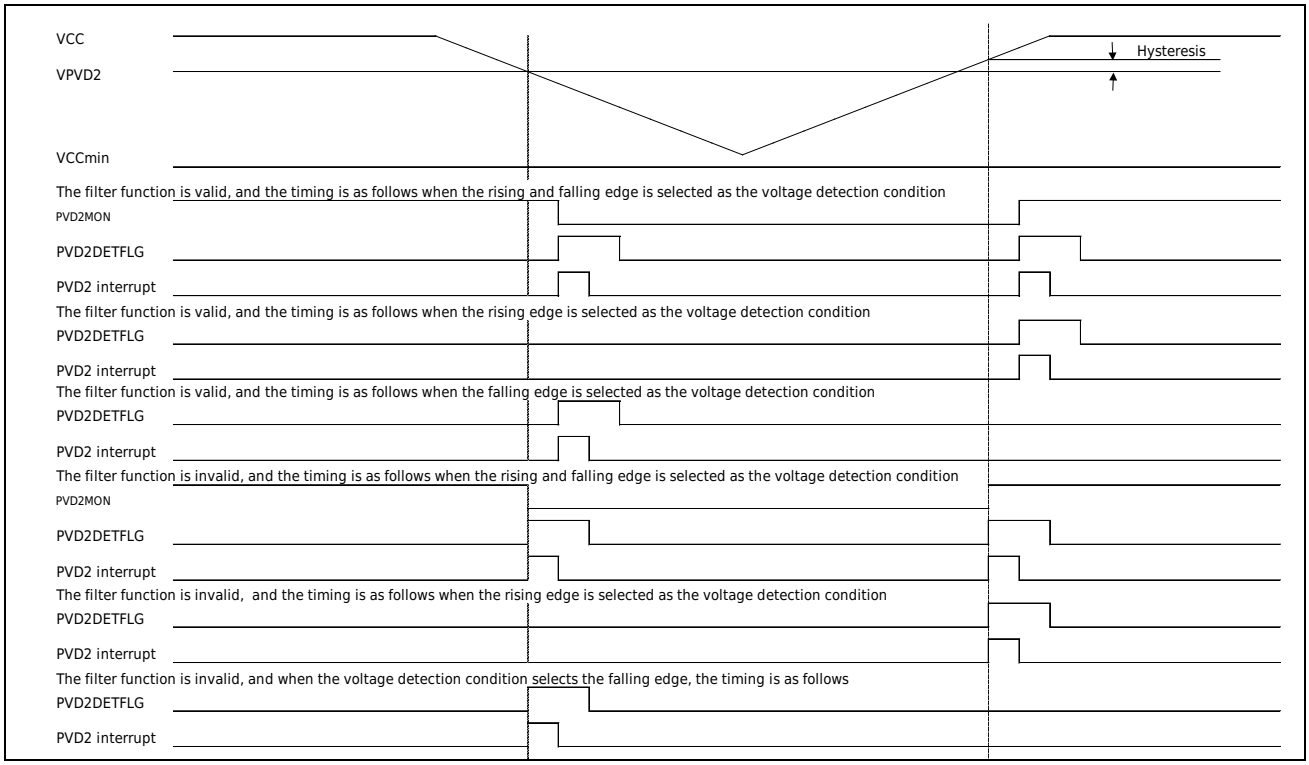


Figure 5-8 Voltage Detection 2 Interrupt Operation Sequence Diagram

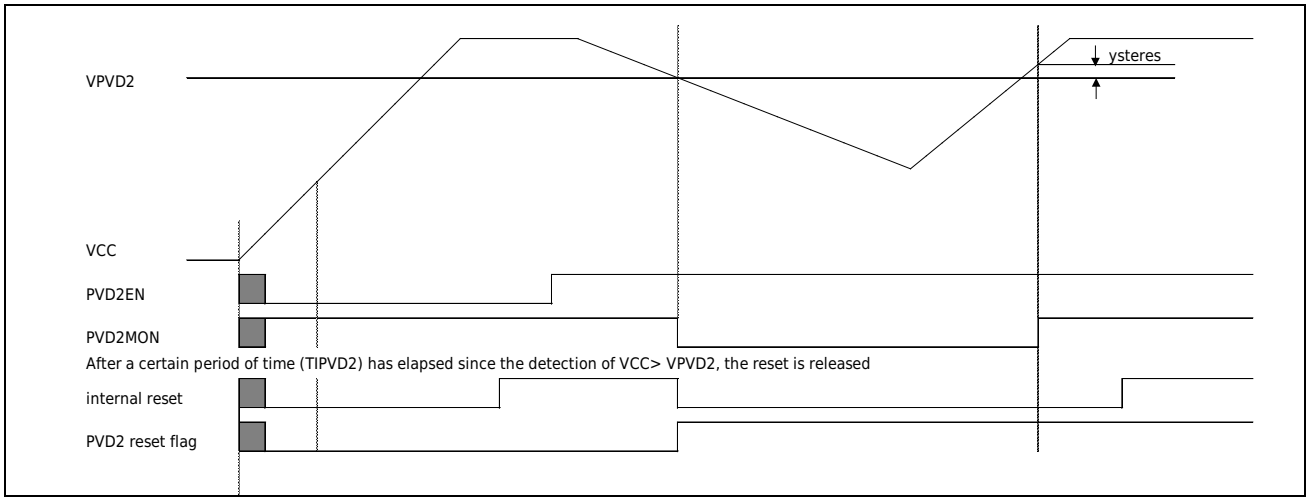


Figure 5-9 Voltage Detection 2 Reset Operation Sequence Diagram

5.3.8 Internal Voltage Sampling and Detection Function

The internal voltage sampling and detection functions of the chip include two functions: reference voltage measurement and AVCC voltage measurement. The reference voltage measurement path

is a function for measuring the reference voltage using ADC. The internal reference voltage is approximately 1.13V.

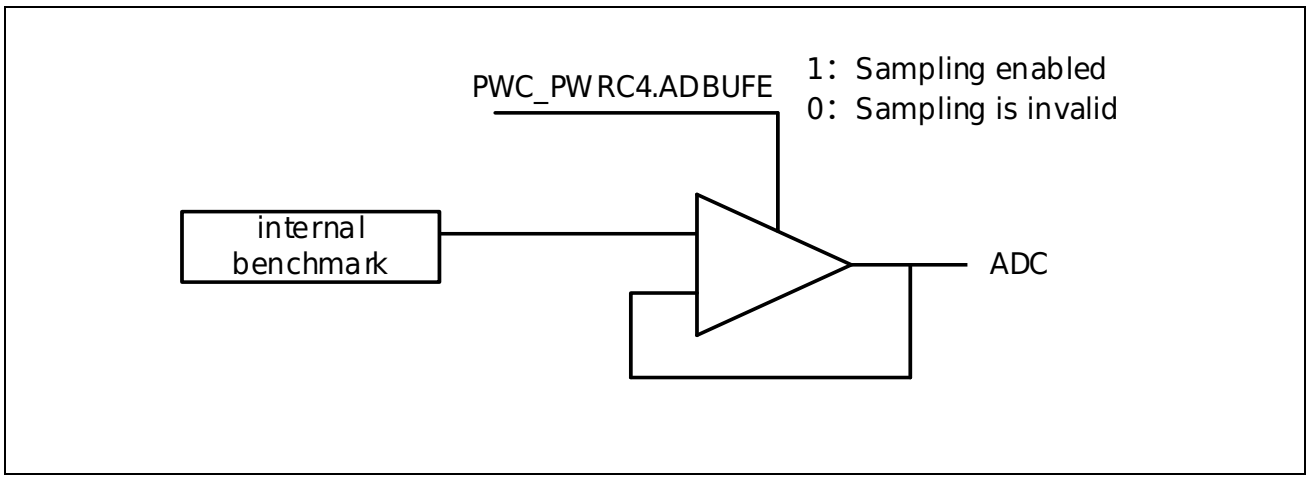


Figure 5-10 Internal Voltage Sampling Diagram

■ Reference Voltage Measurement Path

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

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

5.4 Wake-up timer

The chip has a built-in wake-up timer WKTM, which can select the internal low-speed oscillator (LRC) or external low-speed oscillator(XTAL32) as the clock source. When the RTC uses an external low-speed oscillator as the clock action, WKTM can also select the 64Hz internal clock signal as the clock source. This counter is an accumulative counter. After PWC_WKTCR.WKTCE is set, the counter starts counting. When the count value is equal to one cycle of the value set by PWC_WKTCR.WKTCMP[11:0], a match consistency event is generated. This event can be used as a normal interrupt and can be used to wake up the Stop Mode and power-down mode.

The value of the WKTM register is uncertain after power-on reset. After power-on reset, set PWC_WKTCR. Before starting WKTM, make sure the counting clock is turned on.

5.5 Action Mode and Low Power Mode

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 running mode. In run mode, the CPU provides the clock through HCLK and executes program code. The operating modes in which the chip can be configured are as Table 5-3 shown.

In order to save power consumption when the CPU does not need to run, the system provides three low-power modes: Sleep Mode, Stop Mode, and power-down mode. The low Power Consumption modes that the chip can be configured are as Table 5-4 shown. In Sleep Mode, the Cortex-M4 core of the chip stops operating, and the peripherals keep running; in Stop Mode, the peripherals and CPU of the chip stop operating; in power-down mode, the power supply of VDD domain is turned off, and the VDD domain Peripherals stop operating. The real-time clock and wake-up timer in the VDDR domain can operate in low-power mode, and the Ret SRAM can keep data; when the real-time clock, wake-up timer and Ret SRAM in the VDDR domain are not used, the VDDR domain can be set to be closed The voltage regulator RLDO can further reduce Power Consumption after entering power-down mode.

Users can choose 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.Users can 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.

Users can optimize Power Consumption during operation by the following methods.

1) Reduce system clock speed
2) Turn off unused clock source
3) Set the function clock control register PWC_FCGn (n=0~3) to turn off unused functions
4) Power off the RAM

The operating conditions of the low-power mode and the status of each module in the low-power mode are as Table 5-5 shown.

Table 5-3 Operation Mode

Operating Mode	Description
High Speed Run Mode	Main frequency 150MHz and below

Table 5-4 Low Power Modes

Mode		Description
Sleep Mode		CPU clock stopped, peripherals keep running
Stop Mode		The clocks of the chip peripherals and the CPU are stopped
Power Down Mode (Power Down Mode)	Power down mode 1 (PDMD1)	VDD domain power down
	Power down mode 2 (PDMD2)	The voltage detection unit is invalid except the VDD domain is powered off
	Power down mode 3 (PDMD3)	In addition to VDD domain power-off, VDDR domain power-off, power-on reset circuit enters low Power Consumption mode, and voltage detection unit (PVD) is invalid. Compared with power-down mode 4, other chips except PWC_PDWKF0/PWC_PDWKF1/RMU_RSTF0 when power-down wake up Complete reset.
	Power down mode 4 (PDMD4)	In addition to the VDD domain power-off, the VDDR domain power-off, the power-on reset circuit enters the low Power Consumption

Mode		Description
		mode, and the voltage detection unit (PVD) is invalid

Table 5-5 Operating Conditions of Low Power Consumption Mode and Status of Each Module in Low Power Consumption Mode

Item	Sleep Mode	Stop Mode	Power Down Mode
Entry	PWC_STPMCR.STOP=0, PWC_PWRC0.PWDN=0, WFI	PWC_STPMCR.STOP=1, PWC_PWRC0.PWDN=0, WFI	PWC_STPMCR.STOP=1, PWC_PWRC0.PWDN=1, WFI
Release	To interrupt or reset arbitrarily.	Interrupt or reset that can be used in Stop Mode	Wake-up event or reset available in power-down mode
External high-speed oscillator	Work can be done	Stop	Stop
External low-speed oscillator	Work can be done	Work can be done	Work can be done
Internal high-speed oscillator	Work can be done	Stop	power down
internal medium speed oscillator	Work can be done	Stop	power down
Internal low-speed oscillator	Work can be done	Work can be done	Work can be done
WDT dedicated clock oscillator	Work can be done	Work can be done	power down
PLLH	Work can be done	Stop	power down
CPU	To stop (hold).	To stop (hold).	power down
RAM (Except Ret SRAM)	Work can be done Can be set to work, power down	To stop (hold). According to the setting before entering standby, it can maintain power-off or sleep	power down
Flash	Work can be done	To stop (hold).	power down, content remains
DMA	Work can be done	To stop (hold).	power down
Voltage regulator	Work Drive adjustable	Work Drive adjustable	Stop
Power-on reset circuit	Work	Work	Work The accuracy of the reset circuit in power-down mode 1 and power-down mode 2 can be guaranteed, and the power-on reset voltage in power-down mode 3 and power-down mode 4 is not guaranteed
Brown-out Reset BOR	Work can be done	Work can be done	Power-down mode 1 work can be set Stop in power-down mode 2/3/4
Voltage detection module PVD	Work can be done	Work can be done	Power-down mode 1 work can be set Stop in power-down mode 2/3/4
WDT	Work can be done	To stop (hold).	power down
SWDT	Work can be done	Work can be done	power down
RTC	Work can be done	Work can be done	Power-down mode 1/2 work

Item	Sleep Mode	Stop Mode	Power Down Mode
			can be set Power down in power down mode 3/4
Timer0	Work can be done	Work can be done	power down
Ret SRAM	Work can be done Can be set to work, power down, sleep	Stop (Data hold) Can be set to power down, sleep	Power-down mode 1/2 stop (data retention), can be set to power-down, sleep Power down in power down mode 3/4
WKTM	Work can be done	Work can be done	Power-down mode 1/2 work can be set Power down in power down mode 3/4
Other peripheral modules	Work can be done	To stop (hold).	power down
ADC	Work can be done	Stop	power down
DAC	Work can be done	Work can be done	power down
CMP	Work can be done	Work can be done	power down
PA0~PA12, PB0~PB2, PB5~PB15, PC0~PC13, PD2, PF2	Work can be done	Hold on	Hold or high resistance
PC14~PC15	Work can be done	When used as an external low-speed oscillator pin, keep the oscillator active; when set as GPIO or other peripheral functions, please set to keep the two pins at the same level	When set to GPIO or other peripheral functions, the state of PC14 and PC15 can be set to hold or high impedance, please set to keep the two pins at the same level
PF0~PF1	Work can be done	When used as an external high-speed oscillator, oscillator stops concussion and the pin state keeps the state before entering Stop Mode. When GPIO or other peripheral functions are set, keep the status before STOP	When used as an external high-speed oscillator, oscillator stops concussion and the pin state keeps the state before entering power down mode. When GPIO or other peripheral functions are set, keep the status before power down mode
NRST reset pin	The chip is pulled out of the circuit to the VCC	Outside the chip is pulled up to VCC through resistance	Outside the chip is pulled up to VCC through resistance
PF3/MD	Work can be done	Maintain;	Maintain; The outside of the chip is connected to ground through a resistor
PA13~PA15，PB3，PB4	Work may be set up; The built-in pull-up circuit works as a JTAG function	Maintain; The built-in pull-up circuit works as a JTAG function	Maintain; The built-in pull-up circuit works as a JTAG function

5.5.1 Sleep Mode

In Sleep Mode, the CPU stops running and its internal register remains in Sleep Mode. The operating status of peripherals and other system modules other than the watchdog and special watchdog does not change.

When the ICG is set to start automatically, if the WDTSLPOFF bit of ICG0 is 1, the watchdog stops counting in Sleep Mode; if the WDTSLPOFF bit is 0, the watchdog continues counting in Sleep Mode. If the ICG is not set to start automatically, start the watchdog by software startup, then if the WDT_CR.SLPOFF bit is 1, the watchdog stops counting in Sleep Mode; if the WDT_CR.SLPOFF bit is 0, the watchdog is in Sleep Mode does not stop counting.

When the ICG is set to start automatically, if the ICG0 SWDTSLPOFF bit is 1, the special watchdog stops counting in Sleep Mode; if the SWDTSLPOFF bit is 0, the special watchdog continues counting in Sleep Mode.

- Enter Sleep Mode

  When PWC_STPMCR.STOP=0, WFI instructions can be executed to enter Sleep Mode.
- Quit 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 the reset exits the Sleep Mode.

5.5.2 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.

If the SWDTSLPOFF bit of ICG0 is set to 1, the special watchdog stops counting in Stop Mode. If SWDTSLPOFF bit is 0, the special watchdog continues counting in Stop Mode.

Before you execute the WFI command into Stop Mode, make sure that FLASH is not programmed or erased and that the oscillation stop delection function is invalid, otherwise the chip will enter Sleep Mode instead of Stop Mode.

Before executing the WFI instruction to enter Stop Mode, you must ensure that PWC_PWRC0.PDMDS[1:0] is configured to 0b00.

In Stop Mode, the ADC and DAC also consume power unless disabled before entering Stop Mode. To disable DAC, you need to clear DACR.DAE, DACR.DA1E and DACR.DA2E to 0. To disable ADC, clear the ADC_STR.STRT bit, set PWC_FCG3.ADC3, PWC_FCG3.ADC2, and PWC_FCG3.ADC1, and then execute the WFI instruction to enter Stop Mode.

When waking up from Stop Mode, through the CKSMRC bit in the PWC_STPMCR register. CKSMRC is used to control the timer after wakeup. When CKSMRC = 1, the timer of wakeup system is MRC. When CKSMRC = 0, the wake-up system clock remains the same as the timer before STOP.

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

Before executing the WFI instruction to enter the Stop Mode, you need to ensure that the division ratio of HCLK to PCLK1/2/3/4 is controlled at 1:1, 2:1 or 4:1. If the HCLK to any peripheral clock division ratio exceeds 4:1, you need to configure PCLK1=PCLK3 and PCLK1 and PCLK3 use their maximum clock division (CMU_SCFGR.PCLK1S=0b110, CMU_SCFGR.PCLK3S=0b110).

The leakage current in Stop Mode is different at different voltage temperatures, and the drive capability must meet the leakage demand of the chip.

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

When the Stop Mode is related through a non-mail interrupt, the corresponding bit of int_nmier need to enable the interrupt; when the state mode is Released Throud a Maskable Interrupt, The Corresponding bit of the int_wupenr register needed to be set to enable the wake-up permission of the interrupt.

- Enter Stop Mode

1. Clear PWC_LPMCSR.LPMF;
2. When PWC_STPMCR.STOP=1, PWC_PWRC0.PWDN=0, execute the WFI command to enter the Stop Mode;
3. 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.

Table 5-5 shows the status of the chip's peripherals and clock sources in stop mode.

- UnStop Mode

  The Stop Mode can be removed by resetting and terminating. The reset methods that can be used to release Stop Mode include pin reset, power-on reset, brown-out reset (BOR), programmable voltage delection 1/2 reset, and special watchdog reset. The interrupt events that can be used to unblock the Stop Mode are as follows:

Pin interrupt EIRQ0-15, voltage detection 1 interrupt, Voltage delecting 2 interrupt, special watchdog underflow interrupt, real-time clock cycle interrupt, alarm clock interrupt, Wake-up timer interrupt, comparator interrupt, USART1 RX interrupt, Timer0 compare match interrupt.

When the chip interrupts the Stop Mode, it starts the used timer before entering the Stop Mode. After all of the timer stabilizes, the chip unstops the mode.

When the WFI instruction is executed to enter the Stop Mode, an interrupt event that is not used to release the Stop Mode is generated. The chip will respond to the interrupt first instead of entering the Stop Mode. If you need to enter the Stop Mode first without executing this interrupt, you need to disable the interrupt that is not used to release the Stop Mode before executing the WFI instruction.

5.5.3 Power Down Mode

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

When the ICG is set to start automatically, if the SWDTSLPOFF bit of the ICG0 is 1, the special watchdog will be the same as other modules in the VDD domain, the power will be cut off and no longer count. If the SWDTSLPOFF bit of the ICG0 is 0, the chip will enter Stop Mode instead of power-down mode, and the special watchdog oscillator and special watchdog will continue to run.

When the reset of Voltage Detect 1 and Voltage Detect 2 is enabled, the chip will enter Stop Mode instead of power-down mode.

Before executing the WFI instruction to enter the power-down mode, you need to ensure that the FLASH is not in the programming or erasing state and the oscillation stop detection function is invalid, otherwise the chip will enter the Sleep Mode instead of the power-down mode.

By setting PWC_PWRC0.PDMDS[1:0], the Power Consumption in power-down mode can be further reduced. The sub-modes of power-down mode are Table 5-6 shown. In power-down mode 1, the voltage detection circuit can be used and the power-on reset detection circuit is in action. Since there is no need to wait for the VCC domain reference voltage, the voltage detection circuit and the power-on reset detection circuit to be stable when waking up, the wake-up time is minimized while achieving low power consumption. In power-down mode 2, the VCC domain reference voltage circuit and the voltage detection circuit stop working, and the power-on reset detection circuit is in an action state. When waking up, it is necessary to wait for the stabilization time of the VCC domain reference voltage circuit and the voltage detection circuit. In power-down mode 3, the VCC domain reference voltage circuit, voltage detection circuit, and power-on reset detection circuit all stop working. When waking up, it is necessary to wait for these circuits to stabilize. Therefore, while achieving the lowest power consumption, the wake-up time is longer than power-down mode 2 and power-down mode 1. Power-down mode 4 has the same circuit that stops working in power-down mode 3, so power-down mode 4 has the same power consumption as power-down mode 3. For specific power consumption values and wake-up time, please refer to the electrical characteristics section of the corresponding data sheet.

The VDDR domain can work in power-down mode 1 and power-down mode 2, so the real-time clock module, wake-up timer can continue to run, and can be used to wake up power-down mode. Ret SRAM can still retain data in power-down mode. If the real-time clock, wake-up timer, and Ret SRAM are not needed in power-down mode, you can set PWR_PWRC5.VVDRSD to turn off the low-power voltage regulator to further reduce power consumption.

Table 5-6 Power-down Mode Submodes

Power Down Mode	PDMDS[1:0]	Power Consumption	Wake-up Time	Description
Power down mode 1	0b00	I_{PD1}	T_{PD1}	VCC domain supply voltage detection unit is valid
Power down mode 2	0b01	I_{PD2}	T_{PD2}	VCC domain POR, PDR detection circuit is valid, BOR, PVD1, PVD2 are invalid
Power down mode 3	0b10	I_{PD3}	T_{PD3}	VCC domain POR, PDR, BOR, PVD1, PVD2 invalid
Power down mode 4	0b11	I_{PD4}	T_{PD4}	VCC domain POR, PDR, BOR, PVD1, PVD2 invalid

Relationship between Power Consumption and wake-up time: $I_{pd1}>I_{pd2}>I_{pd3}=I_{pd4}$, $T_{pd1}<T_{pd2}<T_{pd4}<T_{pd3}$

- enter power-down mode
  1. Clear PWC_LPMCSR.LPMF;
  2. When PWC_STPMCR.STOP=1，PWC_PWRC0.PWDN=1, execute the WFI command to enter the power-down mode.
  3. Read PWC_LPMCSR.LPMF. If this bit is 0, return to step 2. If a power-off reset occurs normally, step 3 will not be executed.

Note:

When selecting Power-Down Mode 1 (PWC_PWRC0.PDMDS=0b00), please set PWC_PWRC4.PD1PW before executing the WFI instruction.

- Exit power-down mode

  Power-down mode can be released by power-down mode wake-up event or reset. The resets that can be used to wake up the power-down mode include pin reset, power-on reset and brown-out reset. Events that can be used to wake up from power-down mode include:

WKUPn_0/1/2/3 (n=0~3) wake-up event, real-time clock alarm and timing event, voltage detection 1 wake-up event, voltage detection 2 wake-up event, wake-up timer wake-up event

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

In power-down mode 3, POR, PDR, BOR, PVD1, and PVD2 circuits are all in an invalid state. After waking up from power-down mode 3, all registers except PWC_PDWKF0/PWC_PDWKF1/RSTF0 are reset, and the chip follows a similar power-on reset mode; the reset flag bit can be queried through RMU_RSTF0.PDRF.

In power-down mode 4, POR, PDR, BOR, PVD1, and PVD2 circuits are all in an invalid state, and the program is re-executed after the chip is reset; the reset flag bit can be queried through RMU_RSTF0.PDRF.

The power-down wake-up event is controlled by the power-down wake-up enable register (PWC_PDWKE0-PDWKE3) and the power-down wake-up event edge selection register

(PWC_PDWKES). When a power-down wake-up event occurs, the corresponding power-down wake-up flag (PWC_PDWKF0-PWC_PDWKF1) is set. After power-down wake-up, if the power-down wake-up flag is not cleared, the chip cannot enter power-down mode again. The edge of power-down wake-up event can be selected by PWC_PDWKES.

When waking up from power-down mode, the VDD domain will re-supply power, the system executes power-down wake-up reset, and the internal medium-speed oscillator works as clock. When waking up from power-down mode Registers that are not reset the following table.

Power Down Mode	Registers that are not reset
Power down mode 1 Power down mode 2 Power down mode 4	PWC_PWRC0，PWC_PWRC1，PWC_PWRC3，PWC_PDWKE0， PWC_PDWKE1，PWC_PDWKE2 PWC_PDWKES，PWC_PDWKF0，PWC_PDWKF1，PWC_PVDCR0， PWC_PVDCR1，PWC_PVDFCR，PWC_PVDLCR，PWC_PVDICR，PWC_PVDDSR
Power down mode 3	PWC_PDWKF0，PWC_PDWKF1，RMU_RSTF0

- Pin status after exiting power-down mode

  In power-down mode, the chip pins will remain in the state before entering power-down mode or in high-impedance state according to the register setting value. If PWC_PWRC0.IORTN[1:0]=0b10 or 0b11, the pin state is high-impedance in power-down mode. After the power-down mode is released, the pin is initially in high-impedance state. If PWC_PWRC0.IORTN[1:0]= 0b00, the state of the pin remains in the state before the power-down mode in the power-down mode, and the pin is initially in a high-impedance state after waking up. If PWC_PWRC0.IORTN[1:0]= 0b01, the pins of the chip will remain in the state before entering the power-down mode, and the state of the pins of the chip will not change even if the registers of the peripherals or pins are set after waking up. After PWC_PWRC0. IORTN is cleared by software, the pin state is controlled by the peripheral or pin register settings.

- WKTM Power-down mode wake-up event

  The chip has a built-in counter WKTM for power-down wake-up. The counter can select an internal low-speed oscillator or an external low-speed oscillator. When the RTC uses an external low-speed oscillator as the clock action, WKTM can also select a 64Hz internal clock signal as the clock source. This counter is an accumulative counter. After WKTC0.WKTCE is set, the counter starts counting. When the count value is equal to one cycle of the value set by WKTCMP[11:0], the counting stops and a wake-up event is generated, waking up the chip from power-down mode. When using WKTM again, you need to reset WKTC0.WKTCE and set it again.

- PTWK Power-down Mode Wake-up Event

  The chip has four PTWK events for waking up from power-down mode: PTWK0, PTWK1, PTWK2, and PTWK3. By setting PWC_PDWKE0/PWC_PDWKE1 through software, one of the four pins WKUPn_0, WKUPn_1, WKUPn_2, and WKUPn_3 can be enabled to trigger a PTWK event. Each

PTWK event can select the rising or falling trigger edge of the trigger pin and has an independent flag bit.

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

1. Configure the corresponding bit of PWC_PDWKES to select the edge
2. Configure PWC_PDWKE0 and PWC_PDWKE1 to enable the corresponding pins
3. Clear the corresponding flag in PWC_PDWKF0

The structural block diagram of configuring PTWKn is shown in Figure 5-11.

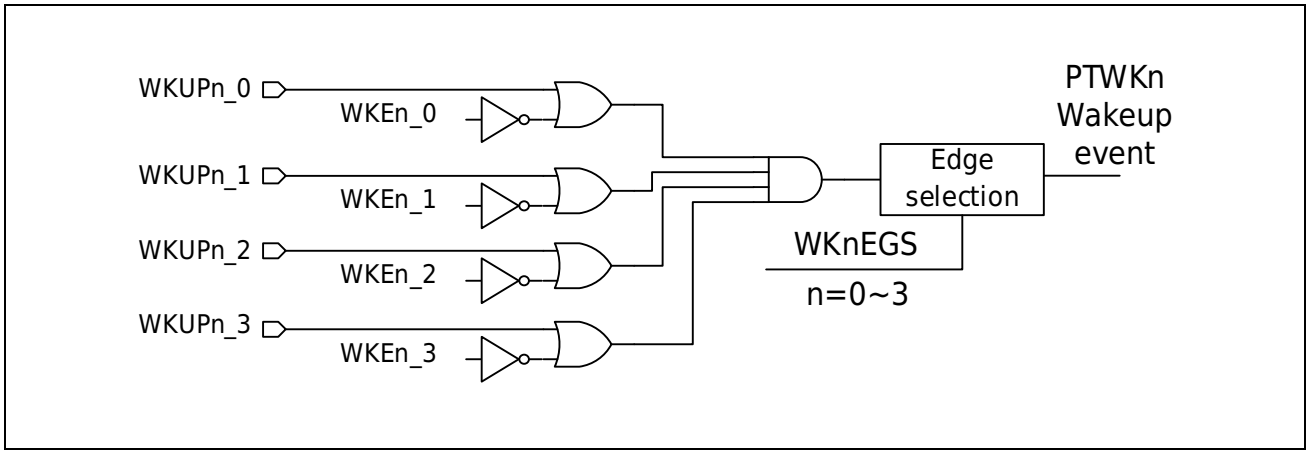


Figure 5-11 PTWKn Structure Block Diagram

5.5.4 Reduce System Clock Speed

The speed of system (HCLK) and peripheral clock PCLK0/PCLK1/PCLK2/PCLK3/PCLK4 can be reduced by programming the prescaler register. These pre-dividers can also be used to reduce the speed of peripherals before entering Sleep Mode. For details, please refer to [Clock Controller (CMU)].

5.5.5 Turn off unused clock source

The system clock of the chip has six timers:

- External high-speed oscillator (XTAL)
- External low-speed oscillator (XTAL32)
- PLLH Clock (PLLH)
- Internal high-speed oscillator (HRC)
- Internal Medium Speed Oscillator (MRC)
- Internal low-speed oscillator (LRC)

SWDT has an independent dedicated internal low-speed oscillator (SWDTLRC); RTC can choose an external low-speed oscillator or an internal low-speed oscillator as a clock source. Both HRC and PLL are equipped with independent power supply circuits. After HRC is turned off, the power supply of HRC can be turned off by setting the PWC_PWRC1.VHRCSD bit to further reduce Power Consumption. After PLLH is turned off, the power supply of PLLH can be turned off by setting PWC_PWRC1.VPLLSD=0b11.

For details, please refer to [Clock Management Unit (CMU)].

5.5.6 Stop Functional Clock

The peripheral module of the chip is provided with the function clock stop function. By using 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 will remain in the state before it stops.

5.5.7 Turn-off Unused RAM

Each RAM module in the chip is configured with a function clock stop bit and a power-down control bit. By setting the module stop bit, it stops providing clocks to RAMs that do not need to be used, thereby reducing Power Consumption. By setting the power-down control bit of the module, the corresponding RAM module can be powered down, thereby reducing Power Consumption. Table 5-7 is the correspondence between the RAM module and the power-off control bit. The corresponding RAM can be powered off by setting the corresponding RAMPDCn (n=0, 10) of the PWC_RAMPC0 register.

Table 5-7 RAM Module and RAM Power-down Control Bits

RAM Module	Description	Power Down Control Bit
SRAM0	System RAM , address range: 0x2000 0000~0x2000 3FFF	PWC_RAMPC0.RAMPDC0
SRAMH	System RAM , address range: 0x1FFF C000~0x1FFF FFFF	PWC_RAMPC0.RAMPDC10
CAN RAM	RAM for CAN	PWC_PRAMLPC.PRAMDC0
CACHERAM	RAM for Cache	PWC_PRAMLPC.PRAMDC2
Ret SRAM	Power-down mode retains RAM, address range: 0x200F 0000~0x200F 0FFF	PWR_PWRC5.SRAMBSD

5.6 Register Protection Function

Register protection is used to invalidate the write operation of the register to protect the register from accidental overwriting. Table 5-8 is a list of register protection bits and protected registers.

Table 5-8 Register Protection List

Protected Register Bit	Protected Register
PWC_FPRC.FPRCB0	CMU_XTALCFGR，CMU_XTALSTBCR，CMU_XTALCR，CMU_XTALSTDCR， CMU_XTALSTDSR，CMU_HRCTRM，CMU_HRCCR，CMU_MRCTRM， CMU_MRCCR，CMU_PLLHCFGR，CMU_PLLHCR， CMU_OSCSTBSR，CMU_CKSWR，CMU_SCFGR，CMU_CANCKCFGR， CMU_TPIUCKCFGR，CMU_MCO1CFGR，CMU_MCO2CFGR， CMU_XTAL32CR，CMU_XTALC32CFGR，CMU_XTAL32NFR， CMU_LRCCR，CMU_LRCTRM
PWC_FPRC.FPRCB1	PWC_PWRC0，PWC_PWRC1，PWC_PWRC2， PWC_PWRC3，PWC_PWRC4，PWC_PWRC5， PWC_PWRC6，PWC_PDWKE0，PWC_PDWKE1， PWC_PDWKE2，PWC_PDWKES，PWC_PDWKF0，PWC_PDWKF1，CMU_PERICKSEL， PWC_STPMCR，PWC_RAMPC0，PWC_PRAMLPC，RMU_RSTF0，PWC_WKTC， RMU_FRST0，RMU_FRST1，RMU_FRST2，RMU_FRST3，RMU_PRSTCR0
PWC_FPRC.FPRCB3	PWC_PVDCR0，PWC_PVDCR1，PWC_PVDFCR，PWC_PVDLCR，PWC_PVDICR，PWC_PVDDSR
PWC_FCG0PC.PRT0	PWC_FCG0

5.7 Register Description

Table 5-9 PWC Register List

Base address 1: 0x4004 CC00

Register	Description	Offset Address	Bit Width	Reset Value
PWC_PWRC0	Power control register 0	0x00	8	0x00
PWC_PWRC1	Power Mode Control Register 1	0x04	8	0xC0
PWC_PWRC4	Power Mode Control Register 4	0x10	8	0x00
PWC_PVDCR0	PVD Control Register 0	0x14	8	0x00
PWC_PVDCR1	PVD Control Register 1	0x18	8	0x00
PWC_PVDFCR	PVD filter control register	0x1C	8	0x11
PWC_PVDLCR	PVD level control register	0x20	8	0x00
PWC_PDWKE0	Power-down wake-up enable register 0	0x28	8	0x00
PWC_PDWKE1	Power-down wake-up enable register 1	0x2C	8	0x00
PWC_PDWKE2	Power-down wake-up enable register 2	0x30	8	0x00
PWC_PDWKES	Power-down wake-up event edge select register	0x34	8	0x00
PWC_PDWKF0	Power-down wake-up flag register 0	0x38	8	0x00
PWC_PDWKF1	Power-down wake-up flag register 1	0x3C	8	0x00
PWC_PWRC5	Power Mode Control Register 5	0x40	8	0x00
PWC_PWRC6	Power Mode Control Register 6	0x44	8	0x00
PWC_PVDICR	PVD Interrupt Control Register	0xC0	8	0x00
PWC_PVDDSR	PVD Detection Status Register	0xC4	8	0x11
PWC_RAMPC0	RAM Power Control Register 0	0xC8	32	0x0000 0000
PWC_PRAMLPC	Peripheral RAM Low Power Control Register	0xD0	32	0x0000 0000
PWC_LPMCSR	Low Power Status Register	0xD8	16	0x0000
PWC_LPMCSRC	Low Power State Clear Register	0xDA	16	0x0000

Base address 2: 0x4005 4000

Register	Description	Offset Address	Bit Width	Reset Value
PWC_STPMCR	Stop Mode Control Register	0x0C	16	0x0000
PWC_FPRC	Function protection control register	0x3FE	8	0x00

Base address 3: 0x4004 C400

Register	Description	Offset Address	Bit Width	Reset Value
PWC_WKTCR	Wake-up Timer Control Register	0x50	16	0x0000

Base address 4: 0x4004 8000

Register	Description	Offset Address	Bit Width	Reset Value
PWC_FCG0	Function Clock Control 0	0x00	32	0xFFFF FA0E
PWC_FCG1	Function Clock Control 1	0x04	32	0xFFFF FFFF
PWC_FCG2	Function Clock Control 2	0x08	32	0xFFFF FFFF
PWC_FCG3	Function Clock Control 3	0x0C	32	0xFFFF FFFF
PWC_FCG0PC	PWC_FCG0 protection control	0x10	32	0x0000 0000

5.7.1 Power Mode Control Register 0 (PWC_PWRC0)

Reset value: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
PWDN	Res	IORTN[1:0]		Res		PDMDS[1:0]	

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b7	PWDN	Power-down Mode Control Bit	0: Disable power-down mode 1: Enable power-down mode	RW
b6	Res	Reserved Bit	Read at “0”, write at “0”	RW
b5~b4	IORTN[1:0]	IO remains in control in power-down mode	00 : IO hold state in power-down mode , and the hardware releases IO hold state after power-down wake-up 01: IO hold status in power-down mode. After power-down wake-up, set IORTN[1:0] to 0b00 to release the IO hold status. 1x: IO is high impedance in power-down mode and after power-down wake-up	RW
b3~b2	Res	Reserved Bit	Read at “0”, write at “0”	RW
b1~b0	PDMDS[1:0]	Power-down mode selection control	00: power down mode 1 01: power down mode 2 10: power down mode 3 11: power down mode 4	RW

5.7.2 Power Mode Control Register 1 (PWC_PWRC1)

Reset value: 0xC0

b7	b6	b5	b4	b3	b2	b1	b0
Res					VHRCSD	VPLLSD[1:0]	

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b7~b3	Res	Reserved Bit	Read at “0b11000”, write at “0b11000”	RW
b2	VHRCSD	HRC Power-off	0: HRC power enable 1: HRC power off When HRC is not in use, set VHRCSD and turn off the power supply for HRC to further reduce Power Consumption. After VHRCSD is cleared, you need to wait 25μs before starting the HRC module	RW
b1~b0	VPLLSD[1:0]	PLL power off	00: PLL power enable 11: PLL power off Other: prohibitions After PLLH is turned off and waiting for 50μs, set VPLLSD = 0b11 and turn off the PLL power supply to further reduce Power Consumption. After setting VPLLSD=0b00, you need to wait 25μs before starting the PLLH module.	RW

5.7.3 Power Mode Control Register 4 (PWC_PWRC4)

Reset value: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
ADBUFE	Res	PD1PW	Res				

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b7	ADBUFE	Internal Voltage Sampling Enable	When using the internal voltage of the AD sampling chip, this bit needs to be set to 1 0: Invalid 1: Valid	RW
b6	Res	Reserved Bit	Read at “0”, write at “0”	RW
b5	PD1PW	Power-down mode 1 wake-up power stabilization control	Before entering Power-Down Mode 1, PD1PW must be set.	RW
b4~b0	Res	Reserved Bit	Read at “0”, write at “0”	RW

5.7.4 Power-Down Wake-up Enable Register 0 (PWC_PDWKE0)

Reset value: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
WKE13	WKE12	WKE11	WKE10	WKE03	WKE02	WKE01	WKE00

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b7	WKE13	WKUP1_3 wakeup event enable	0: WKUP1_3 wakeup event is invalid 1: WKUP1_3 wakeup event enable	RW
b6	WKE12	WKUP1_2 wakeup event enable	0: WKUP1_2 wakeup event is invalid 1: WKUP1_2 wakeup event enable	RW
b5	WKE11	WKUP1_1 wakeup event enable	0: WKUP1_1 wakeup event is invalid 1: WKUP1_1 wakeup event enable	RW
b4	WKE10	WKUP1_0 wakeup event enable	0: WKUP1_0 wakeup event is invalid 1: WKUP1_0 wakeup event enable	RW
b3	WKE03	WKUP0_3 wakeup event enable	0: WKUP0_3 wakeup event is invalid 1: WKUP0_3 wakeup event enable	RW
b2	WKE02	WKUP0_2 wakeup event enable	0: WKUP0_2 wakeup event is invalid 1: WKUP0_2 wakeup event enable	RW
b1	WKE01	WKUP0_1 wakeup event enable	0: WKUP0_1 wakeup event is invalid 1: WKUP0_1 wakeup event enable	RW
b0	WKE00	WKUP0_0 wakeup event enable	0: WKUP0_0 wakeup event is invalid 1: WKUP0_0 wakeup event enable	RW

5.7.5 Power-Down Wake-up Enable Register 1 (PWC_PDWKE1)

Reset value: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
WKE33	WKE32	WKE31	WKE30	WKE23	WKE22	WKE21	WKE20

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b7	WKE33	WKUP3_3 wakeup event enable	0: WKUP3_3 wakeup event is invalid 1: WKUP3_3 wakeup event enable	RW
b6	WKE32	WKUP3_2 wakeup event enable	0: WKUP3_2 wakeup event is invalid 1: WKUP3_2 wakeup event enable	RW
b5	WKE31	WKUP3_1 wakeup event enable	0: WKUP3_1 wakeup event is invalid 1: WKUP3_1 wakeup event enable	RW
b4	WKE30	WKUP3_0 wakeup event enable	0: WKUP3_0 wakeup event is invalid 1: WKUP3_0 wakeup event enable	RW
b3	WKE23	WKUP2_3 wakeup event enable	0: WKUP2_3 wakeup event is invalid 1: WKUP2_3 wakeup event enable	RW
b2	WKE22	WKUP2_2 wakeup event enable	0: WKUP2_2 wakeup event is invalid 1: WKUP2_2 wakeup event enable	RW
b1	WKE21	WKUP2_1 wakeup event enable	0: WKUP2_1 wakeup event is invalid 1: WKUP2_1 wakeup event enable	RW
b0	WKE20	WKUP2_0 wakeup event enable	0: WKUP2_0 wakeup event is invalid 1: WKUP2_0 wakeup event enable	RW

5.7.6 Power-Down Wake-up Enable Register 2 (PWC_PDWKE2)

Reset value: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
WKTMWKE	Res	RTCALMWKE	RTCPRDWKE	Res		VD2WKE	VD1WKE

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b7	WKTMWKE	WKTM wakeup event enable	0: WKTM wakeup event is invalid 1: WKTM wake-up event enable	RW
b6	Res	Reserved Bit	Read at “0”, write at “0”	RW
b5	RTCALMWKE	RTC alarm clock wake-up event enable	0: RTC alarm clock wake-up event is invalid 1: RTC alarm clock wake-up event enable	RW
b4	RTCPRDWKE	RTC cycle wakeup event enable	0: RTC cycle wake-up event is invalid 1: RTC cycle wakeup event enable	RW
b3~b2	Res	Reserved Bit	Read at “0”, write at “0”	RW
b1	VD2WKE	PVD2 wakeup event enable	0: PVD2 wakeup event is invalid 1: PVD2 wakeup event enable	RW
b0	VD1WKE	PVD1 wakeup event enable	0: PVD1 wakeup event is invalid 1: PVD1 wakeup event enable	RW

5.7.7 Power-down Wake-up Event Edge Selection Register (PWC_PDWKES)

Reset value: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
Res		VD2EGS	VD1EGS	WK3EGS	WK2EGS	WK1EGS	WK0EGS

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b7~b6	Res	Reserved Bit	Read at “0”, write at “0”	RW
b5	VD2EGS	VD2 edge selection	0：VCC < VPVD2 1：VCC > VPVD2	RW
b4	VD1EGS	VD1 edge selection	0：VCC < VPVD1 1：VCC > VPVD1	RW
b3	WK3EGS	PTWK3 edge selection	0: Falling edge 1: Rising edge	RW
b2	WK2EGS	PTWK2 edge selection	0: Falling edge 1: Rising edge	RW
b1	WK1EGS	PTWK1 edge select	0: Falling edge 1: Rising edge	RW
b0	WK0EGS	PTWK0 edge selection	0: Falling edge 1: Rising edge	RW

5.7.8 Power-down Wake-up Flag Register 0 (PWC_PDWKF0)

Reset value: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
Res		VD2WKF	VD1WKF	PTWK3F	PTWK2F	PTWK1F	PTWK0F

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b7~b6	Res	Reserved Bit	Read at “0”, write at “0”	RW
b5	VD2WKF	PVD2 wakeup flag bit	0: No PVD2 wakeup event occurred 1: A PVD2 wakeup event occurs After power-down wake-up, you need to write zero to clear this bit.	RW
b4	VD1WKF	PVD1 wake-up flag bit	0: No PVD1 wake-up event occurred 1: A PVD1 wakeup event occurs After power-down wake-up, you need to write zero to clear this bit.	RW
b3	PTWK3F	PTWK3 wakeup flag bit	0: No PTWK3 wakeup event occurs 1: A PTWK3 wakeup event occurs After power-down wake-up, you need to write zero to clear this bit.	RW
b2	PTWK2F	PTWK2 wakeup flag bit	0: No PTWK2 wakeup event occurred 1: A PTWK2 wakeup event occurs After power-down wake-up, you need to write zero to clear this bit.	RW
b1	PTWK1F	PTWK1 wakeup flag bit	0: No PTWK1 wakeup event occurred 1: A PTWK1 wakeup event occurs After power-down wake-up, you need to write zero to clear this bit.	RW
b0	PTWK0F	PTWK0 wakeup flag bit	0: No PTWK0 wakeup event occurred 1: A PTWK0 wakeup event occurs After power-down wake-up, you need to write zero to clear this bit.	RW

5.7.9 Power-down Wake-up Flag Register 1 (PWC_PDWKF1)

Reset value: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
WKTMWKF	Res	RTCALMWKF	RTCPRDWKF	Res			

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b7	WKTMWKF	WKTM wakeup flag bit	0: No WKTM wakeup event occurs 1: A WKTM wakeup event occurs After power-down wake-up, you need to write zero to clear this bit.	RW
b6	Res	Reserved Bit	Read at “0”, write at “0”	RW
b5	RTCALMWKF	RTC alarm clock wake-up flag bit	0: No RTC alarm wakeup event occurs 1: RTC alarm clock wake-up event occurs After power-down wake-up, you need to write zero to clear this bit.	RW
b4	RTCPRDWKF	RTC cycle wake-up flag bit	0: No RTC cycle wake-up event occurs 1: RTC cycle wake-up event occurs After power-down wake-up, you need to write zero to clear this bit.	RW
b3~b0	Res	Reserved Bit	Read at “0”, write at “0”	RW

5.7.10 Power Mode Control Register 5 (PWC_PWRC5)

Reset value: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
CSDIS	Res					SRAMBSD	VVDRSD

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b7	CSDIS	Current source invalid control	0: current source is active 1: The current source is invalid When XTAL32/RTC/WTKM/Ret SRAM are not needed, after turning off XTAL32 and VDDR domain RLDO, CSDIS can be set to reduce Power Consumption.	RW
b6~b2	Res	Reserved Bit	Read at “0”, write at “0”	RW
b1	SRAMBSD	Ret SRAM (Maintain RAM)Power Down Control	0: RetS RAM does not power down 1: RetS RAM power down	RW
b0	VVDRSD	RLDO close control bit	0: RLDO is valid 1: RLDO invalid When VVDRSD is set and the chip enters power-down mode, the power supply of RTC/ WKTM/ Ret SRAM is turned off to reduce Power Consumption.	RW

5.7.11 Power Mode Control Register 6 (PWC_PWRC6)

Reset value: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
Res.						RTCCKSEL[1:0]	

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b7~b2	Res	Reserved Bit	Read at “0”, write at “0”	RW
b1~b0	RTCCKSEL[1:0]	RTC clock selection	00: LRC 01：XTAL fractional division 10: Terminal Input Other: prohibitions.	RW

5.7.12 Low Power Status Register (PWC_LPMCSR)

Reset value: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res															LPMF

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b15~b1	Res	Reserved Bit	Read “0” when reading, write“0” when writing	RW
b0	LPMF	Low Power Mode Flag	0: The chip has not entered the Stop Mode or power-down mode 1: The chip enters Stop Mode **Note:** After the chip enters power-down mode, this bit is cleared	R

5.7.13 Low Power State Clear Register (PWC_LPMCSCR)

Reset value: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res															LPMFC

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b15~b1	Res	Reserved Bit	Read “0” when reading, write“0” when writing	RW
b0	LPMFC	LPMF clear bit	0: Do not clear the LPMF flag 1：Clear the LPMF flag	RW

5.7.14 Function Clock Control 0 (PWC_FCG0)

Reset value: 0xFFFF FA0E

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res								CRC	Res				CTC	AOS	FCM
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res	DMA	Res		PLA	SRAM B	Res					SRAM 0	Res			SRAM H

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b24	Res	Reserved Bit	All bits are "1" when reading, and all bits are written "1" when writing	RW
b23	CRC	CRC function control	0: CRC function enable 1: Invalid CRC function	RW
b22~b19	Res	Reserved Bit	Read at “1”, write at “1”	RW
b18	CTC	CTC Functional Control	0: Internal clock calibrator CTC function enable 1: Internal clock calibrator CTC function invalid	RW
b17	AOS	AOS Functional Control	0: Automatic operation system AOS function enable 1: Automatic operation system AOS function invalid	RW
b16	FCM	FCM Functional Control	0: The clock frequency measurement module FCM function in the clock controller CMU is enabled 1: The function of the clock frequency measurement module FCM in the clock controller CMU is invalid	RW
b15	Res	Reserved Bit	Read at “1”, write at “1”	RW
b14	DMA	DMA Functional Control	0: DMA function enable 1: Invalid DMA function	RW
b13~b12	Res	Reserved Bit	When reading, all bits are "1", and when writing, all bits are written "1".	RW
b11	PLA	PLA Functional Control	0: PLA function enable 1: Invalid PLA function	RW
b10	SRAMB	Ret SRAM Functional Control	0: Ret SRAM function enable 1: Invalid Ret SRAM function	RW
b9-b5	Res	Reserved Bit	Read at “10000”, write at “10000”	RW
b4	SRAM0	SRAM0 Functional Control	0: SRAM0 function enabled 1: Invalid SRAM0 function	RW
b3~b1	Res	Reserved Bit	When reading, all bits are "1", and when writing, all bits are written "1".	RW
b0	SRAMH	SRAMH Functional Control	0: SRAMH function enable 1: Invalid SRAMH function	RW

5.7.15 Function Clock Control 1 (PWC_FCG1)

Reset value: 0xFFFF FFFF

b31	b30	b29	b28	b27	b26	b25	b24
Res							
b23	b22	b21	b20	b19	b18	b17	b16
Res							SPI
b15	b14	b13	b12	b11	b10	b9	b8
Res							
b7	b6	b5	b4	b3	b2	b1	b0
Res			I2C	Res		MCAN2	MCAN1

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b17	Res	Reserved Bit	All bits are "1" when reading, and all bits are written "1" when writing	RW
b16	SPI1	SPI1 Functional Control	0: Serial peripheral interface SPI Unit 1 Function Enable 1: SPI Unit 1 Function Invalid	RW
b15~b5	Res	Reserved Bit	When reading, all bits are "1", and when writing, all bits are written "1".	RW
b4	I2C	I2C Functional Control	0： Integrated circuit bus I2C function enable 1： IC bus I2C function invalid	RW
b3~b2	Res	Reserved Bit	Read at “1”, write at “1”	RW
b1	MCAN2	MCAN2 Functional Control	0: MCAN Unit 2 Function Enable 1: MCAN Unit 2 Function Invalid	RW
b0	MCAN1	MCAN1 Functional Control	0: MCAN Unit 1 Function Enable 1: MCAN Unit 1 Function Invalid	RW

5.7.16 Function Clock Control 2 (PWC_FCG2)

Reset value: 0xFFFF FFFF

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res							TMRA_5	TMRA_4	TMRA_3	TMRA_2	TMRA_1	Res			
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
EMB	Res	TMR0_2	TMR0_1	Res	TMR4	HRPWM_6	HRPWM_5	HRPWM_4	HRPWM_3	HRPWM_2	HRPWM_1	TMR6_4	TMR6_3	TMR6_2	TMR6_1

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b25	Res	Reserved Bit	When reading, all bits are "1", and when writing, all bits are written "1".	RW
b24	TMRA_5	TimerA_5 Functional Control	0: TimerA Unit 5 Function Enable 1: TimerA Unit 5 Function Invalid	RW
b23	TMRA_4	TimerA_4 Functional Control	0: TimerA Unit 4 Function Enable 1: TimerA Unit 4 Function Invalid	RW
b22	TMRA_3	TimerA_3 Functional Control	0: TimerA Unit 3 Function Enable 1: TimerA Unit 3 Function Invalid	RW
b21	TMRA_2	TimerA_2 Functional Control	0: TimerA Unit 2 Function Enable 1: TimerA Unit 2 Function Invalid	RW
b20	TMRA_1	TimerA_1 Functional Control	0: TimerA Unit 1 Function Enable 1: TimerA Unit 1 Function Invalid	RW
b19~b16	Res	Reserved Bit	All bits are "1" when reading, and all bits are "1" when writing	RW
b15	EMB	EMB Functional Control	0： Enable the EMB function of the emergency brake module 1： Emergency brake module EMB function invalid	RW
b14	Res	Reserved Bit	Read at “1”, write at “1”	RW
b13	TMR0_2	Timer0_2 Functional Control	0: Timer0 Unit 2 Function Enable 1: Timer0 Unit 2 Function Invalid	RW
b12	TMR0_1	Timer0_1 Functional Control	0: Timer0 Unit 1 Function Enable 1: Timer0 Unit 1 Function Invalid	RW
b11	Res	Reserved Bit	Read at “1”, write at “1”	RW
b10	TMR4	TimerR4 Functional Control	0: Timer4 Function Unit Enable 1: Timer4 Function Unit Invalid	RW
b9	HRPWM_6	HRPWM Functional Control	0: HRPWM functional unit 6 enable 1: HRPWM function unit 6 invalid	RW
b8	HRPWM_5	HRPWM Functional Control	0: HRPWM functional unit 5 enable 1: HRPWM function unit 5 invalid	RW
b7	HRPWM_4	HRPWM Functional Control	0: HRPWM functional unit 4 enable 1: HRPWM function unit 4 invalid	RW
b6	HRPWM_3	HRPWM Functional Control	0: HRPWM functional unit 3 enable 1: HRPWM function unit 3 invalid	RW
b5	HRPWM_2	HRPWM Functional Control	0: HRPWM functional unit 2 enable 1: HRPWM function unit 2 invalid	RW
b4	HRPWM_1	HRPWM Functional Control	0: HRPWM functional unit 1 enable 1: HRPWM function unit 1 invalid	RW
b3	TMR6_4	Timer6_4 Functional Control	0: Timer6 Unit 4 Function Enable 1: Timer6 Unit 4 Function Invalid	RW
b2	TMR6_3	Timer6_3 Functional Control	0: Timer6 Unit 3 Function Enable 1: Timer6 Unit 3 Function Invalid	RW
b1	TMR6_2	TMR6_2 Functional Control	0: Timer6 Unit 2 Function Enable 1: Timer6 Unit 2 Function Invalid	RW
b0	TMR6_1	TMR6_1 Functional Control	0: Timer6 Unit 1 Function Enable 1: Timer6 Unit 1 Function Invalid	RW

5.7.17 Function Clock Control 3 (PWC_FCG3)

Reset value: 0xFFFF FFFF

b31	b30	b29	b28	b27	b26	b25	b24
Res							
b23	b22	b21	b20	b19	b18	b17	b16
USART4	USART3	USART2	USART1	Res			
b15	b14	b13	b12	b11	b10	b9	b8
Res						CMP3	CMP12
b7	b6	b5	b4	b3	b2	b1	b0
Res		DAC2	DAC1	Res	ADC3	ADC2	ADC1

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b24	Res	Reserved Bit	When reading, all bits are "1", and when writing, all bits are written "1".	RW
b23	USART4	USART4 Functional Control	0: Universal Synchronous Asynchronous Receiver/Transmitter USART Unit 4 Function Enable 1: Universal Synchronous Asynchronous Receiver-Transmitter USART Unit 4 Function Invalid	RW
b22	USART3	USART3 Functional Control	0: Universal Synchronous Asynchronous Receiver/Transmitter USART Unit 3 Function Enable 1: Universal Synchronous Asynchronous Receiver-Transmitter USART Unit 3 Function Invalid	RW
b21	USART2	USART2 Functional Control	0: Universal Synchronous Asynchronous Receiver/Transmitter USART Unit 2 Function Enable 1: Universal Synchronous Asynchronous Receiver-Transmitter USART Unit 2 Function Invalid	RW
b20	USART1	USART1 Functional Control	0: Universal Synchronous Asynchronous Receiver/Transmitter USART Unit 1 Function Enable 1: Universal Synchronous Asynchronous Receiver-Transmitter USART Unit 1 Function Invalid	RW
b19~b10	Res	Reserved Bit	All bits are "1" when reading, and all bits are written "1" when writing	RW
b9	CMP3	CMP3 Functional Control	0: Voltage Comparator CMP Channel 3 Function Enable 1: Voltage Comparator CMP Channel 3 Function Invalid	RW
b8	CMP12	CMP1 and CMP2 function control	0: Voltage Comparator CMP Channel 1 and Channel 2 Functions Enable 1: Voltage Comparator CMP Channel 1 and Channel 2 Functions Invalid.	RW
b7~b6	Res	Reserved Bit	All bits are "1" when reading, and all bits are written "1" when writing	RW
b4	DAC2	DAC2 Functional Control	0: DAC unit 2 Function Enable 1: DAC Unit 2 Function Invalid	RW
b4	DAC1	DAC1 Functional Control	0: DAC unit 1 Function Enable 1: DAC Unit 1 Function Invalid	RW
b3	Res	Reserved Bit	All bits are "1" when reading, and all bits are written "1" when writing	RW
b2	ADC3	ADC3 Functional Control	0: Analog-to-Digital Conversion Module ADC unit 3 function enable 1: Analog-to-Digital Conversion Module ADC unit 3 function invalid	RW
b1	ADC2	ADC2 Functional Control	0: Analog-to-Digital Conversion Module ADC unit 2 function enable 1: Analog-to-Digital Conversion Module ADC unit 2 function invalid	RW
b0	ADC1	ADC1 Functional Control	0: Analog-to-Digital Conversion Module ADC unit 1 function enable 1: Analog-to-Digital Conversion Module ADC unit 1 function invalid	RW

5.7.18 PWC_FCG0 Protection Control (PWC_FCG0PC)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
FCG0PCWE[15:0]															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res															PRT0

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b16	FCG0PCWE[15:0]	PWC_FCG0PC write enable	Change the value of the PRT0 bit while writing 0xA5A5	RW
b15~b1	Res	Reserved Bit	Read at “0”, write at “0”	RW
b0	PRT0	Protection bit	PWC_FCG0 write enable control bit 0: PWC_FCG0 write invalid 1: PWC_FCG0 write enable	RW

5.7.19 Function Protection Control Register (PWC_FPRC)

Reset value: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
PWC_FPRCWE[7:0]								FPRCB7	FPRCB6	FPRCB5	FPRCB4	FPRCB3	Res	FPRCB1	FPRCB0

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b15~b8	PWC_FPRCWE[7:0]	PWC_FPRC register write enable	Write to 0xA5 to update the PWC_FPRC value, otherwise it is invalid for the lower 8-bit write value. Read at 0x00.	RW
b7	FPRCB7	FPRC bit 7	Reserved, "0" when reading, "0" when writing	RW
b6	FPRCB6	FPRC bit 6	Reserved, "0" when reading, "0" when writing	RW
b5	FPRCB5	FPRC bit 5	Reserved, "0" when reading, "0" when writing	RW
b4	FPRCB4	FPRC bit 4	Reserved, "0" when reading, "0" when writing	RW
b3	FPRCB3	FPRC bit 3	Protection register bits, protection objects refer to Table 5-8 0: Write protection 1: Write enable	RW
b2	Res	Reserved Bit	Reserved, "0" when reading, "0" when writing	RW
b1	FPRCB1	FPRC bit 1	Protection register bits, protection objects refer to Table 5-8 0: Write protection 1: Write enable	RW
b0	FPRCB0	FPRC bit 0	Protection register bits, protection objects refer to Table 5-8 0: Write protection 1: Write enable	RW

5.7.20 Stop Mode Control Register (PWC_STPMCR)

Reset value: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
STOP	Res													CKSMRC	FLNWT

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b15	STOP	Stop Mode selection bit	0: Stop Mode is invalid 1: Stop Mode is valid	RW
b14~b2	Res	Reserved Bit	Read at “0”, write at “0”	RW
b1	CKSMRC	Clock switch to MRC option	0: Maintain the system clock and frequency division before entering Stop Mode 1: The system clock switches to MRC and the SCKCFGR register is initialized when waking up from Stop Mode	RW
b0	FLNWT	FLASH stable waiting control	0： Wait for FLASH to stabilize when waking up from Stop Mode 1： Do not wait for FLASH to stabilize when waking up from Stop Mode	RW

5.7.21 RAM Power Control Register 0 (PWC_RAMPC0)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24
Res							
b23	b22	b21	b20	b19	b18	b17	b16
Res							
b15	b14	b13	b12	b11	b10	b9	b8
Res					RAMDC10	Res	
b7	b6	b5	b4	b3	b2	b1	b0
Res							RAMPDC0

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b32~b11	Res	Reserved Bit	Read at “0”, write at “0”	RW
b10	RAMPDC10	RAM power down control bit 10	0: SRAMH action 1: SRAMH power down	RW
b9~b1	Res	Reserved Bit	Read at “0”, write at “0”	
b0	RAMPDC0	RAM power down control bit 0	0: SRAM0 action 1: SRAM0 power down	RW

5.7.22 Peripheral RAM Low Power Control Register (PWC_PRAMLPC)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24
Res							
b23	b22	b21	b20	b19	b18	b17	b16
Res							
b15	b14	b13	b12	b11	b10	b9	b8
Res							
b7	b6	b5	b4	b3	b2	b1	b0
Res					PRAMPDC2	Res	PRAMPDC0

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b3	Res	Reserved Bit	Read at “0”, write at “0”	RW
b2	PRAMPDC2	Peripheral RAM Power-down Control Bit 2	CACHE RAM power down control 0: No power down 1: power down	RW
b1	Res	Reserved Bit	Read at “0”, write at “0”	RW
b0	PRAMPDC0	Peripheral RAM Power-down Control Bit 0	MCAN RAM power down control 0: No power down 1: power down	RW

5.7.23 PVD Control Register 0 (PWC_PVDCR0)

Reset value: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
Res	PVD2EN	PVD1EN	Res				EXVCCINEN

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b7	Res	Reserved Bit	Read at “0”, write at “0”	RW
b6	PVD2EN	Voltage detection 2 permissible	0: Voltage detection 2 circuit is invalid 1: The voltage detection 2 circuit is valid	RW
b5	PVD1EN	Voltage detection 1 permissible	0: Voltage detection 1 circuit is invalid 1: The voltage detection 1 circuit is valid	RW
b4~b1	Res	Reserved Bit	Read at “0”, write at “0”	RW
b0	EXVCCINEN	External VCC voltage input enable	0: External VCC voltage input is invalid 1: External VCC voltage input is valid	RW

5.7.24 PVD Control Register 1 (PWC_PVDCR1)

Reset value: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
Res	PVD2CMPOE	PVD2IRS	PVD2IRE	Res	PVD1CMPOE	PVD1IRS	PVD1IRE

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b7	Res	Reserved Bit	Read at “0”, write at “0”	RW
b6	PVD2CMPOE	PVD2 comparison result output enable	0: Disable output of comparison result of comparator 2 1: Allow to output the comparison result of comparator 2	RW
b5	PVD2IRS	PVD2 interrupt reset selection	0: Generate PVD2 interrupt when VCC changes to meet the detection conditions 1: During the falling process, VPVD2 generates a PVD2 reset. **Note:** When the PVD1IRS bit is "1" or the PVD2IRS bit is "1", you cannot enter the power-down mode. To enter the PD mode, you must set the PVD1IRS bit to "0" and the PVD2IRS bit to "0".	RW
b4	PVD2IRE	PVD2 interrupt reset enable	0: Prohibited 1: Permissible **Note:** Please write "1" to the PVD2IRE bit when the PVD2EN bit is "1" and the PVD2CMPOE bit is "1"	RW
b3	Res	Reserved Bit	Read at “0”, write at “0”	RW
b2	PVD1CMPOE	PVD1 comparison result output enable	0: Disable output of comparison result of comparator 1 1: Allow to output the comparison result of comparator 1	RW
b1	PVD1IRS	PVD1 interrupt reset selection	0: Generate PVD1 interrupt when VCC changes to meet the detection conditions 1: During the falling process, VPVD1 generates PVD1 reset **Note 1:** When the PVD1IRS bit is "1" or the PVD2IRS bit is "1", you cannot enter the power-down mode. To enter the PD mode, you must set the PVD1IRS bit to "0" and the PVD2IRS bit to "0".	RW
b0	PVD1IRE	PVD1 interrupt reset enable	0: Prohibited 1: Permissible **Note:** Please write "1" to the PVD1IRE bit when the PVD1EN bit is "1" and the PVD1CMPOE bit is "1"	RW

5.7.25 PVD Filter Control Register (PWC_PVDFCR)

Reset value: 0x11

b7	b6	b5	b4	b3	b2	b1	b0
Res	PVD2NFCKS[1:0]		PVD2NFDIS	Res	PVD1NFCKS[1:0]		PVD1NFDIS

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b7	Res	Reserved Bit	Read at “0”, write at “0”	RW
b6~b5	PVD2NFCKS	PVD2 digital filter sampling clock selection	00: 2 filter clock cycles 01: 4 filter clock cycles 10: 8 filter clock cycles 11: 16 filter clock cycles Filter clock period = LRC period / 8 **Note:** This bit can only be rewritten when the PVD2NFDIS bit is "1"	RW
b4	PVD2NFDIS	PVD2 Digital Filter Shield	0: Digital filter valid 1: Invalid digital filter	RW
b3	Res	Reserved Bit	Read at “0”, write at “0”	RW
b2~b1	PVD1NFCKS	PVD1 digital filter sampling clock selection	00: 2 filter clock cycles 01: 4 filter clock cycles 10: 8 filter clock cycles 11: 16 filter clock cycles Filter clock period = LRC period / 8 **Note:** This bit can only be rewritten when the PVD1NFDIS bit is "1"	RW
b0	PVD1NFDIS	PVD1 Digital Filter Shield	0: Digital filter valid 1: Invalid digital filter	RW

5.7.26 PVD Level Control Register (PWC_PVDLCR)

Reset value: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
Res	PVD2LVL[2:0]			Res	PVD1LVL[2:0]		

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b7	Res	Reserved Bit	Read at “0”, write at “0”	RW
b6~b4	PVD2LVL[2:0]	PVD2 threshold voltage selection	000：2.1V 001：2.3V 010：2.5V 011：2.6V 100：2.7V 101：2.8V 110：2.9V 111: 1.1V (only valid when PWC_PVDCR0.EXVCCINEN=1, please do not set this value in other cases)	RW
b3	Res	Reserved Bit	Read at “0”, write at “0”	RW
b2~b0	PVD1LVL[2:0]	PVD1 threshold voltage selection	000：2.0V 001：2.1V 010：2.3V 011：2.5V 100：2.6V 101：2.7V 110：2.8V 111：2.9V	RW

5.7.27 PVD Interrupt Control Register (PWC_PVDICR)

Reset value: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
Res	PVD2EDGS[1:0]		PVD2NMIS	Res	PVD1EDGS[1:0]		PVD1NMIS

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b7	Res	Reserved Bit	Read at “0”, write at “0”	RW
b6~b5	PVD2EDGS[1:0]	PVD2 detection condition selection	00: When VCC < VPVD2 (falling) is detected 01: When VCC ≥ VPVD2 (rising) is detected 10: When VCC < VPVD2 (falling) is detected or when VCC ≥ VPVD2 (rising) is detected 11: Prohibitions	RW
b4	PVD2NMIS	PVD2 Interrupt Type	0: PVD2 interrupt is a non-maskable interrupt 1: PVD2 interrupt as a maskable interrupt	RW
b3	Res	Reserved Bit	Read at “0”, write at “0”	RW
b2~b1	PVD1EDGS[1:0]	PVD1 Detection Condition Selection	00: When VCC < VPVD1 (falling) is detected 01: When VCC ≥ VPVD1 (rising) is detected 10: When VCC < VPVD1 (falling) is detected or when VCC ≥ VPVD1 (rising) is detected 11: Prohibitions	RW
b0	PVD1NMIS	PVD1 Interrupt Type	0: PVD1 interrupt is a non-maskable interrupt 1: PVD1 interrupt as a maskable interrupt	RW

5.7.28 PVD Detection Status Register (PWC_PVDDSR)

Reset value: 0x11

b7	b6	b5	b4	b3	b2	b1	b0
Res		PVD2DETFLG	PVD2MON	Res		PVD1DETFLG	PVD1MON

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b7~b6	Res	Reserved Bit	Read at “0”, write at “0”	RW
b5	PVD2DETFLG	PVD2 detection flag bit	0: PVD2 does not detect that VCC passes through VPVD2 1: PVD2 detects that VCC passes through VPVD2 Writing 0 after reading can clear this bit. **Note:** This flag is valid when the PVD2EN bit is "1" and the PVD2CMPOE bit is "1"	RW
b4	PVD2MON	PVD2 Monitoring Bit	0: VCC ≤ VPVD2 or external input comparison voltage ≤ PVD2 internal reference voltage 1: When PVD2 is invalid or VCC>VPVD2 or external input comparison voltage>PVD2 internal reference voltage	R
b3~b2	Res	Reserved Bit	Read at “0”, write at “0”	RW
b1	PVD1DETFLG	PVD1 detection flag bit	0: PVD1 does not detect that VCC passes through VPVD1 1: PVD1 detects that VCC passes through VPVD1 Writing 0 after reading can clear this bit. **Note:** This flag is valid when the PVD1EN bit is "1" and the PVD1CMPOE bit is "1"	RW
b0	PVD1MON	PVD1 Monitoring Bit	0: VCC<VPVD1 1: When PVD1 is invalid or VCC>VPVD1	R

5.7.29 Wake-up Timer Control Register (PWC_WKTCR)

Reset value: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
WKTCE	WKCKS[1:0]		WKOVF	WKTMCMP[11:0]											

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b15	WKTCE	WKTM enable	0: WKTM stop 1： WKTM count Clearing the WKTCE bit while WKTM is counting will clear the counter and stop the counter.	RW
b14~b13	WKCKS[1:0]	WKTM clock selection	00: 64Hz clock 01：XTAL32 10：LRC 11: Reserved	RW
b12	WKOVF	Timer comparison result flag	0: The counter is inconsistent with the WKTMCMP value 1: The counter is consistent with the WKTMCMP value Writing 0 clears this flag.	RW
b11~b0	WKTMCMP[11:0]	WKTM compare bit	Comparison value of WKTM counter	RW

6 Initialization Configuration (ICG)

6.1 Overview

After the chip reset is released, the hardware circuit will read the Flash address 0x0000 0400~0x0000 045F and load the data into the initialization configuration register. Addresses 0x0000 0408~0x0000 040B, 0x0000 0410~0x0000 041F, 0x0000 0438~0x0000 045F are reserved addresses, please write all 1s to ensure normal operation of the chip. When Flash boot swap invalid, this area is located in FLASH sector 0; when Flash boot swap valid and the OTP of Flash sector 0 is not latched (0x0300 0A80~0x0300 0A83 data are all 1), this area is located in Flash sector 1; otherwise, this area is located in Flash sector 0. The user can modify the Initialization Configuration (ICG) register by programming or erasing Sector 0. Address 0x0000 0420~0x0000 0437 is the data security protection enable area. For detailed specifications, see [Data Security Protection]. The register reset value is determined by the Flash data.

6.2 Register Description

Table 6-1 ICG Base Address

Name	Base Address	Description
ICG	0x0000 0400	ICG基地址

Table 6-2 ICG Rgister List

Register	Description	Offset Address	Bit Width	Reset Value
ICG0	Initialize Configuration Register0	0x000	32	Indefinite
ICG1	Initialize Configuration Register 1	0x004	32	Indefinite
ICG3	Initialize Configuration Register 3	0x00C	32	Indefinite

6.2.1 Initialization Configuration Register 0 (ICG0)

Reset value:Indefinite

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res			WDTSLPOFF	WDTWDPT]3:0]				WDTCKS[3:0]				WDTPERI[1:0]		WDTITS	WDTAUTS
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res			SWDTSLPOFF	SWDTWDPT]3:0]				SWDTCKS[3:0]				SWDTPERI[1:0]		SWDTITS	SWDTAUTS

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b29	Res	Reserved bit	Functional reservation bit	R
b28	WDTSLPOFF	WDT counting stops in Sleep Mode	0: WDT counting does not stop in Sleep Mode 1: WDT counting stops in Sleep Mode	R
b27~b24	WDTWDPT[3:0]	Refresh Allowed Area Count%	WDT Count Refresh Allowed Interval 0000：0%~100% 0001：0%~25% 0010：25%~50% 0011：0%~50% 0100：50%~75% 0101：0%~25%，50%~75% 0110：25%~75% 0111：0%~75% 1000：75%~100% 1001：0%~25%，75%~100% 1010：25%~50%，75%~100% 1011：0%~50%，75%~100% 1100：50%~100% 1101：0%~25%，50%~100% 1110：25%~100% 1111：0%~100%	R
b23~b20	WDTCKS[3:0]	WDT counter clock	0010：PCLK3/4 0110：PCLK3/64 0111：PCLK3/128 1000：PCLK3/256 1001：PCLK3/512 1010：PCLK3/1024 1011：PCLK3/2048 1101：PCLK3/8192 Other values: Reserved	R
b19~b18	WDTPERI[1:0]	WDT count overflow cycle	00: 256 cycles 01: 4096 cycles 10: 16384 cycles 11: 65536 cycles	R
b17	WDTITS	WDT interrupt selection	0: Interrupt request 1: Reset request	R
b16	WDTAUTS	WDT auto start	0: WDT starts automatically after reset (hardware starts) 1: WDT stops after reset	R
b15~b13	Res	Reserved Bit	Functional reservation bit	R

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b12	SWDTSLPOFF	SWDT count stops in Sleep, Stop Mode	0: SWDT does not stop counting in Sleep Mode and Stop Mode 1: SWDT stops counting in Sleep Mode and Stop Mode	R
b11~b8	SWDTWDPT[3:0]	Refresh Allowed Area Count%	SWDT Count Refresh Allowed Interval 0000：0%~100% 0001：0%~25% 0010：25%~50% 0011：0%~50% 0100：50%~75% 0101：0%~25%，50%~75% 0110：25%~75% 0111：0%~75% 1000：75%~100% 1001：0%~25%，75%~100% 1010：25%~50%，75%~100% 1011：0%~50%，75%~100% 1100：50%~100% 1101：0%~25%，50%~100% 1110：25%~100% 1111：0%~100%	R
b7~b4	SWDTCKS[3:0]	SWDT counter clock	0000：SWDTCLK 0100：SWDTCLK/16 0101：SWDTCLK/32 0110：SWDTCLK/64 0111：SWDTCLK/128 1000：SWDTCLK/256 1011：SWDTCLK/2048 Other values: Reserved	R
b3~b2	SWDTPERI[1:0]	SWDT count overflow cycle	00: 256 cycles 01: 4096 cycles 10: 16384 cycles 11: 65536 cycles	R
b1	SWDTITS	SWDTinterrupt selection	0: Interrupt request 1: Reset request	R
b0	SWDTAUTS	SWDT auto start	0: SWDT starts automatically after reset (hardware starts) 1: SWDT stops after reset	R

6.2.2 Initialization Configuration Register 1 (ICG1)

Reset value: Indefinite

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res													BOR DIS	BOR_LEV[1:0]	
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res							HRC STOP	Res							HRCF REQS

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b19	Res	Reserved Bit	Functional reservation bit	R
b18	BORDIS	BOR action selection	0: Allow BOR action after reset 1: Disable BOR action after reset	R
b17~b16	BOR_LEV[1:0]	BOR threshold voltage selection	00：1.9V 01：2.0V 10：2.1V 11：2.3V	R
b15~b9	Res	Reserved Bit	Functional reservation bit	R
b8	HRCSTOP	HRC Oscillation Stop Bit	0: HRC oscillation 1: HRC stop	R
b7~b1	Res	Reserved Bit	When reading, all bits are "1", and when writing, all bits are written "1".	R
b0	HRCFREQSEL	HRC Frequency Slection	0：20MHz 1：16MHz	R

6.2.3 Initialization Configuration Register 3 (ICG3)

Reset value: Indefinite

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res											DBUSPSZ[4:0]				
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
DBUSPRT[15:0]															

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b21	Res	Reserved Bit	Reserved, "1" when reading, "1" when writing	R
b20~b16	DBUSPSZ[4:0]	D-BUS Read Protection Space	Set D-BUS Read Protection Space. Valid when DBUSPRT[15:0]=0x4450. 00000~ 01100: Reserved, setting is prohibited 01101：8KB 01110：16KB 01111：32KB 10000: 64KB (64KB product settings prohibited) 10001~ 11111: Reserved, setting is prohibited	R
b15~b0	DBUSPRT[15:0]	D-BUS Read Protection Function	When DBUSPRT[15:0]=0x4450, Enable D-BUS read protection for the space starting from 0x0000 0000; otherwise, D-BUS read protection invalid.	R

7 Embedded Flash Memory (EFM)

7.1 Overview

The Flash interface accesses the Flash through the ICODE, DCODE, and MCODE buses, and can perform programming, erasing, and full-erase operations on the Flash; it accelerates code execution through instruction prefetching and caching mechanisms.

7.2 Main Features

- Maximum 5KBytes of OTP space
- ICODE bus 16Bytes prefetch instruction
- Two independent buffers: ICODE bus buffer space 1KBytes; DCODE bus buffer space 128Bytes
- Support boot swap function
- Support Data Security Protection

7.3 Embedded Flash

Flash has the following main features:

- The capacity is up to 128KBytes, consisting of a 128KBytes Flash, a total of 32 sectors, each sector is 4KBytes. Sector 0 can be configured as an OTP area.
- The OTP (One Time Program) area is 5KBytes in total.
  4KBytes are configured at address 0x0000 0000~0x0000 0FFF (sector 0), and 1KBytes are configured at address 0x0300 0C00~0x0300 0FFF.
  Address 0x0300 0A80~0x0300 0AFF is the OTP data latch area. 0x0300 0A00~0x0300 0A03 is the OTP enable area.
- 64-bit wide data read and 64-bit wide read cache buffer accelerates code execution.
- The programming unit is 4Bytes, and the erasing unit is 4KBytes.

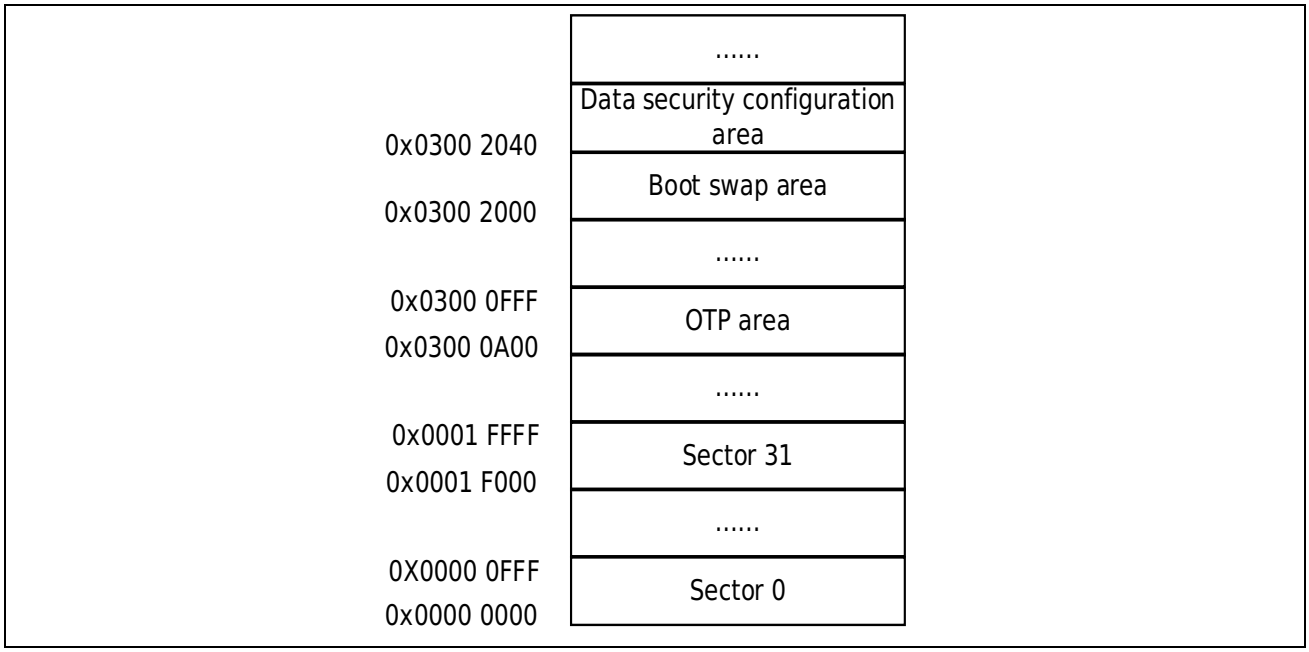


Figure 7-1 Flash AddressAllocation

The OTP area 0x0300 0A00~0x0300 0FFF has 1 sector and can only be programmed. Erasing/full erasing is prohibited.

The boot swap enable address 0x0300 2000~0x0300 2003 and the data security configuration area 0x0300 2040~0x0300 204B share the same sector and can be programmed and erased, but full erase is prohibited. The data security configuration area 0x0300 2040~0x0300 204B needs to be reprogrammed after each erasure.

These two sectors are physically subordinate to FLASH, and the erase flag follows FLASH, but the data of these two sectors will remain unchanged if the FLASH is completely erased.

7.4 Read interface

7.4.1 Relationship between CPU clock and Flash read time

To read Flash data correctly, the user needs to correctly set the number of wait cycles (FLWT[3:0]) in the Flash Read Mode Register (EFM_FRMC) according to the CPU operating frequency.

After the system is reset, the CPU clock source is MRC (8MHz) and the Flash read wait cycle is 0. It is recommended that users modify the CPU main frequency and Flash read wait cycle bit according to the following steps. Please refer to the number of waiting cyclesTable 7-1.

CPU Frequency Increase Step:

1. Write the new read wait period setting value (FLWT[3:0]) into the register EFM_FRMC.
2. Read registerEFM _ FRMC and check whether the new waiting period is set successfully.
3. Increase the CPU clock frequency by setting the system clock source switching register CMU_CKSWR (CKSW[2:0]) or the system clock configuration register CMU_SCFGR (HCLKS[2:0]).
4. Read the register CMU_CKSWR or CMU_SCFGR to check whether the new setting is successful.

CPU Frequency Reduction Step:

1. Reduce the CPU clock frequency by setting the system clock source switching register CMU_CKSWR (CKSW[2:0]) or the system clock configuration register CMU_SCFGR (HCLKS[2:0]).
2. Read the register CMU_CKSWR or CMU_SCFGR to check whether the new setting is successful.
3. Write the new read wait period setting value (FLWT[3:0]) into the register EFM_FRMC.
4. Read registerEFM _ FRMC and check whether the new waiting period is set successfully.

Table 7-1 CPU Clock Frequency and Flash Read Wait Cycle Comparison Table

CPU Clock Frequency (hclk)	Read Wait Cycle Setting
100MHz<HCLK≤150MHz	FLWT[3:0]=0b0010
50MHz<HCLK≤100MHz	FLWT[3:0]=0b0001
HCLK≤50MHz	FLWT[3:0]=0b0000

7.5 Flash Read Cache

Each Flash read operation is a 64-bit read. The data is sent to the CPU and stored in the buffer register at the same time. The 64-bit data can be 2 lines of 32-bit instructions or 4 lines of 16-bit instructions, depending on the program burned into the Flash.

In order to quickly read Flash data, the Flash controller is configured with a read acceleration cache to optimize the read wait cycle. In order to maximize the performance of the processor, the accelerator saves the ICODE and DCODE bus access data of the Flash into the cache memory, thereby improving the program execution speed.

The system provides 1.125Kbytes of space as cache memory to effectively reduce the time loss caused by instruction jump. The cache function is enabled by setting the Cache Enable (ICACHE/DCACHE) bit in the EFM_FRMC register to 1. Whenever an instruction or data miss occurs (that is, the requested instruction or data is not present in the currently used instruction line or cache memory), the system copies the newly read data line (64 bits) into the cache memory. If the CPU requests an instruction or data that already exists in the cache, it can be acquired immediately without any delay. After the cache is full, the LRU (least recently used) policy is used to determine the data to be replaced in the cache.

When the CPU reads instructions or data, the Flash address is in the buffer. When the cache hits, the number of Flash read cycles will change. For details, please refer to Table 7-2.

Table 7-2 Flash Actual Read Cycle Number

EFM_FRMC. FLWT[3:0] Setting	Cache not Enabled		Cache Enable	
	Buffer Hit	Buffer Miss	Buffer or Cache Hit	Buffer, Cache Miss
0	1	1	1	1
1	1	2	1	2
2	1	3	1	3
N (N>2)	1	N+1	1	N+1

7.6 Flash Programming and Erasing Operations

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

For Flash programming, the last bit of the sector/erase address must be aligned with 4 (the last bit of the address is: 0x0, 0x4, 0x8, 0xC), the programming unit is 4Bytes, and the sector erase unit is 4KBytes. Flash programming methods are divided into three types: single programming without readback mode, single programming readback mode, and continuous programming mode. During Flash programming and erasing, set EFM_FWMC.BUSHLDCTL=0, then the bus is occupied until the erasing is completed; set EFM_FWMC.BUSHLDCTL=1, then the bus is released and the program needs to be executed on RAM to achieve programming or erasing of Flash while the program is being executed. Before programming and erasing Flash, please disable cache enable and prefetch. The following describes the setup steps for programming and erasing operations.

7.6.1 Unlock the EFM_KEY1 Register

After reset is released, the Flash programming and erase mode register (EFM_FWMC) is in a write-prohibited state. You need to release the Flash access protection register (EFM_FAPRT) first, and then release the protection of EFM_KEY1. The following steps are used to remove the above protection.

1) Remove the Flash register access write protection (EFM_FAPRT first writes 0x0123, then writes 0x3210)
2) Unlock EFM_KEY1 (write 0x0123 4567 to EFM_KEY1 first, then write 0xFEDC BA98)

If an incorrect sequence is written to EFM_KEY1, a bus error will occur and the register will be locked. The EFM_FWMC register will be write-prohibited until a reset occurs.

After writing the set value to the EFM_FWMC register correctly, if you need to return to the EFM_FWMC write-disabled state, write 1 to EFM_FWMC.KEY1LOCK.

7.6.2 Write Protection Function

Each sector of Flash has 1 write protection bit, which exists in the write protection register EFM_F0NWPRT. Each write protection register has a write protection lock bit (WLOCK0). Once the write protection lock bit is set to 1, the write protection register bit can only be set from write permission to write prohibition. When the Flash sector is set to write-prohibited, the programming and erasing operations on the sector address will not occur, and an error flag will occur (EFM_FSR.PRTWERR=1).

In full erase mode, if one or more sectors of the Flash block are set to write-prohibited, full erase does not occur and an error flag occurs (EFM_FSR.PRTWERR=1).

7.6.3 Single Programming Without Read BackFunction

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

1) Remove the Flash register write protection (EFM_FAPRT first writes 0x0123, then writes 0x3210);
2) Unlock EFM_KEY1 (write 0x0123 4567 to EFM_KEY1 first, then write 0xFEDC BA98);
3) Set the single programming mode (EFM_FWMC.PEMODE[2:0]=0b001);
4) Release write protection (set the corresponding bit of EFM_F0NWPRT to 1);
5) Write 32-bit data to the programming address;
6) Wait for Flash to be in idle state (EFM_FSR.RDY=1);
7) Read out the programming address value to judge whether it is consistent with the written value;
   If consistent, it means the programming is successful; if inconsistent, it means the Flash address has been destroyed and permanently discarded.
8) Clear the programming end flag bit (EFM _ FSR.OPTEND).

If a single programming without reply read/write operation is sent to the latched OTP address, the programming fails and the flag EFM_FSR.OTPWERR is set.

7.6.4 Single Program Readback Function

The single programming read-back mode means that after programming is completed, the hardware automatically reads the programming address data and compares it with the written data, and judges the flag bit EFM_FSR.MISMTCH to verify that the written data is correct.

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

1) Remove the Flash register write protection (EFM_FAPRT first writes 0x0123, then writes 0x3210);
2) Unlock EFM_KEY1 (write 0x0123 4567 to EFM_KEY1 first, then write 0xFEDC BA98);
3) Set the one-time programming readback mode (EFM_FWMC.PEMODE[2:0]=0b010);
4) Release write protection (set the corresponding bit of EFM_F0NWPRT to 1);
5) Write 32-bit data to the programming address;
6) Wait for Flash to be in idle state (EFM_FSR.RDY=1);
7) Determine the programming self-read result flag (EFM_FSR.MISMTCH):
   If it is 0, it means the programming is successful; if it is 1, it means the Flash address has been destroyed and permanently discarded.
8) Clear the programming end flag bit (EFM _ FSR.OPTEND).

Send a single programming read/write operation to the latched OTP address. If the programming fails, the flag EFM_FSR.OTPWERR is set.

7.6.5 Continuous ProgrammingFunction

When programming Flash addresses continuously, it is recommended to use the continuous programming mode. Continuous programming mode can save more than 50% time than single programming mode. In continuous programming mode, the interval between write commands to the Flash address cannot exceed 16μs. Continuous programming operations are set as follows:

1) Remove the Flash register write protection (EFM_FAPRT first writes 0x0123, then writes 0x3210);
2) Unlock EFM_KEY1 (write 0x0123 4567 to EFM_KEY1 first, then write 0xFEDC BA98);
3) Set continuous programming mode (EFM_FWMC.PEMOD[2:0]=0b011);
4) Release write protection (set the corresponding bit of EFM_F0NWPRT to 1);
5) Write 32-bit data to the programming address. (The execution program runs outside the Flash area);
6) Wait for the operation end flag (EFM_FSR.OPTEND) to be set;
7) Clear the operation end flag until EFM_FSR.OPTEND is read as 0;
8) Repeat 5), 6), 7) until all data is written;
9) Change the continuous programming mode to non-continuous programming mode (EFM_FWMC.PEMOD[2:0]=0b000);
10) Wait for Flash to be in idle state (EFM_FSR.RDY=1);
11) Read the programming address and determine whether it is consistent with the written value: If consistent, it means the programming is successful; if inconsistent, it means the Flash address has been destroyed and permanently discarded.

Send continuous programming write operations to the latched OTP address. If the programming fails, the flag EFM_FSR.OTPWERR is set.

7.6.6 Erase Function

After the Flash is erased, the data at the address (4KBytes space) in the sector is refreshed to all 1s. The steps to set up the sector erase operation are as follows:

1) Remove the Flash register write protection (EFM_FAPRT first writes 0x0123, then writes 0x3210);
2) Unlock EFM_KEY1 (write 0x0123 4567 to EFM_KEY1 first, then write 0xFEDC BA98);
3) Set the erase mode (EFM_FWMC.PEMOD[2:0]=0b100);
4) Release write protection (set the corresponding bit of EFM_F0NWPRT to 1);
5) Write any 32-bit value to any address in the sector to be erased (the address must be aligned to 4);

6) Wait for Flash to be in idle state (EFM_FSR.RDY=1);

7) Clear the erase end flag bit (EFM_FSR.OPTEND).

Send an erase operation to the latched OTP address. If the erase fails, the OTP area data is retained and the flag EFM_FSR.OTPWERR is set.

7.6.7 Full Erase Function

EFM provides a single-block Flash full erase method. The steps to set up the full erase operation are as follows:

1) Remove the Flash register write protection (EFM_FAPRT first writes 0x0123, then writes 0x3210);

2) Unlock EFM_KEY1 (write 0x0123 4567 to EFM_KEY1 first, then write 0xFEDC BA98);

3) When erasing a single Flash block, set EFM_FWMC.PEMOD[2:0]=0b101;

4) Remove write protection (all bits of EFM_F0NWPRT are set to 1);

5) Erase a single Flash block: Write any 32-bit value to any address of the Flash that needs to be erased (the address must be aligned to 4);

6) Wait until the Flash is idle. (EFM_FSR.RDY=1);

7) Clear the erase end flag bit. (EFM_FSR.OPTEND).

After OTP is enabled, a single-block full erase and write operation is sent to the Flash address. If the full erase fails, the Flash data is retained and the flag EFM_FSR.OTPWERR is set.

7.6.8 Data Security Protection

This product provides 4 levels of protection for Flash data to prevent untrusted users from reading and tampering with Flash through the debug interface (JTAG and SWD interface), ISP interface (In System Program) and test interface.

Protection level 0: no protection

The debug interface, ISP interface and test interface can access (read and rewrite) MCU resources, including Flash data.

Protection level 1:

After the Flash address 0x0000 0430-0x0000 0433 is programmed with data 0xAF18 0402 and the address 0x0300 2040~0x0300 204B is programmed with the 96-bit password, protection level 1 is enabled.

After protection level 1 is enabled

- The debug interface, ISP interface, and test interface cannot access MCU resources.
- Sector 0 and sector 1 cannot be programmed or erased by user programs.

- The data at address 0x0300 2040~0x0300 204B cannot be read.

After activation of protection level 1, it can be restored to protection level 0 through password authentication and full erasure. If OTP is enabled, full erase invalid. Please consult the sales office for the password authentication method and the full erasing method.

Protection level 2:

Flash address 0x0000 0434-0x0000 0437 is programmed with data 0xA851 73AE, protection level 2 is enabled. After protection level 2 is enabled

- The debug interface, ISP interface, and test interface cannot access MCU resources.
- Sector 0 and sector 1 cannot be programmed or erased by user programs.

After activation of protection level 2, it can be reverted to protection level 0 by full erasing. If OTP is enabled, full erase invalid.

Protection level 3:

When the flash address 0x0000 0420-0x0000 0423, 0x0000 0424-0x0000 0427, 0x0000 0428-0x0000 042B is programmed with data of 0x4254 5048, protection level 3 is enabled.

After protection level 3 is enabled

- The debug interface, ISP interface, and test interface cannot rewrite Flash data.
- -Sector 0 and sector 1 cannot be programmed or erased by user programs.

After activation of protection level 3, it can be reverted to protection level 0 by full erasing. If OTP is enabled, full erase invalid.

Protection level 1, protection level 2, and protection level 3 can be enabled individually or simultaneously. When enabled at the same time, the protection measures take effect superimposed.

7.6.9 D-BUS Read Protection Function

The D-BUS protection function read-protects the D-BUS read protection space set by the ICG3 register DBUSPSZ[4:0]. To enable D-BUS protection, you need to set the ICG3 register DBUSPRT[15:0] to 0x4450 and DBUSPSZ[4:0] to set the space. After reset and restart, the function is enabled. When the CPU-PC pointer is in this D-BUS read protection space (that is, the CPU executes the program in this area), the CPU can access it normally; when the CPU-PC pointer is outside this D-BUS read protection space (that is, the CPU executes the program in the space outside the D-BUS read protection), the CPU is prohibited from reading D-BUS within the D-BUS read protection space and returns a bus error. D-BUS is the data access bus of the CPU, including operand access, stack data access, Universal data access, etc.

7.6.10 Bus Hold Function

By setting registerEFM _ FWMC. BUSHLDCTL bits, FLASH programming can be set. During erasing, the bus is on hold or released. In Flash programming, when the erase instruction is executed on the

Flash, this control bit must be set to 0; when the erase instruction is executed in space outside the Flash (such as high-speed RAM), the control bit can be freely set as needed.

When BUSHLDCTL is set to 1 (Flash programming, bus release state during erasing), during programming (except continuous programming) and before the end of erasing (EFM_FSR.RDY=1), the read and write accesses to the Flash will be ignored, and the flag bit EFM_FSR.COLERR will be set.

7.6.11 Interrupt

The EFM module has three interrupts, namely PE (program/erase) error interrupt, bus conflict interrupt and operation end interrupt. The status flag can be queried by reading the EFM_FSR register.

1. PE error interruptEFM _ PEERR:

Position:

- Issue program, erase, and full erase operations to the latched OTP address (OTPWERR = 1).
- After the OTP function is enabled, the space from 0x0300 0A00 to 0x0300 0FFF is erased, and a full erase operation is performed (OTPWERR=1).
- Issue program, erase, and full erase operations to write-protected sectors (PRTWERR=1).
- The program address is not aligned by 4 or the data size is not 32 bits (PGSZERR=1).
- In single programming read-back mode, the programming address hardware automatically reads the value that is inconsistent with the written value (MISMTCH=1).

Clearance:
RegisterEFM _ FSCLR corresponds to flag clear bit write 1 and status bit reset.

2. FLASH read and write conflict interrupt EFM_COLERR:

Position:

- Read and write operations are performed on this Flash block while the Flash is busy (RDY=0). (Continuous programming mode, except for continuous writing to Flash)

Clearance:
RegisterEFM _ FSCLR corresponds to purge position 1, status bit zeroed.

3. End of operation interruptEFM _ OPTEND:

Position:

- Programming mode: Single address programming ends (OPTEND=1).
- Erase mode: sector erase, full erase ends (OPTEND=1).

Clearance:
RegisterEFM _ FSCLR corresponds to purge position 1, status bit zeroed.

7.7 One-Time Programmable Byte (OTP)

This MCU provides a maximum OTP area of 5KBytes, divided into 1 4KBytes and 16 64Bytes. The addresses are distributed as follows.

Table 7-3 OTP Address Distribution Table

Sector	OTP Block Data Address	Capacity	OTP Block Latch Address
0	0x0000 0000~0x0000 0FFF	4KB	0x0300 0A80~0x0300 0A83
1	0x0300 0C00~0x0300 0C3F	64B	0x0300 0AC0~0x0300 0AC3
2	0x0300 0C40~0x0300 0C7F	64B	0x0300 0AC4~0x0300 0AC7
3	0x0300 0C80~0x0300 0CBF	64B	0x0300 0AC8~0x0300 0ACB
4	0x0300 0CC0~0x0300 0CFF	64B	0x0300 0ACC~0x0300 0ACF
5	0x0300 0D00~0x0300 0D3F	64B	0x0300 0AD0~0x0300 0AD3
6	0x0300 0D40~0x0300 0D7F	64B	0x0300 0AD4~0x0300 0AD7
7	0x0300 0D80~0x0300 0DBF	64B	0x0300 0AD8~0x0300 0ADB
8	0x0300 0DC0~0x0300 0DFF	64B	0x0300 0ADC~0x0300 0ADF
9	0x0300 0E00~0x0300 0E3F	64B	0x0300 0AE0~0x0300 0AE3
10	0x0300 0E40~0x0300 0E7F	64B	0x0300 0AE4~0x0300 0AE7
11	0x0300 0E80~0x0300 0EBF	64B	0x0300 0AE8~0x0300 0AEB
12	0x0300 0EC0~0x0300 0EFF	64B	0x0300 0AEC~0x0300 0AEF
13	0x0300 0F00~0x0300 0F3F	64B	0x0300 0AF0~0x0300 0AF3
14	0x0300 0F40~0x0300 0F7F	64B	0x0300 0AF4~0x0300 0AF7
15	0x0300 0F80~0x0300 0FBF	64B	0x0300 0AF8~0x0300 0AFB
16	0x0300 0FC0~0x0300 0FFF	64B	0x0300 0AFC~0x0300 0AFF

Each OTP data block corresponds to a 4-Byte latch address. The latch address is used to latch the corresponding data block. When the latch address data is all 1, the corresponding OTP area data block can be programmed; when the latch address data contains 0, the corresponding OTP area data cannot be programmed. OTP data blocks 1 to 16 and latch addresses cannot be erased.

To enable the OTP function, you need to pre-program the address 0x0300 0A00~0x0300 0A03 and write 32 bits containing 0 data (it is recommended to write 32 bits of all 0). When the OTP block latch address data contains 0, the block space corresponding to the latch address cannot be programmed, erased, or fully erased again.

The sector0 of OTP shares 4KBytes physical space with Flash address 0x0000 0000~0x0000 0FFF (sector 0); if the latch address data of the space is all 1, this area is a normal Flash space and can be programmed and erased multiple times; if the corresponding latch address data contains 0, the corresponding sector is read-only, cannot be programmed and erased, and the Flash will not be able to be fully erased.

Programming the OTP latch address requires unlocking the EFM_KEY2 register (first write 0x1032 5476 to EFM_KEY2, then write 0xEFCD AB89). If an incorrect sequence of writes is made to the EFM_KEY2 register, a bus error will occur and the EFM_KEY2 register will be locked until a reset occurs.

To operate the OTP area, please follow the steps below:

1) Remove the Flash register write protection (EFM_FAPRT first writes 0x0123, then writes 0x3210);
2) Unlock EFM_KEY1 (write 0x0123 4567 to EFM_KEY1 first, then write 0xFEDC BA98);
3) Set programming mode (EFM_FWMC.PEMODE[2:0]=0b001, 0b010, 0b011);
4) Write data to the OTP address as needed;
5) Unlock EFM_KEY2 (EFM_KEY2 first writes 0x1032 5476, then writes 0xEFCD AB89);
6) Write 0 to any bit of the 32-bit OTP enable address 0x0300 0A00~0x0300 0A03 (it is recommended to write all 32 bits to 0);
7) Write 0 to any bit of the OTP latch address (it is recommended to write all 32 bits to 0).

After the chip is reset and started, the hardware will automatically load the OTP latch address data into the circuit. Therefore, if the OTP enable address 0x0300 0A00~0x0300 0A03 has been programmed with 0, step 6 needs to be omitted.

Programming, erasing and fully erasing the latched OTP data area address will generate an OTPWERR error flag.

After writing the OTP latch address, if you need to return to the EFM_KEY2 locked state, write 1 to EFM_FWMC.KEY2LOCK.

7.8 Boot Swap

If the user wants to upgrade the boot program, it is necessary to erase sector 0 (0x0000 0000~0x0000 0FFF). If an unexpected accident (such as reset, power failure) occurs during the erasing process, the entire chip may not be able to start normally. Before erasing sector 0, write the new boot program to sector 1 (0x0000 1000~0x0000 1FFF) in advance, and then program the EFM address 0x0300 2000 with data 0x5A5A5A. After the MCU is reset, the CPU starts a new boot program from sector 1, and then erases sector 0 and reprograms a new user program.

Please refer to the start exchange operation Figure 7-2.

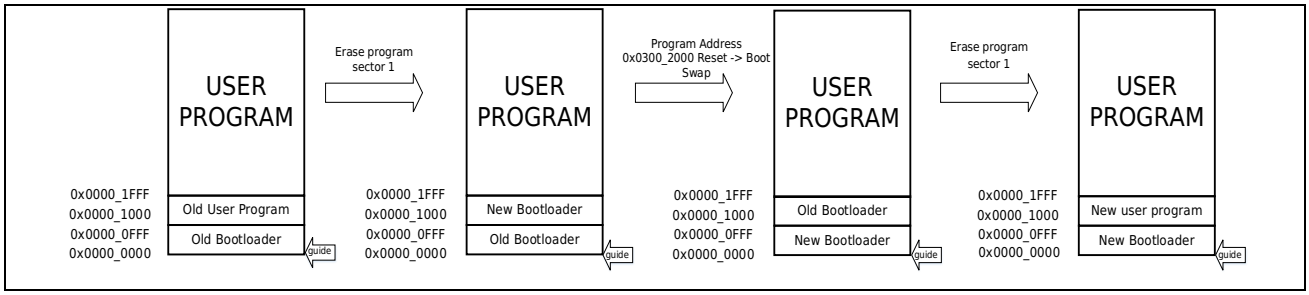


Figure 7-2 Start sector swap function 1

When the user needs to upgrade the boot program again, since the address 0x0300 2000 that stores the boot sector swap information has been programmed (the user can determine whether the boot swap function has been used by reading the FLASH address or the EFM_FSWP register), it is necessary to erase the sector at 0x0300 2000 (Note) and upgrade the boot program again.

Before erasing sector 0, write the new boot program to sector 1 in advance, and then erase 0x0300 2000 (Note). After the MCU is reset, the CPU starts a new boot program from sector 1, and then erases sector 0 and reprograms a new user program.

Note:

1. *The data security configuration area 0x0300 2040~0x0300 204B needs to be reprogrammed after erasing.*
2. *The boot swap enable address 0x0300 2000~0x0300 2003 and the data security configuration area 0x0300 2040~0x0300 204B share the same sector and can be programmed and erased, but full erase is prohibited. The data security configuration area 0x0300 2040~0x0300 204B needs to be reprogrammed after each erasure.*

The operation process is as follows Figure 7-3.

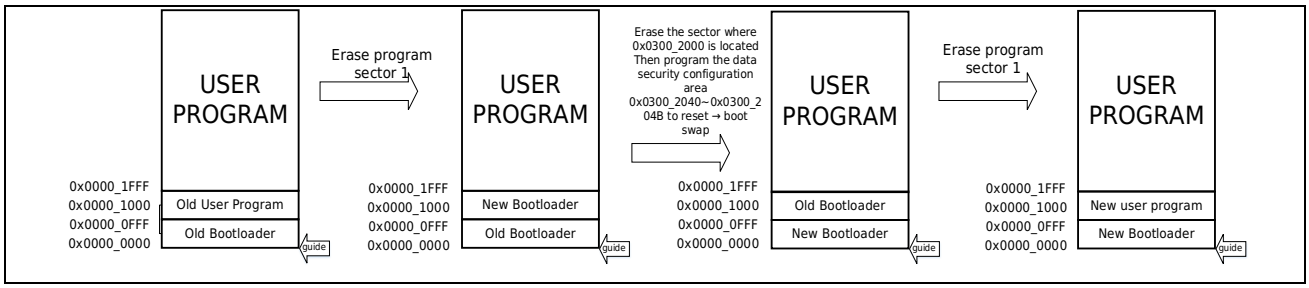


Figure 7-3 Start Swap Function 2

If the OTP of sector 0 is not latched (0x0300 0A80~0x0300 0A83 data are all 1), after the boot swap valid, the entire addresses of sectors 0 and 1 of the Flash are swapped; and the OTP latch address of sector 0 (0x0300 0A80~0x0300 0A83) cannot be programmed.

If the OTP of sector 0 is latched (0x0300 0A80~0x0300 0A83 data is not all 1), when trying to set the boot swap to be valid, the boot swap will also be invalid, and the addresses of sector 0 and sector 1 of the Flash will not be swapped.

7.9 Register Description

Table 7-4 EFM Base Address

Name	Base Address	Description
EFM	0x4001 0400	EFM Base Address

Table 7-5 EFM Register List

Register	Description	Offset Address	Bit Width	Reset Value
EFM_FAPRT	Flash Access Write Protection Register	0x0000	32	0x0000 0000
EFM_KEY1	Flash Key 1 Register	0x0004	32	0x0000 0000
EFM_KEY2	Flash Key 2 Register	0x0008	32	0x0000 0000
EFM_FSTP	Flash Stop Register	0x0014	32	0x0000 0000
EFM_FRMC	Flash Read Mode Register	0x0018	32	0x0000 0000
EFM_FWMC	Flash Erase Mode Register	0x001C	32	0x0003 0000
EFM_FSR	Flash Status Register	0x0020	32	0x0000 0100
EFM_FSCLR	Flash Status Clear Register	0x0024	32	0x0000 0000
EFM_FITE	Flash Interrupt Enable Register	0x0028	32	0x0000 0000
EFM_FSWP	Flash Boot Swap Status Register	0x002C	32	Indefinite
EFM_CHIPID	Chip-specific flag register	0x0040	32	0x5848 0334
FEM_UQID0	UNIQUE ID Register 0	0x0050	32	Indefinite
EFM_UQID1	UNIQUE ID Register 1	0x0054	32	Indefinite
EFM_UQID2	UNIQUE ID Register 2	0x0058	32	Indefinite
EFM_WLOCK	Flash Write Protection Latch Register	0x0180	32	0x0000 0000
EFM_F0NWPRT	Flash Write Protection Register	0x0190	32	0x0000 0000

7.9.1 Access Protection Register (EFM _ FAPRT)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
FAPRT[15:0]															

Bit/Bitfield	Marking	Bit Name		Function	Read and Write
b31~b16	Res	Reserved bit		Read at “0”, write at “0”	RW
b15~b0	FAPRT[15:0]	EFMregister protection	write	EFM Register Access Write Protection Register. Method: Write "16-bit data 0x0123" before "16-bit data 0x3210" on FAPRT. In the unprotected state, any data is written, and EFMregister enters the protected state again. When the EFMregister access protection is valid, the register read value is 0x0000 0000. When EFMregister access protection is invalid, the register read value is 0x0000 0001.	RW

7.9.2 Flash Key 1 Register (EFM_KEY1)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
KEY1[31:16]															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
KEY1[15:0]															

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b0	KEY1[31:0]	Key 1 Register	EFM_FWMC Write Protection Register. First write 0x0123 4567 to EFM_KEY1, then write 0xFEDC BA98 to release the write protection of EFM_FWMC. The value of this register is read as 0x0000 0000.	RW

7.9.3 Flash Key 2 Register (EFM_KEY2)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
KEY2[31:16]															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
KEY2[15:0]															

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b0	KEY2[31:0]	Key 2 Register	Write Protection Register for OTP Latch Address. First write 0x1032 5476 to EFM_KEY2, then write 0xEFCD AB89 to release the write protection of the OTP latch address. The value of this register is read as 0x0000 0000.	RW

7.9.4 Flash Stop Register (EFM_FSTP)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res															FSTP

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b1	Res	Reserved Bit	Read at “0”, write at “0”	RW
b0	FSTP	Flash Stop Mode Control	0: Flash Active State 1: Flash is in Stop Mode When the register bit is set from 1 to 0, please confirm that the FSR.RDY bit is 1 before accessing the Flash.	RW

7.9.5 Read Mode Register (EFM _ FRMC)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res												CRST	PREFETE	DCACHE	ICACHE
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res							LVM	Res				FLWT[3:0]			

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b20	Res	Reserved Bit	Read at “0”, write at “0”	RW
b19	CRST	Cache Reset	0: Cache Data Reset 1: Cache Data Reset (cache, buffer, prefetch value)	RW
b18	PREFETE	ICODE Prefetch Permission Bit	0: Disable ICODE prefetch function 1: ICODE prefetch Function Enable	RW
b17	DCACHE	DCODE Cache Permission Bits	0: Disable DCODE Cache Function 1: DCODE Cache Function Enable	RW
b16	ICACHE	ICODE Cache Permission Bits	0: Disable ICODE Cache Function 1: ICODE Cache Function Enable	RW
b15~b9	Res	Reserved Bit	Read at “0”, write at “0”	RW
b8	LVM	Low Voltage Read Mode	0: Normal Voltage Reading Mode 1: Low Voltage Read Mode In ultra-low speed operation mode, LVM needs to be set to "1"	RW
b7~b4	Res	Reserved Bit	Read at “0”, write at “0”	RW
b3~b0	FLWT[3:0]	Flash read insertion wait cycle	0000: Do not insert a read wait cycle 0001: Insert 1 read wait cycle 0010: Insert 2 read wait cycles …… 1110: Insert 14 read wait cycles 1111: Insert 15 read wait cycles	RW

7.9.6 Flash Erase Mode Register (EFM_FWMC)

Reset value: 0x0003 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res														KEY2 LOCK	KEY 1LO CK

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res							BUSH LDCTL	Res						PEMOD[2:0]	

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b18	Res	Reserved Bit	Read at “0”, write at “0”	RW
b17	KEY2LOCK	Flash Key 2 Protection Status Bit	0: EFM_KEY2 register lock release 1: EFM_KEY2 register lock This bit can only be written as 1; writing 0 invalid. When an error sequence is written to EFM_KEY2, this bit remains 1 until reset. This bit reads 0: Indicates that the EFM_KEY2 sequence is unlocked. This bit reads 1: Indicates that the EFM_KEY2 sequence is not unlocked.	RW
b16	KEY1LOCK	Flash key 1 protection status bit	0: EFM_KEY1 register lock is released. 1: EFM_KEY1 register is locked. This bit can only be written as 1; writing 0 invalid. When an error sequence is written to EFM_KEY1, this bit remains 1 until reset. This bit reads 0: Indicates that the EFM_KEY1 sequence is unlocked. This bit reads 1: Indicates that the EFM_KEY1 sequence is not unlocked.	RW
b15~b9	Res	Reserved Bit	Read at “0”, write at “0”	RW
b8	BUSHLDCTL	Flash erasing, bus control during programming	0: The bus is occupied during Flash programming or erasing. 1: The bus is released during Flash programming or erasing.	RW
b7~b3	Res	Reserved Bit	Read at “0”, write at “0”	RW
b2~b0	PEMOD[2:0]	Flash programming, erasing, full erase mode	000: Read-only mode 001: Single programming mode 010: Single-programming read-back mode 011: Continuous programming mode 100: Sector Wipe Mode 101: Single Flash full erase mode 110: Reserved value, setting prohibited 111: Read-only mode	RW

7.9.7 Flash Status Register (EFM_FSR)

Reset value: 0x0000 0100

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res							RDY	Res		COLERR	OPTEND	MISMTCH	PGSZERR	PRTWER	OTPWER

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b9	Res	Reserved Bit	Read at “0”, write at “0”	RW
b8	RDY	Flash busy/idle status	0: Flash busy 1: Flash idle state	R
b7~b6	Res	Reserved Bit	Read at “0”, write at “0”	RW
b5	COLERR	Flash read/write conflict error flag	Setting conditions: When RDY is not set, read and write operations are performed on the Flash. In continuous programming mode, the Flash write bit is not set. Clearing Conditions: Write 1 to the clear bit of EFM_FSCLR.	R
b4	OPTEND	Flash space programming, erasing, full erase operation end flag	Setting conditions: End of programming/erase/whole erase operation. Clearing Conditions: Write 1 to the clear bit of EFM_FSCLR.	R
b3	MISMTCH	Flash space single programming readback value inconsistent flag bit	Setting conditions: In single programming read-back mode, the read data after Flash programming is inconsistent with the written value. Clearing Conditions: EFM_FSCLR corresponding clear bit write 1	R
b2	PGSZERR	Flash programming address and size misalignment flag	Setting conditions: The programming address is not 4-aligned, or a non-32-bit write operation occurs. Clearing Conditions: EFM_FSCLR corresponding clear bit write 1	R
b1	PRTWERR	Flash write protection address error flag	Setting conditions: Program, erase, and fully erase write-protected sectors. Clearing Conditions: EFM_FSCLR corresponding clear bit write 1	R
b0	OTPWERR	Flash OTP erase error flag	Setting conditions: Program, erase, and full erase operations are performed on the OTP latch area. Clearing Conditions: EFM_FSCLR corresponding clear bit write 1	R

7.9.8 Flash Status Clear Register (EFM_FSCLR)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res										COLERRCLR	OPTENDCLR	MISMTCHCLR	PGSZERRCLR	PRTWERRCLR	OTPWERRCLR

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b6	Res	Reserved Bit	Read at “0”, write at “0”	RW
b5	COLERRCLR	Clear read and write conflict error flag bit	0: No purge action 1: Clear FSR.COLERR When read, this bit is always 0.	RW
b4	OPTENDCLR	End-of-purge flag	0: No purge action 1: Clear FSR.OPTEND bit When read, this bit is always 0.	RW
b3	MISMTCHCLR	Clear programming read-back mismatch flag bit	0: No purge action 1: Clear FSR.MISMTCH bit When read, this bit is always 0.	RW
b2	PGSZERRCLR	Clear programming address and size misalignment flag	0: No purge action 1: Clear FSR.PGSZERR bit When read, this bit is always 0.	RW
b1	PEPRTERRCLR	Clear the write protection error flag	0: No purge action 1: Clear the FSR.PRTWERR bit When read, this bit is always 0.	RW
b0	OTPWERRCLR	Clear the OTP erase error flag	0: No purge action 1: Clear FSR.OTPWERR bit When read, this bit is always 0.	RW

7.9.9 Flash Interrupt Enable Register (EFM_FITE)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res													COLERRITE	OPTENDITE	PEERRITE

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b3	Res	Reserved Bit	Read at “0”, write at “0”	RW
b2	COLERRITE	read-write conflict error interrupt permission	0: Read and write conflict error interrupt is not allowed 1: Read and write conflict error interrupt permission	RW
b1	OPTENDITE	End of operation interrupt enable	0: End of operation interrupt disallowed 1: End of operation interrupt enable	RW
b0	PEERRITE	Programming/Erase Error Interrupt License	0: Programming/Erase Error Interrupt Not Permitted 1: Programming/Erase Error Interrupt License	RW

7.9.10 Flash Boot Swap Status Register (EFM_FSWP)

Reset value: Indefinite

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res															FSWP

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31-b1	Res	Reserved Bit	Read at “0”, write at “0”	RW
b0	FSWP	Sector 0 and Sector 1 Address Swap Bits	0: After reset, the CPU starts from sector 0. 1: After reset, the CPU starts from sector 1. The initial value of the register is determined by the value of Flash address 0x0300 2000~0x0300 2003. When the data is 0x005A 5A5A and the OTP of sector 0 is not latched (0x0300 0A80~0x0300 0A83 data are all 1), the initial value is 1, and the rest are 0.	R

7.9.11 Chip Specific Flag Register (EFM_CHIPID)

Reset value: 0x5848 0334

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
CHIPID[31:16]															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
CHIPID[15:0]															

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b0	CHIPID[31:0]	Chip Specific Flag	CHIPID[31:16]：0x5848 CHIPID[15:0]：0x0334 Register Read-only.	R

7.9.12 UNIQUE ID Register (EFM_UQID0)

Reset value: Indefinite

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
LOT_ID[7:0]								WAFER_ID[7:0]							
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
X_LOCATION[7:0]								Y_LOCATION[7:0]							

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b24	LOT_ID[7:0]	LOT ID	Chip LOT ID [7:0]	R
b23~b16	WAFER_ID[7:0]	Wafer ID	Wafer ID	R
b15~b8	X_LOCATION[7:0]	X coordinate	X coordinate of the chip on the wafer	R
b7~b0	Y_LOCATION[7:0]	Y coordinate	Y coordinate of the chip on the wafer	R

7.9.13 UNIQUE ID Register1 (EFM_UQID1)

Reset value: Indefinite

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
LOT_ID[39:24]															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
LOT_ID[23:8]															

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b0	LOT_ID[39:8]	LOT ID	Chip LOT ID [39:8]	R

7.9.14 UNIQUE ID Register2 (EFM_UQID2)

Reset value: Indefinite

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res								LOT_ID[47:40]							

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b8	Res	Reserved Bit	Fixed to all 0	R
b7~b0	LOT_ID[47:40]	LOT ID	Chip LOT ID [47:40]	R

7.9.15 Flash Write Protection Latch Register (EFM_WLOCK)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res															WLOCK0

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b1	Res	Reserved Bit	Read at “0”, write at “0”	RW
b0	WLOCK0	F0NWPRT Register Lock Bit	0: The sector write prohibition controlled by the F0NWPRT register is not locked up 1: Sector write lockup controlled by F0NWPRT register Once this bit is set to 1, it can only be restored to 0 by resetting it.	RW

7.9.16 Flash Write Protection Register (EFM_F0NWPRT)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
F0NWPRT31	F0NWPRT30	F0NWPRT29	F0WPRT28	F0NWPRT27	F0NWPRT26	F0NWPRT25	F0NWPRT24	F0NWPRT23	F0NWPRT22	F0NWPRT21	F0NWPRT20	F0NWPRT19	F0NWPRT18	F0NWPRT17	F0NWPRT16
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
F0NWPRT15	F0NWPRT14	F0NWPRT13	F0NWPRT12	F0NWPRT11	F0NWPRT10	F0NWPRT9	F0NWPRT8	F0NWPRT7	F0NWPRT6	F0NWPRT5	F0NWPRT4	F0NWPRT3	F0NWPRT2	F0NWPRT1	F0NWPRT0

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31	F0NWPRT31	Flash sector 31 write permission bit	0: Flash sector 31 write disabled 1: Write permission for Flash sector 31 When WLOCK[0]=0, writing 0 to this bit is permitted, and writing 1 is permitted. When WLOCK[0]=1, this bit can only be 0, and writing 1 invalid.	RW
b30	F0NWPRT30	Write protection bit of Flash sector 30	0: Flash sector 30 write disabled 1: Write permission for Flash sector 30 When WLOCK[0]=0, writing 0 to this bit is permitted, and writing 1 is permitted. When WLOCK[0]=1, this bit can only be 0, and writing 1 invalid.	RW
b29~b2	…	…	…	RW
b1	F0NWPRT1	Write protection bit of Flash sector 1	0: Flash sector 1 write disabled 1: Write permission for Flash sector 1 When WLOCK[0]=0, writing 0 to this bit is permitted, and writing 1 is permitted. When WLOCK[0]=1, this bit can only be 0, and writing 1 invalid.	RW
b0	F0NWPRT0	Write protection bit of Flash sector 0	0: Flash sector 0 write disabled 1: Write permission for Flash sector 0 When WLOCK[0]=0, writing 0 to this bit is permitted, and writing 1 is permitted. When WLOCK[0]=1, this bit can only be 0, and writing 1 invalid.	RW

7.10Precautions

1) When the Flash is being erased or written, if a reset occurs, the erase operation will be forced to stop and the Flash data will not be guaranteed. The user needs to erase the address after resetting.

2) After the erase and write operation is completed, the hardware circuit will automatically reset the cache loop.

3) Repeated programming of the same address cannot ensure the correctness of the data.

4) In the continuous programming mode, the Flash analog circuit will be in a high voltage state. Long-term high voltage state will affect the Flash features. Once the continuous programming is completed, please exit the continuous programming mode immediately. In continuous programming mode, MCU is not allowed to enter low-power mode (Sleep Mode, Stop Mode, power-down mode).

5) When using read acceleration cache, make sure that the cache RAM is not powered off (PWC_PRAMLPC.PRAMPDC2=0).

6) When setting the bus release (FWMC.BUSHLDCTL=1) during programming and erasing, if you need to respond to interrupts during programming and erasing, please set the vector and subroutine to RAM.

8 Built-in SRAM (SRAM)

8.1 Overview

This product has 32KB system SRAM (SRAMH/SRAM0) and 4KB power-down mode retention SRAM (Ret SRAM).

Each SRAM can be accessed as a byte, half-word (16 bits), or full-word (32 bits). All SRAM read and write operations can be performed at the fastest CPU speed (150MHz).

Ret SRAM can provide 4KB of data retention space in Power Down mode 1/2.

SRAMH, SRAM0 and Ret SRAM have ECC (Error Checking and Correcting). The ECC check is a one-correction-two-check code, which can correct one-bit error and detect two-bit errors. The detailed definition of SRAM is shown in Table 8-1.

Table 8-1 SRAM Space Allocation

Name	Capacity	Address Range	Check Mode
SRAM0	16KB	0x2000 0000~0x2000 3FFF	ECC
SRAMH	16KB	0x1FFF C000~0x1FFF FFFF	ECC
Ret SRAM	4KB	0x200F 0000~0x200F 0FFF	ECC

Note:

– *When NMI interrupt or reset is allowed due to RAM ECC check errors, when accessing data, the RAM (SRAMH, SRAM0, Ret SRAM) used must be initialized in word units.*

All SRAMs have ECC checks with error injection capabilities. According to the settings of the SRAM error injection enable register and the SRAM error injection bit enable register, the hardware circuit inverts the SRAM read data bit by bit to achieve the purpose of error injection. By setting the SRAM error injection bit enable register, 1-bit errors and multi-bit errors can be created. Since the ECC circuit can detect 2-bit errors, errors exceeding 2 bits may not be detected.

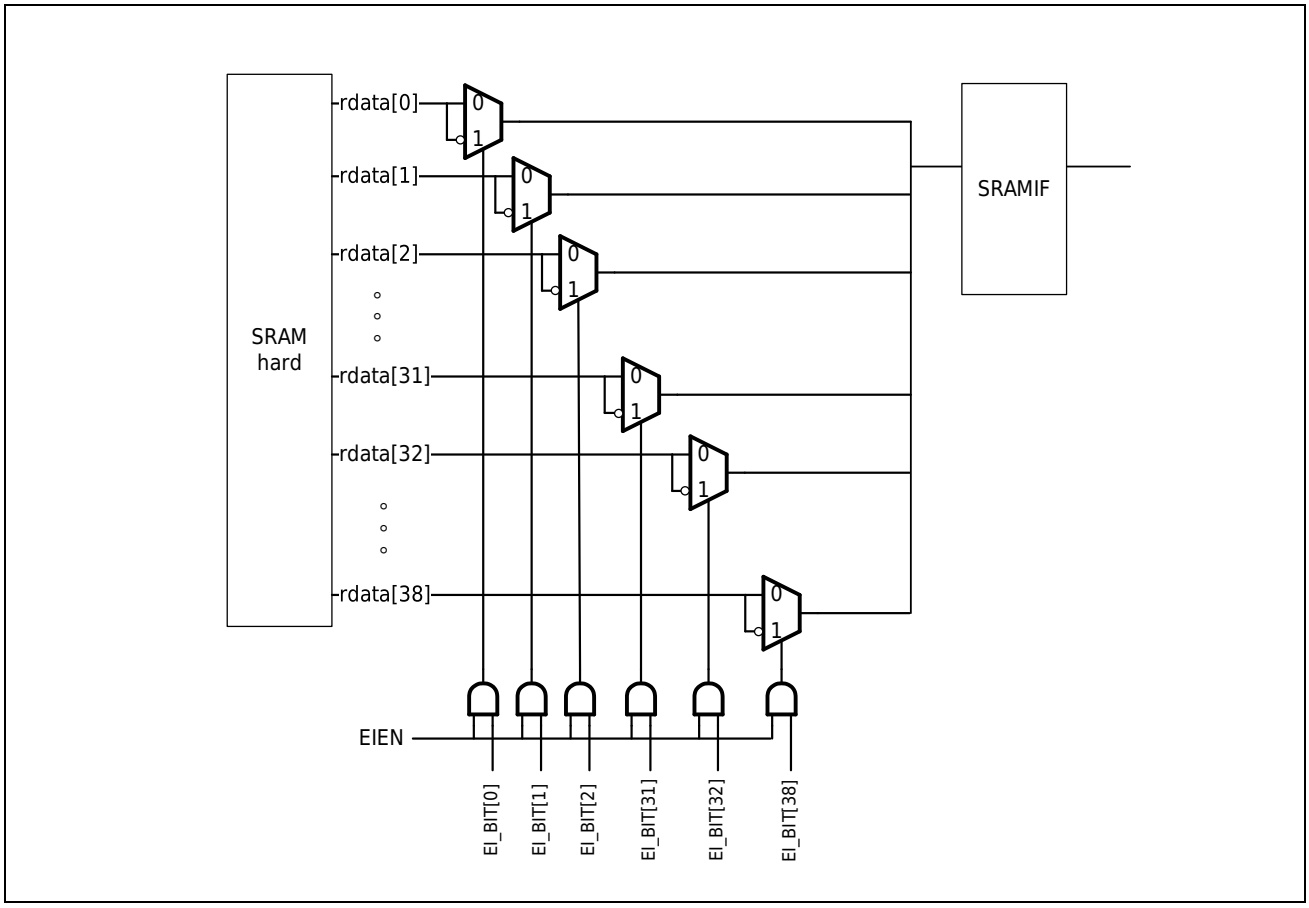


Figure 8-1 Schematic Diagram of Error Injection Function

Note:

0x2000 0000~0x2000 0002 does not support error injection and ECC checking.

8.2 Register Description

Table 8-2 SRAM Base Address

Name	Base Address	Description
SRAMC	0x4005 0800	SRAM Control Register Base Address

Table 8-3 SRAM Register List

Register	Description	Offset Address	Bit Width	Reset Value
SRAMC_CKCR	SRAM Check Control Register	0x08	32	0x0000 0000
SRAMC_CKPR	SRAM Check Control Protection Register	0x0C	32	0x0000 0000
SRAMC_ CKSR	SRAM Check Status Register	0x10	32	0x0000 0000
SRAM0_EIEN	SRAM0 Error Injection Enable Register	0x14	32	0x0000 0000
SRAM0_EIBIT0	SRAM0 Error Injection Bit Enable Register 0	0x18	32	0x0000 0000
SRAM0_EIBIT1	SRAM0 Error Injection Bit Enable Register 1	0x1C	32	0x0000 0000
SRAM0_ECCERRADDR	SRAM0 Check Error Address Register	0x20	32	0x0000 0000
SRAMH_EIEN	SRAMH Error Injection Enable Register	0x54	32	0x0000 0000
SRAMH_EIBIT0	SRAMH Error Injection Bit Enable Register 0	0x58	32	0x0000 0000
SRAMH_EIBIT1	SRAMH Error Injection Bit Enable Register 1	0x5C	32	0x0000 0000
SRAMH_ECCERRADDR	SRAMH Check Error Address Register	0x60	32	0x0000 0000
SRAMB_EIEN	Ret SRAM Error Injection Enable Register	0x64	32	0x0000 0000
SRAMB_EIBIT0	Ret SRAM Error Injection Bit Enable Register 0	0x68	32	0x0000 0000
SRAMB_EIBIT1	Ret SRAM Error Injection Bit Enable Register 1	0x6C	32	0x0000 0000
SRAMB_ECCERRADDR	Ret SRAM Check Error Address Register	0x70	32	0x0000 0000

8.2.1 SRAM Check Control Register (SRAM_CKCR)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res		HECCMOD [1:0]		BECCMOD [1:0]		ECCMOD [1:0]		Res					HECC OAD	BECC OAD	ECC OAD
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res															PYOA D

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b30	Res	Reserved Bit	Read at “0”, write at “0”	RW
b29~b28	HECCMOD[1:0]	SRAMH ECC Check Enable Bit	00: disable ECC check function 01: If 1 bit error, ECC error correction, No 1-bit error flag, no interrupt/reset; If 2 bits are wrong, ECC error detection, Generate 2-bit error flag, generate interrupt/reset. 10: If 1 bit error, ECC error correction, A 1-bit error flag is generated, no interrupt/reset is generated; If 2 bits are wrong, ECC error detection, Generate 2-bit error flag, generate interrupt/reset. 11: If 1 bit error, ECC error correction, Generate 1-bit error flag, generate interrupt/reset; If 2 bits are wrong, ECC error detection, Generate 2-bit error flag, generate interrupt/reset.	RW
b27~b26	BECCMOD[1:0]	Ret SRAM ECC Check Enable Bit	00: disable ECC check function 01: If 1 bit error, ECC error correction, No 1-bit error flag, no interrupt/reset; If 2 bits are wrong, ECC error detection, Generate 2-bit error flag, generate interrupt/reset. 10: If 1 bit error, ECC error correction, A 1-bit error flag is generated, no interrupt/reset is generated; If 2 bits are wrong, ECC error detection, Generate 2-bit error flag, generate interrupt/reset. 11: If 1 bit error, ECC error correction, Generate 1-bit error flag, generate interrupt/reset; If 2 bits are wrong, ECC error detection, Generate 2-bit error flag, generate interrupt/reset.	RW
b25~b24	ECCMOD[1:0]	ECC parity enable bit of SRAM0	00: disable ECC check function 01: If 1 bit error, ECC error correction, No 1-bit error flag, no interrupt/reset; If 2 bits are wrong, ECC error detection, Generate 2-bit error flag, generate interrupt/reset. 10: If 1 bit error, ECC error correction, A 1-bit error flag is generated, no interrupt/reset is generated; If 2 bits are wrong, ECC error detection, Generate 2-bit error flag, generate interrupt/reset. 11: If 1 bit error, ECC error correction, Generate 1-bit error flag, generate interrupt/reset; If 2 bits are wrong, ECC error detection, Generate 2-bit error flag, generate interrupt/reset.	RW
b23~b19	Res	Reserved Bit	Read at “0”, write at “0”	RW
b18	HECCOAD	Operation after SRAMH ECC Check Error	0：Non-maskable interrupt 1：Reset	RW
b17	BECCOAD	Ret SRAM ECC Check Error Operation	0：Non-maskable interrupt 1：Reset	RW
b16	ECCOAD	Operation after SRAM0 ECC Check Error	0：Non-maskable interrupt 1：Reset	RW
b15~b1	Res	Reserved Bit	Read at “0”, write at “0”	RW
b0	PYOAD	Cache RAM parity check Error Operation	0：Non-maskable interrupt	RW

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
			1：Reset	

Note:

When NMI interrupt or reset is allowed due to RAM ECC check errors, when accessing data, the RAM (SRAMH, SRAM0, Ret SRAM) used must be initialized in word units.

8.2.2 SRAM Check Control Protection Register (SRAMC_CKPR)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res								CKPRKW[6:0]							CKPRC

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b8	Res	Reserved Bit	Read at “0”, write at “0”	RW
b7~b1	CKPRKW[6:0]	Write key	When writing to the current register, you need to write "0x3B" in these bits to enable the current register	RW
b0	CKPRC	SRAM parity control register write enable	0: SRAM verification control register writing disabled 1: SRAM verification control register write enable	RW

Note:

CKPRC controls the write operation of register SRAM_CKCR. When CKPRC is 1, writing to SRAM_CKCR is allowed; when CKPRC is 0, writing to SRAM_CKCR is not allowed. When writing this bit, 0x3B must be written to CTPRKW [6:0] at the same time.

8.2.3 SRAM Check Status Register (SRAMC_CKSR)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res							CACHE_PYERR	SRAMB_2ERR	SRAMB_1ERR	SRAM0_2ERR	SRAM0_1ERR	SRAMH_2ERR	SRAMH_1ERR	Res	

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b9	Res	Reserved Bit	Read at “0”, write at “0”	RW
b8	CACHE_PYERR	CACHERAM parity check error flag	0: No parity error occurs 1: A parity error occurs	RW
b7	SRAMB_2ERR	Ret SRAM ECC 2-Bit Error Flag	0: No 2-bit ECC error occurs 1: 2-bit ECC error occurs	RW
b6	SRAMB_1ERR	Ret SRAM ECC 1-Bit Error Flag	0: No 1-bit ECC error occurs 1: 1-bit ECC error occurs	RW
b5	SRAM0_2ERR	SRAM0 occurred ECC 2-bit error flag	0: No 2-bit ECC error occurs 1: 2-bit ECC error occurs	RW
b4	SRAM0_1ERR	SRAM0 occurred ECC 1-bit error flag	0: No 1-bit ECC error occurs 1: 1-bit ECC error occurs	RW
b3	SRAMH_2ERR	SRAMH ECC 2-Bit Error Flag	0: No 2-bit ECC error occurs 1: 2-bit ECC error occurs	RW
b2	SRAMH_1ERR	SRAMH ECC 1-Bit Error Flag	0: No 1-bit ECC error occurs 1: 1-bit ECC error occurs	RW
b1~b0	Res	Reserved Bit	Read at “0”, write at “0”	RW

Note:

Writing 1 to the flag bit in this register clears it to 0.

8.2.4 SRAM0 Error Injection Enable Register (SRAM0_EIEN)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res															EIEN

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b1	Res	Reserved Bit	Read at “0”, write at “0”	RW
b0	EIEN	Fault injection enable	0: Fault injection disabled 1: Error injection invalid	RW

8.2.5 SRAM0 Error Injection Bit Enable Register 0 (SRAM0_EIBIT0)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
EIBIT[31:16]															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
EIBIT[15:0]															

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b0	EIBIT[31:0]	Error injection bit enable	Controls whether bits 0 to 31 of SRAM0 output data are inverted 0: Not negated 1: Negate	RW

8.2.6 SRAM0 Error Injection Bit Enable Register 1 (SRAM0_EIBIT1)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res									EIBIT[38:32]						

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b7	Res	Reserved Bit	Read at “0”, write at “0”	RW
b6~b0	EIBIT[38:32]	Error injection bit enable	Controls whether bits 32 to 38 of SRAM0 output data are inverted 0: Not negated 1: Negate	RW

Note:

Bits 32 to 38 of the SRAM0 output data are ECC bits.

8.2.7 SRAM0 Check Error Address Register (SRAM0_ECCERRADDR)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res	ECCERRADDR [14:0]														

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b15	Res	Reserved Bit	Read at “0”, write at “0”	RW
b14~b0	ECCERRADDR [14:0]	Checksum error address	Record the access address (offset address) when a 1-bit or 2-bit ECC error occurs in SRAM0	R

8.2.8 SRAMH Error Injection Enable Register (SRAMH_EIEN)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res															EIEN

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b1	Res	Reserved Bit	Read at “0”, write at “0”	RW
b0	EIEN	Fault injection enable	0: Fault injection disabled 1: Error injection invalid	RW

8.2.9 SRAMH Error Injection Bit Enable Register 0 (SRAMH_EIBIT0)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
EIBIT[31:16]															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
EIBIT[15:0]															

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b0	EIBIT[31:0]	Error injection bit enable	Controls whether bits 0 to 31 of the SRAMH output data are inverted 0: Not negated 1: Negate	RW

8.2.10 SRAMH Error Injection Bit Enable Register 1 (SRAMH_EIBIT1)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res									EIBIT[38:32]						

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b7	Res	Reserved Bit	Read at “0”, write at “0”	RW
b6~b0	EIBIT[38:32]	Error injection bit enable	Controls whether bits 32 to 38 of the SRAMH output data are inverted 0: Not negated 1: Negate	RW

Note:

Bits 32 to 38 of the SRAMH output data are ECC bits.

8.2.11 SRAMH Check Error Address Register (SRAMH_ECCERRADDR)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res	ECCERRADDR [14:0]														

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b15	Res	Reserved Bit	Read at “0”, write at “0”	RW
b14~b0	ECCERRADDR [14:0]	Checksum error address	Record the access address (offset address) when a 1-bit or 2-bit ECC error occurs in SRAMH	R

8.2.12 Ret SRAM Error Injection Enable Register (SRAMB_EIEN)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Rev															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Rev															EIEN

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b1	Res	Reserved Bit	Read at “0”, write at “0”	RW
b0	EIEN	Fault injection enable	0: Fault injection disabled 1: Error injection invalid	RW

8.2.13 Ret SRAM Error Injection Bit Enable Register 0 (SRAMB_EIBIT0)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
EIBIT[31:16]															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
EIBIT[15:0]															

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b0	EIBIT[31:0]	Error injection bit enable	Controls whether bits 0 to 31 of the Ret SRAM output data are inverted 0: Not negated 1: Negate	RW

8.2.14 Ret SRAM Error Injection Bit Enable Register 1 (SRAMB_EIBIT1)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res									EIBIT[38:32]						

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b7	Res	Reserved Bit	Read at “0”, write at “0”	RW
b6~b0	EIBIT[38:32]	Error injection bit enable	Controls whether bits 32 to 38 of the Ret SRAM output data are inverted 0: Not negated 1: Negate	RW

Note:

The 32~38 bits of the Ret SRAM output data are ECC bits.

8.2.15 Ret SRAM Check Error Address Register (SRAMB_ECCERRADDR)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res				ECCERRADDR [11:0]											

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b12	Res	Reserved Bit	Read at “0”, write at “0”	RW
b11~b0	ECCERRADDR [11:0]	Checksum error address	Record the access address (offset address) when a 1-bit or 2-bit ECC error occurs in Ret SRAM	R

8.3 Precautions

When the CPU or DMA accesses any address of SRAM0 at the same time, the DMA or CPU should not continuously access addresses 0x2000 0000 ~ 0x2000 0002 of SRAM0.

9 General-Purpose I/Os (GPIO)

Some of the abbreviations used in this chapter are:

- Px (x=A~D, F) represents a group of ports, such as PA represents the 16 I/O ports of the group PA0~PA15.
- Pxy (x=A~D, F, y=0~15, the same below) represents a single I/O port. For example, the PB10 port represents the 11th I/O in the PB group.
- General Purpose Input Output (GPIO).
- NOD (Nmos Open Drain) NMOS open drain output mode.

9.1 Introduction

Main Features:

- Each port group has 16 I/O Pins, which may be less than 16 depending on actual configuration
- Support pull-up and pull-down input
- Support push-pull, open-drain output mode
- Supports high, medium and low drive modes
- Supports free switching between CMOS/Schmitt input modes
- Support for external interrupt input
- Support I/O pin peripheral function multiplexing, one I/O pin can have up to 51 optional multiplexing functions
- Individual I/O pins can be programmed independently
- Each I/O pin can select 2 functions to be valid at the same time (does not support 2 output functions to be valid at the same time)

9.2 Functional Description

9.2.1 GPIO Port Basic Structure

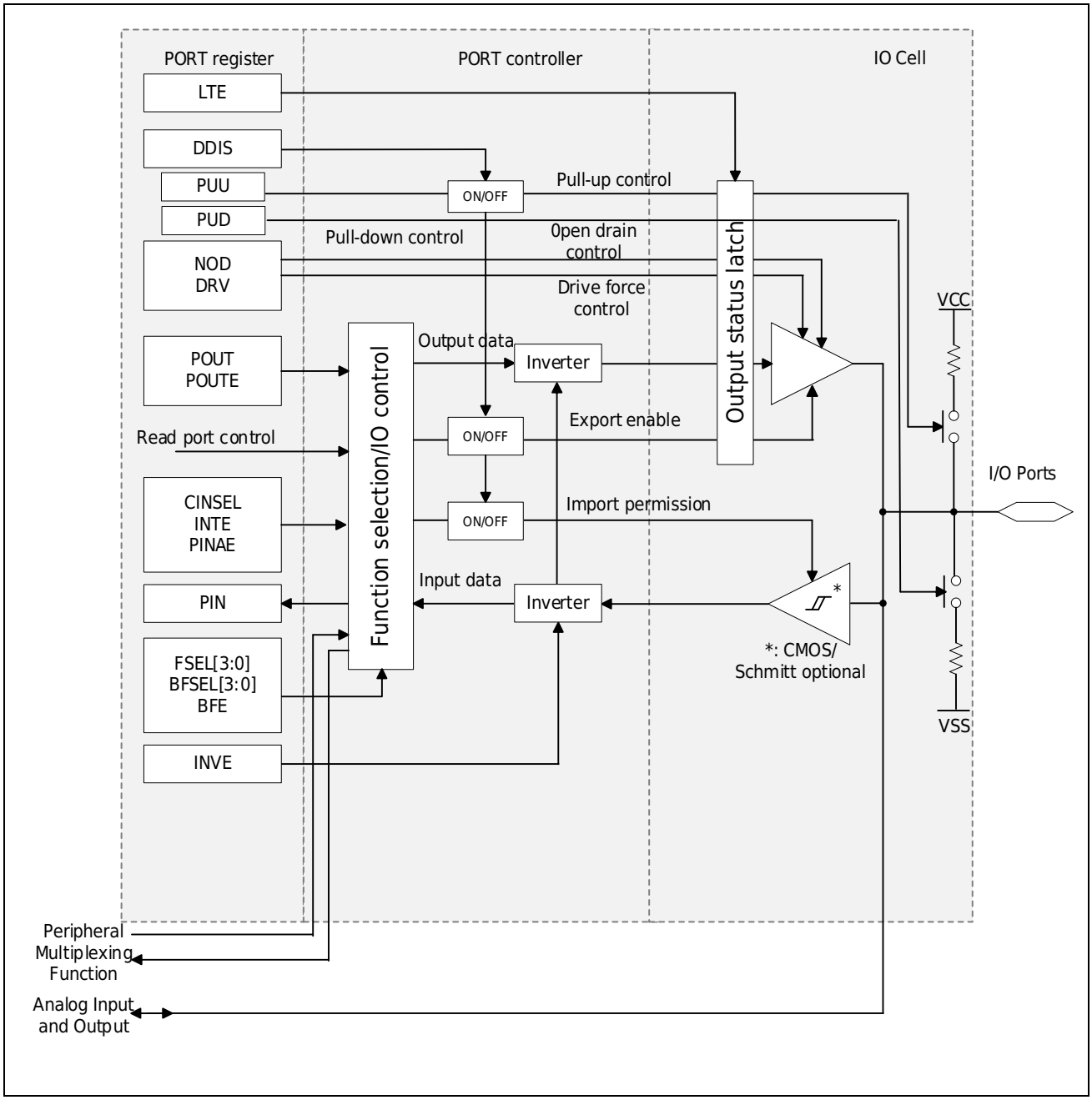


Figure 9-1 Port Basic Structure Diagram

For detailed GPIO port number, 5V durability, and drive capability configuration, please refer to the **Pin Configuration and Function Section in the Data Sheet.**

9.2.2 General purpose Input/Output Function

General Purpose Input Function GPI

Each I/O has a general-purpose input GPI function, and when the digital function disable bit PCRxy.DDIS is 0, the GPI function is always valid, regardless of the setting value of FSEL[5:0] in the function selection register PFSRxy. The status of the current port can be obtained by accessing port input data registerPIDRx. The corresponding single I/O port status can also be queried through the PIN bit of the port control register PCRxy, 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. It will only be turned on when reading the PIDRx and PCRxy registers. If needed, you can also set the register PCRxy.PINAE to1 to keep the I/O input MOS in the open state.

When the system runs at a high-speed clock and PCRxy.PINAE is set to 0, the input status value may not be read correctly in a single cycle due to the delay of the I/O input. At this time, it is necessary to set the register PCCR.RDWT[2:0] and insert several wait cycles. For details, refer to the description of the register PCCR.

General Output Function GPO

Except for input-specific ports, other I/O ports have general purpose output GPO capabilities. The GPO function can be enabled by setting the port function selection register PFSRxy.FSEL[5:0] to 0x0.

When the GPO function is active, the output value can be controlled by setting the generic output permission registerPOERx to allow or disable the output of I/O and the generic output data registerPODRx. The output values of I/O can also be controlled using the following three registers: Output data zeroed registerPORRx, output data set registerPOSRx, and output data reversed registerPOTRx. Writing 1 to the corresponding bits in the register can make the corresponding I/O output 0, 1, invert. I/O output status does not change at write 0.

The registers above are operated by 16 PORT groups. For easy control of a single I/O, I/O output can also be allowed or disabled by setting PCRxy.POUTE, the PCRxy.POUTEregister bit is equivalent to POERx.POUTEy. The output value of I/O can be controlled by setting PCRxy.POUT, where the PCRxy.POUTregister bit is equivalent to PODRx.POUTy. PCRxy is suitable for controlling a single PORT and POERx/PODRx is suitable for controlling an 16-bit set of PORTs.

After the system is reset, except for the JTAG multiplexing ports PA13, PA14, PA15, PB3, PB4, and the sub-oscillator multiplexing ports PC14 and PC15, the initial functions of other ports are GPO (FSEL[5:0]=0x0) and are in high-impedance state (output disabled POUTE=0).

9.2.3 Peripheral Functions

Through the FSEL[5:0] of the function selection register PFSRxy, each port can configure up to 51 functions. These include the generic output GPO function corresponding to FSEL [5: 0] = 0x0. For

the specific configuration functions of each port, please refer to the pin function table in the Data Manual.

JTAG/SWD debug function, use register PSPCR to select. PSPCR.SPFE[z] (z=0~4), when z=0~4 is 1, the PFSRxy.FSEL[5:0] register bit of the corresponding port is invalid, that is, the priority of SPFE is higher than that of FSEL. The initial value of the PSPCR register is 0x1F, and the JTAG/SWD function is valid. If you want to set these ports as functions other than JTAG/SWD, you need to write 0 to the corresponding SPFE[z] bit first.

9.2.4 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[5:0], then select the second function by setting the common control register PCCR.BFSEL[5:0], and set PFSRxy.BFE to 1. For example: set PFSRxy.FSEL[5:0]=0x2, PCCR.BFSEL[5:0]=0x5, PFSRxy.BFE=0x1, then function 2 and function 5 on Pxy will be valid at the same time. It is forbidden to enable 2 output functions on the same port at the same time.

9.2.5 Event Port Input and Output Functions

Supports 4 groups of Event Ports, of which Event Port (0~3) has 16 ports in each group and Event Port4 has only one port. Event Port1 includes EVNTP100~EVNTP115, Event Port2 includes EVNTP200~EVNTP215, and so on. The EVNTPmn (m=1~4, n=0~15) port can be used as a trigger source to generate events based on the port input to trigger other peripherals (such as TIMER, ADC, DMA, etc.) to start specific actions. 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 EVNT_TRGSELm 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 PEVNTIDRx when the selected event occurs.

When using the Event Port function, it is necessary to first set the automatic operation system AOS function enable bit of thefunction clock control 0 register (PWC_FCG0) to valid.

9.2.6 External Interrupt EIRQ Input Function

Except for the PF3/MD pin, each I/O port has an external interrupt input function. When the PCRxy.INTE bit is set to 1, this I/O will be enabled as the external interrupt source EIRQy input (for example: PA0 corresponds to EIRQ0, PA2 corresponds to EIRQ2). Each EIRQy can be configured with more than one I/O. Please do not allow multiple I/O inputs simultaneously for each EIRQy. The EIRQy input function and the peripheral functions selected by PFSRxy.FSEL (including GPIO) can be valid at the same time.

When I/O port is used as external interruptEIRQ, I/O port needs to combine with interrupt controller INTC, set filter, interrupt trigger edge, interrupt number and so on. Please refer to [Interrupt controller (INTC)] for details.

9.2.7 Analog Function

Some I/O ports have analog input and output functions (including primary and secondary oscillators). When used as an analog function, please write 1 to the register PCRxy.DDIS to disable the digital function of the current port.

9.2.8 General-Purpose Control

1. Pull-up and Pull-down Resistors

Each I/O port has an internal pull-up resistor. You can set the registerPCRxy.PUU bit to allow this feature, which is in a weak 1 state when I/O ports do not have inputs. Pull-up is automatically invalid when the I/O port is in the output state.

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 invalid.

Each I/O port has an internal pull-down resistor. You can set the register PCRxy.PUD bit to allow this feature, which is in a weak 0 state when I/O ports do not have inputs. Pull-down is automatically invalid when the I/O port is in the output state.

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 invalid.

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

2. Drive Capability Control

Each I/O port has three levels of high, medium and low drive capability adjustable, and the register PCRxy.DRV[1:0] can be set as required.

This feature is only valid if the port is in the output state.

3. 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 valid.

4. 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.

The general purpose control features described above, if not specified, are independent of the port-specific selected features, FSEL [5: 0] settings.

9.3 Register Description

Table 9-1 GPIO Base Address

Name	Base Address	Description
GPIO	0x4005 3800	GPIO Base Address

Table 9-2 GPIO Register

Register	Description	Offset Address	Bit Width	Reset Value
PIDRx	General purpose input data register	0x00+0x10*n (1)	16/32	0xXXXX
PODRx	General purpose output data register	0x04+0x10*n	16/32	0x0000
POERx	General purpose output register	0x06+0x10*n	16/32	0x0000
POSRx	General purpose output set register	0x08+0x10*n	16/32	0x0000
PORRx	General purpose output reset register	0x0A+0x10*n	16/32	0x0000
POTRx	General purpose output flip register	0x0C+0x10*n	16/32	0x0000
PHRPWM	PHRPWM Control Register	0x3F0	16/32	0x0000 0000
PSPCR	Special Control Registers	0x3F4	16/32	0x001F
PCCR	Common control	0x3F8	16/32	0x1000
PWPR	Write protection register	0x3FC	16/32	0x0000
PCRxy	General purpose control register	0x400+0x40*n+0x4*y (1)	16/32	0x0X00 (2)
PFSRxy	Function Select Register	0x402+0x40*n+0x4*y	16/32	0x0000

Note:
(1) In the address calculation formula, x=A~D, F corresponds to n=0~3, 5; y=0~15.
(2) The reset value of the 32KHz sub-oscillator multiplexing ports PCRC14 and PCRC15 is 0x8100.

Table 9-3 Event Port Base Address

Name	Base Address	Description
Event Port	0x4001 0800	Event Port Base Address

Table 9-4 Event Port Register

Register	Description	Offset Address	Bit Width	Reset Value
PEVNTDIRRm	Event port direction selection register	0x100+0x1C* (m-1)	32	0x0000 0000
PEVNTIDRm	Event Port input data register	0x104+0x1C* (m-1)	32	0x0000 0000
PEVNTODRm	Event port output data register	0x108+0x1C* (m-1)	32	0x0000 0000
PEVNTORRm	Event port output data reset register	0x10C+0x1C* (m-1)	32	0x0000 0000
PEVNTOSRm	Event Port output data setting register	0x110+0x1C* (m-1)	32	0x0000 0000
PEVNTRISRm	Event Port rising edge input permission register	0x114+0x1C* (m-1)	32	0x0000 0000
PEVNTFALRm	Event Port falling edge input permission register	0x118+0x1C* (m-1)	32	0x0000 0000
PEVNTNFCR	Event Port input filter control register	0x170	32	0x0000 0000

Note: m=1~4

9.3.1 General Purpose Input Data Register (PIDRx)

Reset value: 0xXXXX

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
PIN15	PIN14	PIN13	PIN12	PIN11	PIN10	PIN9	PIN8	PIN7	PIN6	PIN5	PIN4	PIN3	PIN2	PIN1	PIN0

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b15~b0	PINy y=0~15	Input status	Each I/O port corresponds to a PIN bit 0: I/O port input status is low 1: I/O port input status is high	R

Note:

This register is read-only and the write is invalid. When the digital function is not disabled and DDIS=0, the input status of the port can be obtained by reading this register, regardless of the setting value of PFSRxy.FSEL[5:0] of the function selection register. There is no variable read value for the corresponding bit of the port. DDIS=1 When the digital function of the port is disabled, the corresponding PIN bit readout value is fixed value 0x1 because the I/O input MOS is off.

9.3.2 General PurposeOutput Data Register (PODRx)

Reset value: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
POUT15	POUT14	POUT13	POUT12	POUT11	POUT10	POUT9	POUT8	POUT7	POUT6	POUT5	POUT4	POUT3	POUT2	POUT1	POUT0

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b15~b0	POUTy y=0~15	Output data	Each I/O port has a POUT bit 0: Output low level 1: 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.

9.3.3 General Purpose Output License Register (POERx)

Reset value: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
POUTE15	POUTE14	POUTE13	POUTE12	POUTE11	POUTE10	POUTE9	POUTE8	POUTE7	POUTE6	POUTE5	POUTE4	POUTE3	POUTE2	POUTE1	POUTE0

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b15~b0	POUTEy y=0~15	Export enable	Each I/O port has a POUTE bit 0: Output disable 1: Export enable	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. Do not write 1 without port bits.

9.3.4 General Purpose Output Set Register (POSRx)

Reset value: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
POS15	POS14	POS13	POS12	POS11	POS10	POS9	POS8	POS7	POS6	POS5	POS4	POS3	POS2	POS1	POS0

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b15~b0	POSy y=0~15	Output height	Each I/O port has a POS bit 0: No change to PODRx.POUT 1: Set to 1 corresponding to PODRx.POUT	W

Note:

The read value for this register is always 0x0000. During 32bit access, when POR[y] and POS[y] of the same I/O write 1 at the same time, POR[y] has a higher priority, that is, the corresponding POUT[y] is cleared.

9.3.5 General Purpose Output Reset Register (PORRx)

Reset value: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
POR15	POR14	POR13	POR12	POR11	POR10	POR9	POR8	POR7	POR6	POR5	POR4	POR3	POR2	POR1	POR0

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b15~b0	PORy y=0~15	Output low	Each I/O port has a POR bit 0: No change to PODRx.POUT 1: corresponding to clear PODRx.POUT	W

Note:

The read value for this register is always 0x0000.

9.3.6 General Purpose Output Flip Register (POTRx)

Reset value: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
POT15	POT14	POT13	POT12	POT11	POT10	POT9	POT8	POT7	POT6	POT5	POT4	POT3	POT2	POT1	POT0

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b15~b0	POTy y=0~15	Output Inversion	Each I/O port has a POT bit 0: No change to PODRx.POUT 1: Reverse PODRx.POUT	W

Note:

The read value for this register is always 0x0000.

9.3.7 HRPWM Control Register (HRPWM)

Reset value: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res				HREX[11:0]											
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res				HRSEL[11:0]											

Bit	Marking	Bit Name	Function	Read and Write
b31~28	Res	Reserved Bit	Read at 0, write at 0	RW
b27	HREX[11]	High-precision PWM output switching	0: HRPWM_6_PWM outputs on PC9 are not swapped 1: HRPWM_6_PWM outputs on PC9 are swapped to symmetrical channels	RW
b26	HREX[10]	High-precision PWM output switching	0: HRPWM_6_PWM outputs on PC8 are not swapped 1: HRPWM_6_PWM outputs on PC8 are swapped to symmetrical channels	RW
b25	HREX[9]	High-precision PWM output switching	0: HRPWM_5_PWM outputs on PB15 are not swapped 1: HRPWM_5_PWM outputs on PB15 are swapped to symmetrical channels	RW
b24	HREX[8]	High-precision PWM output switching	0: HRPWM_5_PWM outputs on PB14 are not swapped 1: HRPWM_5_PWM outputs on PB14 are swapped to symmetrical channels	RW
b23	HREX[7]	High-precision PWM output switching	0: HRPWM_4_PWM outputs on PB13 are not swapped 1: HRPWM_4_PWM outputs on PB13 are swapped to symmetrical channels	RW
b22	HREX[6]	High-precision PWM output switching	0: HRPWM_4_PWM outputs on PB12 are not swapped 1: HRPWM_4_PWM outputs on PB12 are swapped to symmetrical channels	RW
b21	HREX[5]	High-precision PWM output switching	0: HRPWM_3_PWM outputs on PA11 are not swapped 1: HRPWM_3_PWM outputs on PA11 are swapped to symmetrical channels	RW
b20	HREX[4]	High-precision PWM output switching	0: HRPWM_3_PWM outputs on PA10 are not swapped 1: HRPWM_3_PWM outputs on PA10 are swapped to symmetrical channels	RW
b19	HREX[3]	High-precision PWM output switching	0: HRPWM_2_PWM outputs on PA9 are not swapped 1: HRPWM_2_PWM outputs on PA9 are swapped to symmetrical channels	RW
b18	HREX[2]	High-precision PWM output switching	0: HRPWM_2_PWM outputs on PA8 are not swapped 1: HRPWM_2_PWM outputs on PA8 are swapped to symmetrical channels	RW
b17	HREX[1]	High-precision PWM output switching	0: HRPWM_1_PWM outputs on PC7 are not swapped 1: HRPWM_1_PWM outputs on PC7 are swapped to symmetrical channels	RW
b16	HREX[0]	High-precision PWM output switching	0: HRPWM_1_PWM outputs on PC6 are not swapped 1: HRPWM_1_PWM outputs on PC6 are swapped to symmetrical channels	RW
b15~12	Res	Reserved Bit	Read at 0, write at 0	RW
b11	HRSEL[11]	High-precision PWM function selection	0: HRPWM_6_PWMB output function on PC9 is not selected 1: HRPWM_6_PWMB output function on PC9 is selected	RW
b10	HRSEL[10]	High-precision PWM function selection	0: HRPWM_6_PWMA output function on PC8 is not selected 1: HRPWM_6_PWMA output function on PC8 is selected	RW
b9	HRSEL[9]	High-precision PWM function selection	0: HRPWM_5_PWMB output function on PB15 is not selected 1: HRPWM_5_PWMB output function on PB15 is selected	RW
b8	HRSEL[8]	High-precision PWM function selection	0: HRPWM_5_PWMA output function on PB14 is not selected 1: HRPWM_5_PWMA output function on PB14 is selected	RW
b7	HRSEL[7]	High-precision PWM function selection	0: HRPWM_4_PWMB output function on PB13 is not selected 1: HRPWM_4_PWMB output function on PB13 is selected	RW
b6	HRSEL[6]	High-precision PWM function selection	0: HRPWM_4_PWMA output function on PB12 is not selected 1: HRPWM_4_PWMA output function on PB12 is selected	RW
b5	HRSEL[5]	High-precision PWM function selection	0: HRPWM_3_PWMB output function on PA11 is not selected 1: HRPWM_3_PWMB output function on PA11 is selected	RW
b4	HRSEL[4]	High-precision PWM function selection	0: HRPWM_3_PWMA output function on PA10 is not selected 1: HRPWM_3_PWMA output function on PA10 is selected	RW
b3	HRSEL[3]	High-precision PWM function selection	0: HRPWM_2_PWMB output function on PA9 is not selected 1: HRPWM_2_PWMB output function on PA9 is selected	RW
b2	HRSEL[2]	High-precision PWM function selection	0: HRPWM_2_PWMA output function on PA8 is not selected 1: HRPWM_2_PWMA output function on PA8 is selected	RW

Bit	Marking	Bit Name	Function	Read and Write
b1	HRSEL[1]	High-precision PWM function selection	0: HRPWM_1_PWMB output function on PC7 is not selected 1: HRPWM_1_PWMB output function on PC7 is selected	RW
b0	HRSEL[0]	High-precision PWM function selection	0: HRPWM_1_PWMA output function on PC6 is not selected 1: HRPWM_1_PWMA output function on PC6 is selected	RW

9.3.8 Special Control Register (PSPCR)

Reset value: 0x001F

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res											SPFE[4:0]				

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b15~b5	Res	Reserved Bit	Read at 0, write at 0	RW
b4	SPFE[4]	Special function selection	0: Invalid NJTRST feature 1: The NJTRST function is valid	RW
b3	SPFE[3]	Special function selection	0: JTDI function invalid 1: JTDI function valid	RW
b2	SPFE[2]	Special function selection	0: JTDO_TRACESWO function invalid 1: JTDO_TRACESWO function valid	RW
b1	SPFE[1]	Special function selection	0: JTMS_SWDIO function invalid 1: JTMS_SWDIO function valid	RW
b0	SPFE[0]	Special function selection	0: JTCK_SWCLK function invalid 1: JTCK_SWCLK function valid	RW

Note:

The SPFE[4:0] function selection bits have higher priority than the PFSRxy.FSEL[5:0] function selection bits.

9.3.9 Common Control Register (PCCR)

Reset value: 0x1000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res	RDWT[2:0]			Res						BFSEL[5:0]					

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b15	Res	Reserved Bit	Read at 0, write at 0	RW
b14~b12	RDWT[2:0]	Read Port Waiting	Set the number of waiting cycles inserted when reading registers PIDRx and PCRxy. The default value is 0b001. Operating frequency / Waiting period / Set value ~50MHz / 0 cycle / 0~7 50~100MHz / 1 cycle / 1~7 100~150MHz / 2 cycle / 2~7	RW
b11~b6	Res	Reserved Bit	Read at 0, write at 0	RW
b5~b0	BFSEL[5:0]	Secondary function selection	For the function configuration of each port, please refer to the **Pin Function Table in the Data Manual**	RW

9.3.10 Write Protection Register (PWPR)

Reset value: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
WP[7:0]								Res							WE

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b15~b8	WP[7:0]	Write protection code	Read at 0x00 When b15 ~ b8 write value is 0xA5, b0 write WE When writing a value other than 0xA5, WE automatically clears	W
b7~b1	Res	Reserved Bit	Read at 0, write at 0	RW
b0	WE	Write permission	0: PSPCR, PCCR, PCRxy, PFSRxy registers write disabled 1: PSPCR, PCCR, PCRxy, PFSRxy register write permission	RW

9.3.11 General Purpose Control Register (PCRxy)

Reset Value: 0x0X00

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
DDIS	LTE	PINAE	INTE	Res	CINSEL	INVE	PIN	PUD	PUU	DRV[1:0]		Res	NOD	POUTE	POUT

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b15	DDIS	Digital function disabled	0: digital function is valid 1: digital function disabled	RW
b14	LTE	Output state latch	0: Invalid output latch 1: Output latch effective	RW
b13	PINAE	Input normally open	0: Input MOS normally open invalid 1: Input MOS normally open	RW
b12	INTE	External interrupt enable	0: External interrupt input disable 1: External interrupt input enable	RW
b11	Res	Reserved Bit	Read at 0, write at 0	RW
b10	CINSEL	Input mode selection	0: Schmitt input 1: CMOS input	RW
b9	INVE	Inverse enable	0: Input/output data not inverted 1: Inverted input/output data	RW
b8	PIN	Input status	0: I/O port input status is low 1: I/O port input status is high The same function as POUTy in register PIDRx	R
b7	PUD	Pull-down permission	0: Internal pull-down resistor invalid 1: Internal pull-down resistor valid	RW
b6	PUU	Pull-up enable	0: Internal pull-up resistor invalid 1: Pullup resistance valid	RW
b5~b4	DRV[1:0]	Drive mode selection	00: Low driving force mode 01: Medium driving force mode 10, 11: High drive mode	RW
b3	Res	Reserved Bit	Read at 0, write at 0	RW
b2	NOD	NMOS Opening Leakage	0: Normal CMOS Output Mode 1: NMOS open-drain output	RW
b1	POUTE	Export enable	0: Output disable 1: Export enable The same function as POUTEy in register POERx	RW
b0	POUT	Output data	0: Output low level 1: Output high level Consistent with the POUT [y] function in registerPODRx	RW

Note:

1. *When DDIS is set to 1, all digital functions of the corresponding port are forcibly invalid, including general-purpose input and output, peripheral digital input and output, pull-up functions, and external interrupt input functions. When the port is used as an analog input, please set the DDIS bit to 1.*
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.*
5. *The reset value of the general purpose control register PCR of the following ports is not 0x0X00, please pay attention. XTAL32_IN, XTAL32_OUT multiplex port PC15, the reset value of PCRC15 and PCRC14 registers of PC14 is 0x8100.*

9.3.12 Function Select Register (PFSRxy)

Reset value: 0x0000 (1)

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res							BFE	Res		FSEL[5:0]					

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b15~b9	Res	Reserved Bit	Read at 0, write at 0	RW
b8	BFE	Secondary feature license	Control whether the secondary function selected by PCCR.BFSEL[5:0] is valid 0: Minor function disabled 1: The auxiliary function is valid	RW
b7~b6	Res	Reserved Bit	Read at 0, write at 0	RW
b5~b0	FSEL[5:0]	Functional selection	For the function configuration of each port, please refer to the **Pin Function Table in the Data Manual**	RW

Note:

1. *Each I/O port can select one of several features configured on the port via FSEL [5: 0]. Refer to the **Pin Function Table in the Data Sheet**. Setting FSEL[5:0] to 0 means selecting Func0, setting it to 1 means selecting Func1, and so on. Setting it to 15 means selecting Func15. Among them, Func0 Corresponds to General Purpose Output Function GPO.*
2. *The initial state of PA13, PA14, PA15, PB3, and PB4 ports is that the JTAG/SWD function is valid after reset. When configuring the FSEL[5:0] selection function, it is necessary to write 0 to the corresponding bit of the register PSPCR to invalidate the JTAG/SWD function.*
3. *After resetting ports PC14 and PC15, their initial state is low-speed oscillation port, with digital functions disabled. When selecting digital functions, the XTAL32STP of the corresponding register CMU_XTAL32CR should first be set to 1 to turn off the external low-speed oscillator, and the DDIS bit of the PCRxy register should be written as 0 to enable the digital functions.*

9.3.13 Event Port Direction Selection Register (PEVNTDIRRm)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
PDIR[15:0]															

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b16	Res	Reserved Bit	Read at 0, write at 0	RW
b15~b0	PDIR15:0]	Direction selection	0: Event Port is an input function 1: Event Port is an output function	RW

9.3.14 Event Port Input Data Register (PEVNTIDRm)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
POUT[15:0]															

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b16	Res	Reserved Bit	Read at 0, write at 0	RW
b15~b0	POUT[15:0]	Port input state	0: Event Port input state is low level when the event is triggered 1: The Event Port input state is high when the event is triggered	R

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.

9.3.15 Event Port Output Data Register (PEVNTODRm)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
POUT[15:0]															

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b16	Res	Reserved Bit	Read at 0, write at 0	RW
b15~b0	POUT[15:0]	Port output value	0: Event Port output low level 1: 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 flipped, and output to the EVNTPmn port at the same time.

9.3.16 Event Port Output Data Reset Register (PEVNTORRm)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
POR[15:0]															

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b16	Res	Reserved Bit	Read at 0, write at 0	RW
b15~b0	POR[15:0]	Output value reset	0: The corresponding PEVNTODRm.POUT does not change when the event is triggered 1: The corresponding PEVNTODRm.POUT is reset when the event is triggered	RW

Note:

When both PEVNTORRm.POR and PEVNTm.POS are set to 1, the corresponding PEVNTODRm.POUT is flipped when the event is triggered.

9.3.17 Event Port Output Data Set Register (PEVNTOSRm)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
POS[15:0]															

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b16	Res	Reserved Bit	Read at 0, write at 0	RW
b15~b0	POS[15:0]	output value set	0: The corresponding PEVNTODRm.POUT does not change when the event is triggered 1: The corresponding PEVNTODRm.POUT is set when the event is triggered	RW

Note:

When both PEVNTORRm.POR and PEVNTm.POS are set to 1, the corresponding PEVNTODRm.POUT is flipped when the event is triggered.

9.3.18 Event Port Rising Edge Input Permission Register (PEVNTRISRm)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
RIS[15:0]															

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b16	Res	Reserved Bit	Read at 0, write at 0	RW
b15~b0	RIS[15:0]	Rising edge detection permission	0: EVNTPmn rising edge event detection is invalid 1: EVNTPmn rising edge event detection is valid PEVNTRISRm.RIS[n] corresponds to EVNTPmn	RW

Note:

Event Port is used as an event source. When the RIS bit is set to 1, the event is output corresponding to the rising edge of the EVNTPmn input to trigger other peripherals. The edge events of EVNTPm0~15 are combined into one event EVENT_PORTm output, and any one of the ports will output the event EVENT_PORTm after detecting the edge.

9.3.19 Event Port Falling Edge Input Enable Register (PEVNTFALRm)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
FAL[15:0]															

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b16	Res	Reserved Bit	Read at 0, write at 0	RW
b15~b0	FAL[15:0]	Falling edge detection permission	0: EVNTPmn falling edge event detection is invalid 1: EVNTPmn falling edge event detection is valid PEVNTRISRm.FAL[n] corresponds to EVNTPmn	RW

Note:

Event Port is used as an event source. When the FAL bit is set to 1, the event is output corresponding to the falling edge of the EVNTP input to trigger other peripherals. The edge events of EVNTPm0~15 are combined into one event EVENT_PORTm output, and any one of the ports will output the event EVENT_PORTm after detecting the edge.

9.3.20 Event Port Input Filter Control Register (PEVNTNFCR)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res					DIVS4[1:0]		NFEN4	Res					DIVS3[1:0]		NFEN3
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res					DIVS2[1:0]		NFEN2	Res					DIVS1[1:0]		NFEN1

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b27	Res	Reserved Bit	Read at 0, write at 0	RW
b26-b25	DIVS4[1:0]	Digital filter sampling clock selection	Event Port4 digital filter sampling clock selection 00：PCLK1 01：PCLK1/8 10：PCLK1/32 11：PCLK1/64	RW
b24	NFEN4	Digital Filtering License	0: Event Port4 digital filter is invalid 1: Event Port4 digital filter is valid	RW
b23~b19	Res	Reserved Bit	Read at 0, write at 0	RW
b18-b17	DIVS3[1:0]	Digital filter sampling clock selection	Event Port3 digital filter sampling clock selection 00：PCLK1 01：PCLK1/8 10：PCLK1/32 11：PCLK1/64	RW
b16	NFEN3	Digital Filtering License	0: Event Port3 digital filter is invalid 1: Event Port3 digital filter is valid	RW
b15~b11	Res	Reserved Bit	Read at 0, write at 0	RW
b10-b9	DIVS2[1:0]	Digital filter sampling clock selection	Event Port2 digital filter sampling clock selection 00：PCLK1 01：PCLK1/8 10：PCLK1/32 11：PCLK1/64	RW
b8	NFEN2	Digital Filtering License	0: Event Port2 digital filter is invalid 1: Event Port2 digital filter is valid	RW
b7~b3	Res	Reserved Bit	Read at 0, write at 0	RW
b2-b1	DIVS1[1:0]	Digital filter sampling clock selection	Event Port1 digital filter sampling clock selection 00：PCLK1 01：PCLK1/8 10：PCLK1/32 11：PCLK1/64	RW
b0	NFEN1	Digital Filtering License	0: Event Port1 digital filter is invalid 1: Event Port1 digital filter is valid	RW

9.4 Precautions

1. Do not set the same feature to multiple ports.
2. When using the analog function, please turn off the digital function of the corresponding port (PCRxy.DDIS=1).
3. Please switch the port function when the output latch valid (PCRxy.LTE=1) to avoid unexpected glitches on the port output during switching.
4. The registers described in Section 9.3 support 16-bit and 32-bit access, but do not support 8-bit access, except for the Event Port related registers which only support 32-bit access. The combination of these registers during 32bit access is as follows:

Table 9-5 List of PORT Registers for 32bit Access

Address	b31 ~ b16	b15 ~ b0
0x4005 3800+0x10*n	Res	PIDRx*1
0x4005 3804+0x10*n	POERx	PODRx
0x4005 3808+0x10*n	PORRx	POSRx
0x4005 380C+0x10*n	Res	POTRx
0x4005 3BF0	PHRPWM	PHRPWM
0x4005 3BF4	Res	PSPCR
0x4005 3BF8	Res	PCCR
0x4005 3BFC	Res	PWPR
0x4005 3C00+0x40*n+0x04*y	PFSRxy	PCRxy

Note:

**1: In the address calculation formula, x=A~D, F corresponds to n=0~3,5.*

10 Interrupt controller (INTC)

10.1 Introduction

The interrupt controller (INTC) selects interrupt events as interrupt requests to NVIC to wake up WFI; selects interrupt events as event inputs (RXEV) to wake up WFE; selects interrupt events to wake up the system in low power mode (Sleep Mode and Stop Mode); controls external interrupts and software interrupts.

The main specifications of INTC are as follows:

1) NVIC interrupt request: INTC is equipped with 303 interrupt events, which are processed and sent to NVIC as interrupt requests (IRQs). It supports 145 IRQs, and each IRQ corresponds to one or more interrupt events.
2) Programmable priority: 16 programmable priorities (4-bit interrupt priority register is used).
3) Non-maskable interrupts: Multiple system interrupt events can be independently selected as non-maskable interrupts, and each interrupt event is equipped with independent enable selection, flag, and flag clear registers.
4) Equipped with 16 external interrupt events.
5) Equipped with multiple interrupt events, please refer to [Interrupt Event] for the specific number.
6) Equipped with 32 software interrupt events.
7) Interrupt wakes up system Sleep Mode and Stop Mode.

Table 10-1 External Interrupt Input Pin

Pin Name	I/O	Description
EIRQ0	Input	External interrupt event 0
EIRQ1	Input	External interrupt event 1
EIRQ2	Input	External interrupt event 2
EIRQ3	Input	External interrupt event 3
EIRQ4	Input	External interrupt event 4
EIRQ5	Input	External interrupt event 5
EIRQ6	Input	External interrupt event 6
EIRQ7	Input	External interrupt event 7
EIRQ8	Input	External interrupt event 8
EIRQ9	Input	External interrupt event 9
EIRQ10	Input	External interrupt event 10
EIRQ11	Input	External interrupt event 11
EIRQ12	Input	External interrupt event 12

Pin Name	I/O	Description
EIRQ13	Input	External interrupt event 13
EIRQ14	Input	External interrupt event 14
EIRQ15	Input	External interrupt event 15

10.2System Block Diagram

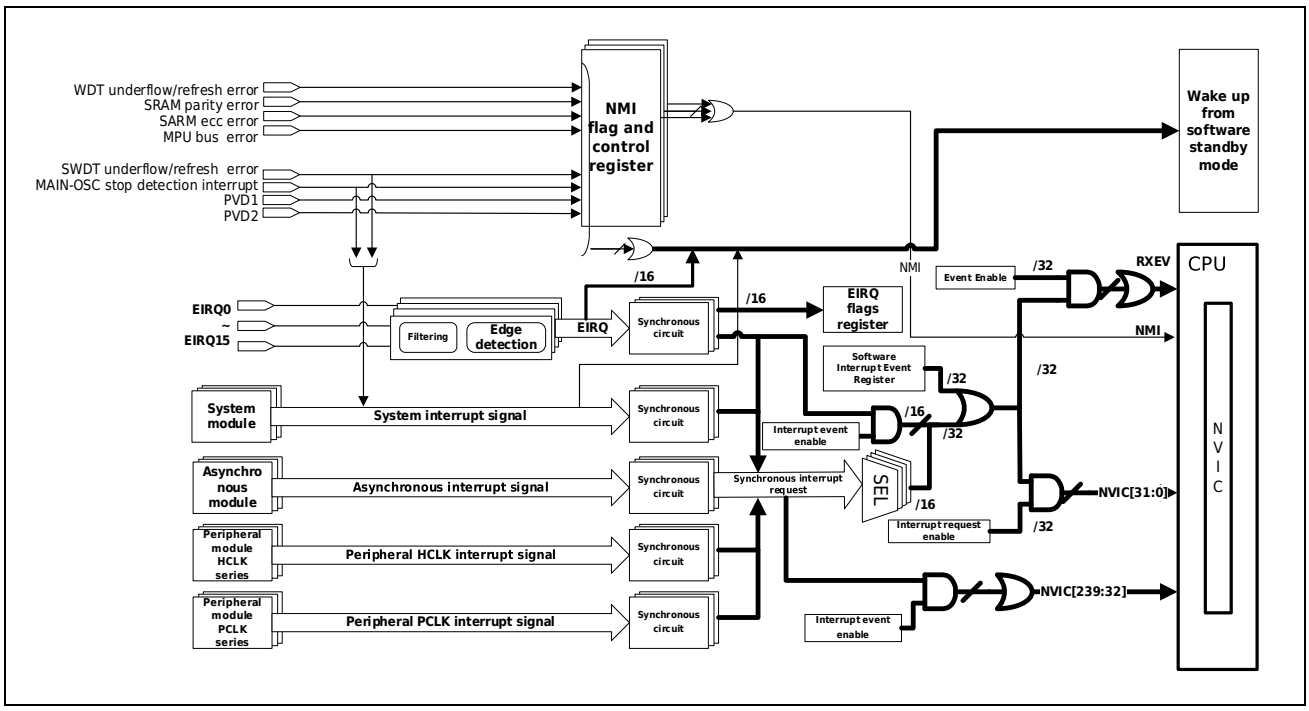


Figure 10-1 Interrupt System Block Diagram

10.3Directional Scale

10.3.1 Interrupt Digraph

Table 10-2 Interrupt Vector Table

Vector Address	Vector No.	IRQ No.	Interrupt Source	Description
ARM Core Interrupts				
0x0000 0000	0	-	ARM core	Initial Stack Pointer
0x0000 0004	1	-	ARM core	Initial Program Counter
0x0000 0008	2	-	ARM core	Non-maskable Interrupt (NMI)
0x0000 000C	3	-	ARM core	Hard Fault
0x0000 0010	4	-	ARM core	MemManage Fault
0x0000 0014	5	-	ARM core	Bus Fault
0x0000 0018	6	-	ARM core	Usage Fault
0x0000 001C	7	-	ARM core	Reserved
0x0000 0020	8	-	ARM core	Reserved
0x0000 0024	9	-	ARM core	Reserved
0x0000 0028	10	-	ARM core	Reserved
0x0000 002C	11	-	ARM core	Supervisor Call (SVCall)
0x0000 0030	12	-	ARM core	Debug Monitor
0x0000 0034	13	-	ARM core	Reserved
0x0000 0038	14	-	ARM core	Pendable request for system service (PendSV)
0x0000 003C	15	-	ARM core	System tick timer (SysTick)
Non-ARM Core Interrupts				
0x0000 0040	16	0	INT_INTSEL0	Register INTC_INTSEL0 selects the interrupt event/software interrupt 0.
0x0000 0044	17	1	INT_INTSEL1	Register INTC_INTSEL1 selects interrupt event/software interrupt 1.
0x0000 0048	18	2	INT_INTSEL2	Register INTC_INTSEL2 selects interrupt event/software interrupt 2.
0x0000 004C	19	3	INT_INTSEL3	Register INTC_INTSEL3 selects interrupt event/software interrupt 3.
0x0000 0050	20	4	INT_INTSEL4	Register INTC_INTSEL4 selects interrupt event/software interrupt 4.
0x0000 0054	21	5	INT_INTSEL5	Register INTC_INTSEL5 selects interrupt event/software interrupt 5.
0x0000 0058	22	6	INT_INTSEL6	Register INTC_INTSEL6 selects interrupt event/software interrupt 6.
0x0000 005C	23	7	INT_INTSEL7	Register INTC_INTSEL7 selects interrupt event/software interrupt 7.
0x0000 0060	24	8	INT_INTSEL8	Register INTC_INTSEL8 selects interrupt event/software interrupt 8.
0x0000 0064	25	9	INT_INTSEL9	Register INTC_INTSEL9 selects interrupt event/software interrupt 9.
0x0000 0068	26	10	INT_INTSEL10	Register INTC_INTSEL10 selects interrupt event/software interrupt 10.
0x0000 006C	27	11	INT_INTSEL11	Register INTC_INTSEL11 selects interrupt event/software interrupt 11.

Vector Address	Vector No.	IRQ No.	Interrupt Source	Description
0x0000 0070	28	12	INT_INTSEL12	Register INTC_INTSEL12 selects interrupt event/software interrupt 12.
0x0000 0074	29	13	INT_INTSEL13	Register INTC_INTSEL13 selects interrupt event/software interrupt 13.
0x0000 0078	30	14	INT_INTSEL14	Register INTC_INTSEL14 selects interrupt event/software interrupt 14.
0x0000 007C	31	15	INT_INTSEL15	Register INTC_INTSEL15 selects interrupt event/software interrupt 15.
0x0000 0080	32	16	PORT	External interrupt 0/software interrupt 16
0x0000 0084	33	17	PORT	External interrupt 1/software interrupt 17
0x0000 0088	34	18	PORT	External interrupt 2/software interrupt 18
0x0000 008C	35	19	PORT	External interrupt 3/software interrupt 19
0x0000 0090	36	20	PORT	External interrupt 4/software interrupt 20
0x0000 0094	37	21	PORT	External interrupt 5/software interrupt 21
0x0000 0098	38	22	PORT	External interrupt 6/software interrupt 22
0x0000 009C	39	23	PORT	External interrupt 7/software interrupt 23
0x0000 00A0	40	24	PORT	External interrupt 8/software interrupt 24
0x0000 00A4	41	25	PORT	External interrupt 9/software interrupt 25
0x0000 00A8	42	26	PORT	External interrupt 10/software interrupt 26
0x0000 00AC	43	27	PORT	External interrupt 11/software interrupt 27
0x0000 00B0	44	28	PORT	External interrupt 12/software interrupt 28
0x0000 00B4	45	29	PORT	External interrupt 13/software interrupt 29
0x0000 00B8	46	30	PORT	External interrupt 14/software interrupt 30
0x0000 00BC	47	31	PORT	External interrupt 15/software interrupt 31
0x0000 00C0	48	32	DMA	Transmission error interrupt
0x0000 00C4	49	33	DMA	Channel 0 transmission completion interrupt/channel 0 data block transmission completion interrupt
0x0000 00C8	50	34	DMA	Channel 1 transmission completion interrupt/channel 1 data block transmission completion interrupt
0x0000 00CC	51	35	DMA	Channel 2 transmission completion interrupt/channel 2 data block transmission completion interrupt
0x0000 00D0	52	36	DMA	Channel 3 transmission completion interrupt/channel 3 data block transmission completion interrupt
0x0000 00D4	53	37	DMA	Channel 4 transmission completion interrupt/channel 4 data block transmission completion interrupt
0x0000 00D8	54	38	DMA	Channel 5 transmission completion interrupt/channel 5 data block transmission completion interrupt
0x0000 00DC	55	39	DMA	Channel 6 transmission completion interrupt/channel 5 data block transmission completion interrupt
0x0000 00E0	56	40	DMA	Channel 7 transmission completion interrupt/channel 5 data block transmission completion interrupt
0x0000 00E4	57	41	EFM	Program (erase) error interrupt / read/write conflict error interrupt
0x0000 00E8	58	42	EFM	Operation end interrupt
0x0000 00EC	59	43	Timer0_1	Channel A, B count compare match interrupt/input capture interrupt/count overflow interrupt
0x0000 00F0	60	44	Timer0_2	Channel A, B count compare match interrupt/input capture interrupt/count overflow interrupt
0x0000 00F4	61	45	RTC	Alarm clock interruption, regular period interruption.

Vector Address	Vector No.	IRQ No.	Interrupt Source	Description
0x0000 00F8	62	46	CMU	XTAL oscillation fault interrupt
0x0000 00FC	63	47	PWC	WKTM compare match interrupt
0x0000 0100	64	48	SWDT	Count underflow/refresh error interrupt
0x0000 0104	65	49	FPU	FPU Interrupt
0x0000 0108	66	50	Timer6_1	Universal count compare match interrupt/input capture interrupt
0x0000 010C	67	51	Timer6_1	Count cycle match interrupt
0x0000 0110	68	52	Timer6_1	Dead time error interrupt
0x0000 0114	69	53	Timer6_1	Dedicated count compare match interrupt
0x0000 0118	70	54	TimerA_1	Count cycle match interrupt
0x0000 011C	71	55	TimerA_1	Count comparison match interrupt
0x0000 0120	72	56	Timer6_2	Universal count compare match interrupt/input capture interrupt
0x0000 0124	73	57	Timer6_2	Count cycle match interrupt
0x0000 0128	74	58	Timer6_2	Dead time error interrupt
0x0000 012C	75	59	Timer6_2	Dedicated count compare match interrupt
0x0000 0130	76	60	TimerA_2	Count cycle match interrupt
0x0000 0134	77	61	TimerA_2	Count comparison match interrupt
0x0000 0138	78	62	Timer6_3	Universal count compare match interrupt/input capture interrupt
0x0000 013C	79	63	Timer6_3	Count cycle match interrupt
0x0000 0140	80	64	Timer6_3	Dead time error interrupt
0x0000 0144	81	65	Timer6_3	Dedicated count compare match interrupt
0x0000 0148	82	66	TimerA_3	Count cycle match interrupt
0x0000 014C	83	67	TimerA_3	Count comparison match interrupt
0x0000 0150	84	68	Timer6_4	Universal count compare match interrupt/input capture interrupt
0x0000 0154	85	69	Timer6_4	Count cycle match interrupt
0x0000 0158	86	70	Timer6_4	Dead time error interrupt
0x0000 015C	87	71	Timer6_4	Dedicated count compare match interrupt
0x0000 0160	88	72	TimerA_4	Count cycle match interrupt
0x0000 0164	89	73	TimerA_4	Count comparison match interrupt
0x0000 0168	90	74	HRPWM_1	Universal count compare match interrupt
0x0000 016C	91	75	HRPWM_1	Count cycle/zero match interrupt
0x0000 0170	92	76	HRPWM_1	Dedicated count compare match interrupt
0x0000 0174	93	77	HRPWM_1	Capture Interrupt
0x0000 0178	94	78	HRPWM_2	Universal count compare match interrupt
0x0000 017C	95	79	HRPWM_2	Count cycle/zero match interrupt
0x0000 0180	96	80	HRPWM_2	Dedicated count compare match interrupt
0x0000 0184	97	81	HRPWM_2	Capture Interrupt
0x0000 0188	98	82	HRPWM	Burst mode cycle match interrupt/calibration complete

Vector Address	Vector No.	IRQ No.	Interrupt Source	Description
				interrupt
0x0000 018C	99	83	EMB	Group0 interruption
0x0000 0190	100	84	EMB	Group1 Interruption
0x0000 0194	101	85	EMB	Group2 Interruption
0x0000 0198	102	86	EMB	Group3 Interruption
0x0000 019C	103	87	HRPWM_3	Universal count compare match interrupt
0x0000 01A0	104	88	HRPWM_3	Count cycle/zero match interrupt
0x0000 01A4	105	89	HRPWM_3	Dedicated count compare match interrupt
0x0000 01A8	106	90	HRPWM_3	Capture Interrupt
0x0000 01AC	107	91	HRPWM_4	Universal count compare match interrupt
0x0000 01B0	108	92	HRPWM_4	Count cycle/zero match interrupt
0x0000 01B4	109	93	HRPWM_4	Dedicated count compare match interrupt
0x0000 01B8	110	94	HRPWM_4	Capture Interrupt
0x0000 01BC	111	95	EMB	Group4 Interruption
0x0000 01C0	112	96	EMB	Group5 Interruption
0x0000 01C4	113	97	EMB	Group6 Interruption
0x0000 01C8	114	98	EMB	Group7 Interruption
0x0000 01CC	115	99	EMB	Group8 Interruption
0x0000 01D0	116	100	HRPWM_5	Universal count compare match interrupt
0x0000 01D4	117	101	HRPWM_5	Count cycle/zero match interrupt
0x0000 01D8	118	102	HRPWM_5	Dedicated count compare match interrupt
0x0000 01DC	119	103	HRPWM_5	Capture Interrupt
0x0000 01E0	120	104	HRPWM_6	Universal count compare match interrupt
0x0000 01E4	121	105	HRPWM_6	Count cycle/zero match interrupt
0x0000 01E8	122	106	HRPWM_6	Dedicated count compare match interrupt
0x0000 01EC	123	107	HRPWM_6	Capture Interrupt
0x0000 01F0	124	108	MCANRAM	MCAN RAM ECC error interrupt
0x0000 01F4	125	109	MCAN_1	MCAN_1 Interrupt0
0x0000 01F8	126	110	MCAN_1	MCAN_1 interrupt1
0x0000 01FC	127	111	MCAN_2	MCAN_2 Interrupt 0
0x0000 0200	128	112	MCAN_2	MCAN_2 Interrupt 1
0x0000 0204	129	113	Timer4_1	Universal count compare match interrupt
0x0000 0208	130	114	Timer4_1	Count cycle match interrupt
0x0000 020C	131	115	Timer4_1	Heavy load count matching interrupt
0x0000 0210	132	116	Timer4_1	Dedicated count compare match interrupt
0x0000 0214	133	117	CMP	CMP1 Interruption
0x0000 0218	134	118	CMP	CMP2 Interruption

Vector Address	Vector No.	IRQ No.	Interrupt Source	Description
0x0000 021C	135	119	CMP	CMP3 Interruption
0x0000 0220	136	120	I2C	I2C Interruption
0x0000 0224	137	121	USART_1	USART_1 interrupts (except TCI and RX line wake-up Stop Mode interrupts)
0x0000 0228	138	122	USART_1	USART_1_TCI Interruption
0x0000 022C	139	123	SPI	SPI Interrupt
0x0000 0230	140	124	TimerA_5	Count cycle match interrupt
0x0000 0234	141	125	TimerA_5	Count comparison match interrupt
0x0000 0238	142	126	PORT	Event Port1 Interrupt
0x0000 023C	143	127	PORT	Event Port2 Interrupt
0x0000 0240	144	128	PORT	Event Port3 Interrupt
0x0000 0244	145	129	PORT	Event Port4 Interrupt
0x0000 0248	146	130	USART_2	USART_2 interrupts (except TCI interrupts)
0x0000 024C	147	131	USART_2	USART_2_TCI Interruption
0x0000 0250	148	132	USART_3	USART_3 interrupts (except TCI interrupts)
0x0000 0254	149	133	USART_3	USART_3_TCI Interruption
0x0000 0258	150	134	USART_4	USART_4 interrupts (except TCI interrupts)
0x0000 025C	151	135	USART_4	USART_4_TCI Interruption
0x0000 0264	153	137	USART_1	RX Stop Mode wake-up interrupt
0x0000 0268	154	138	PWC	Programmable Voltage Detect 1 (PVD1) Interrupt
0x0000 026C	155	139	PWC	Programmable Voltage Detect 1 (PVD2) Interrupt
0x0000 0270	156	140	CMU	FCM clock frequency measurement interrupt
0x0000 0274	157	141	WDT	Count underflow/refresh error interrupt
0x0000 0278	158	142	CTC	HRC frequency calibration value overflow interrupt
0x0000 027C	159	143	ADC_1	ADC_1 Interruption
0x0000 0280	160	144	ADC_2	ADC_2 Interruption
0x0000 0284	161	145	ADC_3	ADC_3 Interruption

Note:

- Please refer to [*Interrupt Event List*] for the interrupt event number selected by the INTC_INTSEL register.
- *The vector corresponding to IRQ number 136 is not configured with any interrupt event, so please do not use it.*

10.3.2 Interrupt Event List

Interrupt events are generated by peripherals. When they are input as IRQ, they are called interrupt sources; when they are input as RXEV, they are called event sources; when they are peripheral trigger events, they are called trigger sources; interrupt events can also be used as conditions for returning from low power mode.

Table 10-3 Interrupt Event Table

SN	Interrupt Event Number	Modules	Interrupt Name	Interrupt Type*1	Interrupt Source	Trigger source	NVIC Interrupt Request (IRQ) Number Mapping	
							Type A (INTC_INTSEL Write Event Number * 2)	Type B (Fixed)
0	0x000	PORT	EIRQ0	P/L	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[16]
1	0x001		EIRQ1	P/L	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[17]
2	0x002		EIRQ2	P/L	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[18]
3	0x003		EIRQ3	P/L	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[19]
4	0x004		EIRQ4	P/L	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[20]
5	0x005		EIRQ5	P/L	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[21]
6	0x006		EIRQ6	P/L	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[22]
7	0x007		EIRQ7	P/L	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[23]
8	0x008		EIRQ8	P/L	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[24]
9	0x009		EIRQ9	P/L	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[25]
10	0x00A		EIRQ10	P/L	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[26]
11	0x00B		EIRQ11	P/L	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[27]
12	0x00C		EIRQ12	P/L	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[28]
13	0x00D		EIRQ13	P/L	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[29]
14	0x00E		EIRQ14	P/L	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[30]
15	0x00F		EIRQ15	P/L	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[31]
16	0x010	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
17	0x011	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
18	0x012	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
19	0x013	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
20	0x014	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
21	0x015	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
22	0x016	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
23	0x017	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
24	0x018	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
25	0x019	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
26	0x01A	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-

SN	Interrupt Event Number	Modules	Interrupt Name	Interrupt Type*1	Interrupt Source	Trigger source	NVIC Interrupt Request (IRQ) Number Mapping	
							Type A (INTC_INTSEL Write Event Number * 2)	Type B (Fixed)
27	0x01B	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
28	0x01C	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
29	0x01D	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
30	0x01E	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
31	0x01F	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
32	0x020	DMA	DMA_ERR	P	✓	-	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[32]
33	0x021		DMA_TC0	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[33]
34	0x022		DMA_BTC0	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
35	0x023		DMA_TC1	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[34]
36	0x024		DMA_BTC1	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
37	0x025		DMA_TC2	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[35]
38	0x026		DMA_BTC2	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
39	0x027		DMA_TC3	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[36]
40	0x028		DMA_BTC3	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
41	0x029		DMA_TC4	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[37]
42	0x02A		DMA_BTC4	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
43	0x02B		DMA_TC5	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[38]
44	0x02C		DMA_BTC5	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
45	0x02D		DMA_TC6	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[39]
46	0x02E		DMA_BTC6	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
47	0x02F		DMA_TC7	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[40]
48	0x030		DMA_BTC7	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
49	0x031	EFM	EFM_PEERR	P	✓	-	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[41]
50	0x032		EFM_RDCOL	P	✓	-	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
51	0x033		EFM_OPTEND	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[42]
52	0x034	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
53	0x035	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
54	0x036	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
55	0x037	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
56	0x038	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
57	0x039	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
58	0x03A	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
59	0x03B	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
60	0x03C	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-

SN	Interrupt Event Number	Modules	Interrupt Name	Interrupt Type*1	Interrupt Source	Trigger source	NVIC Interrupt Request (IRQ) Number Mapping	
							Type A (INTC_INTSEL Write Event Number * 2)	Type B (Fixed)
61	0x03D	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
62	0x03E	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
63	0x03F	FPU	FPU_ERR-	L	✓	-	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[43]
64	0x040	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
65	0x041	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
66	0x042	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
67	0x043	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
68	0x044	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
69	0x045	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
70	0x046	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
71	0x047	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
72	0x048	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
73	0x049	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
74	0x04A	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
75	0x04B	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
76	0x04C	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
77	0x04D	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
78	0x04E	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
79	0x04F	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
80	0x050	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
81	0x051	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
82	0x052	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
83	0x053	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
84	0x054	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
85	0x055	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
86	0x056	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
87	0x057	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
88	0x058	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
89	0x059	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
90	0x05A	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
91	0x05B	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
92	0x05C	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
93	0x05D	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
94	0x05E	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-

SN	Interrupt Event Number	Modules	Interrupt Name	Interrupt Type*1	Interrupt Source	Trigger source	NVIC Interrupt Request (IRQ) Number Mapping	
							Type A (INTC_INTSEL Write Event Number * 2)	Type B (Fixed)
95	0x05F	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
96	0x060	Timer0_1	TMR0_1_CMPA	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[44]
97	0x061		TMR0_1_CMPB	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
98	0x062		TMR0_1_OVFA	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
99	0x063	Timer0_1	TMR0_1_OVFB	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
100	0x064	Timer0_2	TMR0_2_CMPA	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[45]
101	0x065		TMR0_2_CMPB	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
102	0x066		TMR0_2_OVFA	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
103	0x067		TMR0_2_OVFB	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
104	0x068	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
105	0x069	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
106	0x06A	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
107	0x06B	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
108	0x06C	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
109	0x06D	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
110	0x06E	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
111	0x06F	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
112	0x070	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
113	0x071	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
114	0x072	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
115	0x073	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
116	0x074	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
117	0x075	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
118	0x076	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
119	0x077	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
120	0x078	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
121	0x079	RTC	RTC_ALM	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[46]
122	0x07A		RTC_PRD	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
123	0x07B	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
124	0x07C	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
125	0x07D	XTAL	XTAL_STOP	L	✓	-	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[47]
126	0x07E	WKTM	WKTM_PRD	P	✓	-	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[48]
127	0x07F	SWDT	SWDT_REFUDF	P	✓	-	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[49]
128	0x080	Timer6_1	TMR6_1_GCMA	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[50]

SN	Interrupt Event Number	Modules	Interrupt Name	Interrupt Type*1	Interrupt Source	Trigger source	NVIC Interrupt Request (IRQ) Number Mapping	
							Type A (INTC_INTSEL Write Event Number * 2)	Type B (Fixed)
129	0x081		TMR6_1_GCMB	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
130	0x082		TMR6_1_GCMC	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
131	0x083		TMR6_1_GCMD	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
132	0x084		TMR6_1_GCME	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
133	0x085		TMR6_1_GCMF	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
134	0x086	Timer6_1	TMR6_1_GOVF	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[51]
135	0x087		TMR6_1_GUDF	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
136	0x088		TMR6_1_GDTE	P	✓	-	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[52]
137	0x089		TMR6_1_SCMA	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[53]
138	0x08A		TMR6_1_SCMB	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
139	0x08B	TimerA_1	TMRA_1_OVF	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[54]
140	0x08C		TMRA_1_UDF	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
141	0x08D		TMRA_1_CMP	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[55]
142	0x08E	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
143	0x08F	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
144	0x090	Timer6_2	TMR6_2_GCMA	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[56]
145	0x091		TMR6_2_GCMB	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
146	0x092		TMR6_2_GCMC	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
147	0x093		TMR6_2_GCMD	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
148	0x094		TMR6_2_GCME	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
149	0x095		TMR6_2_GCMF	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
150	0x096		TMR6_2_GOVF	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[57]
151	0x097		TMR6_2_GUDF	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
152	0x098		TMR6_2_GDTE	P	✓	-	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[58]
153	0x099		TMR6_2_SCMA	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[59]
154	0x09A		TMR6_2_SCMB	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
155	0x09B	TimerA_2	TMRA_2_OVF	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[60]
156	0x09C		TMRA_2_UDF	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
157	0x09D		TMRA_2_CMP	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[61]
158	0x09E	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
159	0x09F	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
160	0x0A0	Timer6_3	TMR6_3_GCMA	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[62]
161	0x0A1		TMR6_3_GCMB	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
162	0x0A2		TMR6_3_GCMC	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑

SN	Interrupt Event Number	Modules	Interrupt Name	Interrupt Type*1	Interrupt Source	Trigger source	NVIC Interrupt Request (IRQ) Number Mapping	
							Type A (INTC_INTSEL Write Event Number * 2)	Type B (Fixed)
163	0x0A3		TMR6_3_GCMD	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
164	0x0A4		TMR6_3_GCME	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
165	0x0A5		TMR6_3_GCMF	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
166	0x0A6		TMR6_3_GOVF	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[63]
167	0x0A7		TMR6_3_GUDF	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
168	0x0A8		TMR6_3_GDTE	P	✓	-	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[64]
169	0x0A9	Timer6_3	TMR6_3_SCMA	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[65]
170	0x0AA		TMR6_3_SCMB	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
171	0x0AB		TMRA_3_OVF	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[66]
172	0x0AC	TimerA_3	TMRA_3_UDF	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
173	0x0AD		TMRA_3_CMP	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[67]
174	0x0AE	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
175	0x0AF	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
176	0x0B0		TMR6_4_GCMA	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[68]
177	0x0B1		TMR6_4_GCMB	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
178	0x0B2		TMR6_4_GCMC	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
179	0x0B3		TMR6_4_GCMD	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
180	0x0B4		TMR6_4_GCME	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
181	0x0B5	Timer6_4	TMR6_4_GCMF	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
182	0x0B6		TMR6_4_GOVF	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[69]
183	0x0B7		TMR6_4_GUDF	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
184	0x0B8		TMR6_4_GDTE	P	✓	-	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[70]
185	0x0B9		TMR6_4_SCMA	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[71]
186	0x0BA		TMR6_4_SCMB	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
187	0x0BB		TMRA_4_OVF	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[72]
188	0x0BC	TimerA_4	TMRA_4_UDF	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
189	0x0BD		TMRA_4_CMP	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[73]
190	0x0BE	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
191	0x0BF	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
192	0x0C0		HRPWM_1_GCMA	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[74]
193	0x0C1		HRPWM_1_GCMB	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
194	0x0C2	HRPWM_1	HRPWM_1_GCMC	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
195	0x0C3		HRPWM_1_GCMD	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
196	0x0C4		HRPWM_1_GCME	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑

SN	Interrupt Event Number	Modules	Interrupt Name	Interrupt Type*1	Interrupt Source	Trigger source	NVIC Interrupt Request (IRQ) Number Mapping	
							Type A (INTC_INTSEL Write Event Number * 2)	Type B (Fixed)
197	0x0C5		HRPWM_1_GCMF	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
198	0x0C6		HRPWM_1_GOVF	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[75]
199	0x0C7		HRPWM_1_GUDF	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
200	0x0C8		HRPWM_1_SCMA	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[76]
201	0x0C9		HRPWM_1_SCMB	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
202	0x0CA		HRPWM_1_GCAPA	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[77]
203	0x0CB		HRPWM_1_GCAPB	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
204	0x0CC	HRPWM_2	HRPWM_2_GCMA	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[78]
205	0x0CD		HRPWM_2_GCMB	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
206	0x0CE		HRPWM_2_GCMC	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
207	0x0CF		HRPWM_2_GCMD	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
208	0x0D0		HRPWM_2_GCME	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
209	0x0D1		HRPWM_2_GCMF	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
210	0x0D2		HRPWM_2_GOVF	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[79]
211	0x0D3		HRPWM_2_GUDF	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
212	0x0D4		HRPWM_2_SCMA	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[80]
213	0x0D5		HRPWM_2_SCMB	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
214	0x0D6		HRPWM_2_GCAPA	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[81]
215	0x0D7		HRPWM_2_GCAPB	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
216	0x0D8	HRPWM	HRPWM_BMOVF	P	√	-	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[82]
217	0x0D9		HRPWM_CALENF	P	√	-	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
218	0x0DA		HRPWM_DLYPRT	P	√	-	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
219	0x0DB		HRPWM_EEDET	P	-	√	IRQ[0]~[15] (INTC_INTSEL0~15)	-
220	0x0DC	EMB	EMB_GR0	L	√	-	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[83]
221	0x0DD		EMB_GR1	L	√	-	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[84]
222	0x0DE		EMB_GR2	L	√	-	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[85]
223	0x0DF		EMB_GR3	L	√	-	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[86]
224	0x0E0	HRPWM_3	HRPWM_3_GCMA	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[87]
225	0x0E1		HRPWM_3_GCMB	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
226	0x0E2		HRPWM_3_GCMC	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
227	0x0E3		HRPWM_3_GCMD	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
228	0x0E4		HRPWM_3_GCME	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
229	0x0E5		HRPWM_3_GCMF	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
230	0x0E6		HRPWM_3_GOVF	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[88]

SN	Interrupt Event Number	Modules	Interrupt Name	Interrupt Type*1	Interrupt Source	Trigger source	NVIC Interrupt Request (IRQ) Number Mapping	
							Type A (INTC_INTSEL Write Event Number * 2)	Type B (Fixed)
231	0x0E7		HRPWM_3_GUDF	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
232	0x0E8		HRPWM_3_SCMA	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[89]
233	0x0E9		HRPWM_3_SCMB	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
234	0x0EA		HRPWM_3_GCAPA	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[90]
235	0x0EB		HRPWM_3_GCAPB	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
236	0x0EC	HRPWM_4	HRPWM_4_GCMA	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[91]
237	0x0ED		HRPWM_4_GCMB	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
238	0x0EE		HRPWM_4_GCMC	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
239	0x0EF	HRPWM_4	HRPWM_4_GCMD	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
240	0x0F0		HRPWM_4_GCME	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
241	0x0F1		HRPWM_4_GCMF	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
242	0x0F2		HRPWM_4_GOVF	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[92]
243	0x0F3		HRPWM_4_GUDF	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
244	0x0F4		HRPWM_4_SCMA	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[93]
245	0x0F5		HRPWM_4_SCMB	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
246	0x0F6		HRPWM_4_GCAPA	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[94]
247	0x0F7		HRPWM_4_GCAPB	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
248	0x0F8	EMB	EMB_GR4	L	√	-	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[95]
249	0x0F9		EMB_GR5	L	√	-	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[96]
250	0x0FA		EMB_GR6	L	√	-	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[97]
251	0x0FB		EMB_GR7	L	√	-	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[98]
252	0x0FC		EMB_GR8	L	√	-	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[99]
253	0x0FD	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
254	0x0FE	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
255	0x0FF	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
256	0x100	HRPWM_5	HRPWM_5_GCMA	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[100]
257	0x101		HRPWM_5_GCMB	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
258	0x102		HRPWM_5_GCMC	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
259	0x103		HRPWM_5_GCMD	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
260	0x104		HRPWM_5_GCME	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
261	0x105		HRPWM_5_GCMF	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
262	0x106		HRPWM_5_GOVF	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[101]
263	0x107		HRPWM_5_GUDF	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
264	0x108		HRPWM_5_SCMA	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[102]

SN	Interrupt Event Number	Modules	Interrupt Name	Interrupt Type*1	Interrupt Source	Trigger source	NVIC Interrupt Request (IRQ) Number Mapping	
							Type A (INTC_INTSEL Write Event Number * 2)	Type B (Fixed)
265	0x109		HRPWM_5_SCMB	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
266	0x10A		HRPWM_5_GCAPA	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[103]
267	0x10B		HRPWM_5_GCAPB	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
268	0x10C	HRPWM_6	HRPWM_6_GCMA	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[104]
269	0x10D		HRPWM_6_GCMB	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
270	0x10E		HRPWM_6_GCMC	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
271	0x10F		HRPWM_6_GCMD	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
272	0x110		HRPWM_6_GCME	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
273	0x111		HRPWM_6_GCMF	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
274	0x112	HRPWM_6	HRPWM_6_GOVF	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[105]
275	0x113		HRPWM_6_GUDF	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
276	0x114		HRPWM_6_SCMA	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[106]
277	0x115		HRPWM_6_SCMB	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
278	0x116		HRPWM_6_GCAPA	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[107]
279	0x117		HRPWM_6_GCAPB	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
280	0x118	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
281	0x119	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
282	0x11A	MCANRAM	MCANRAM_ECCER	L	✓	-	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[108]
283	0x11B	MCAN	MCAN_1_INT0	L	✓	-	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[109]
284	0x11C		MCAN_1_INT1	L	✓	-	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[110]
285	0x11D		MCAN_2_INT0	L	✓	-	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[111]
286	0x11E		MCAN_2_INT1	L	✓	-	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[112]
287	0x11F	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
288	0x120	Timer4	TMR4_GCMUH	L	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[113]
289	0x121		TMR4_GCMUL	L	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
290	0x122		TMR4_GCMVH	L	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
291	0x123		TMR4_GCMVL	L	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
292	0x124		TMR4_GCMWH	L	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
293	0x125		TMR4_GCMWL	L	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
294	0x126		TMR4_GCMXH	L	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
295	0x127		TMR4_GCMXL	L	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
296	0x128		TMR4_GOVF	L	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[114]
297	0x129		TMR4_GUDF	L	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
298	0x12A		TMR4_GRLU	L	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[115]

SN	Interrupt Event Number	Modules	Interrupt Name	Interrupt Type*1	Interrupt Source	Trigger source	NVIC Interrupt Request (IRQ) Number Mapping	
							Type A (INTC_INTSEL Write Event Number * 2)	Type B (Fixed)
299	0x12B		TMR4_GRLV	L	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
300	0x12C		TMR4_GRLW	L	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
301	0x12D		TMR4_GRLX	L	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
302	0x12E		TMR4_SCM0	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[116]
303	0x12F		TMR4_SCM1	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
304	0x130		TMR4_SCM2	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
305	0x131		TMR4_SCM3	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
306	0x132		TMR4_SCM4	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
307	0x133		TMR4_SCM5	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
308	0x134		TMR4_SCM6	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
309	0x135	Timer4	TMR4_SCM7	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
310	0x136		CMP_1_IRQ	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[117]
311	0x137	CMP	CMP_2_IRQ	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[118]
312	0x138		CMP_3_IRQ	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[119]
313	0x139	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
314	0x13A	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
315	0x13B	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
316	0x13C		I2C_RXI	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[120]
317	0x13D	I2C	I2C_TXI	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
318	0x13E		I2C_TEI	L	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
319	0x13F		I2C_EEI	L	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
320	0x140	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
321	0x141		USART_1_EI	L	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[121]
322	0x142		USART_1_RI	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
323	0x143		USART_1_TI	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
324	0x144	USART_1	USART_1_RTO	L	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
325	0x145		USART_1_TENDI	L	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
326	0x146		USART_1_TCI	L	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[122]
327	0x147	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
328	0x148	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
329	0x149	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
330	0x14A		SPI_SPRI	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[123]
331	0x14B	SPI	SPI_SPTI	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
332	0x14C		SPI_SPII	L	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑

SN	Interrupt Event Number	Modules	Interrupt Name	Interrupt Type*1	Interrupt Source	Trigger source	NVIC Interrupt Request (IRQ) Number Mapping	
							Type A (INTC_INTSEL Write Event Number * 2)	Type B (Fixed)
333	0x14D		SPI_SPEI	L	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
334	0x14E		SPI_SPEND	P	-	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
335	0x14F	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
336	0x150	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
337	0x151	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
338	0x152	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
339	0x153	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
340	0x154	TimerA_5	TMRA_5_OVF	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[124]
341	0x155		TMRA_5_UDF	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
342	0x156		TMRA_5_CMP	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[125]
343	0x157	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
344	0x158	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
345	0x159	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
346	0x15A	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
347	0x15B	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
348	0x15C	EVENT port	EVENT_PORT1	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[126]
349	0x15D		EVENT_PORT2	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[127]
350	0x15E		EVENT_PORT3	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[128]
351	0x15F		EVENT_PORT4	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[129]
352	0x160	USART_2	USART_2_BRKWKPI	L	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[130]
353	0x161		USART_2_EI	L	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
354	0x162		USART_2_RI	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
355	0x163		USART_2_TI	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
356	0x164		USART_2_RTO	L	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
357	0x165		USART_2_TENDI	L	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
358	0x166		USART_2_TCI	L	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[131]
359	0x167	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
360	0x168	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
361	0x169	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
362	0x16A	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
363	0x16B	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
364	0x16C	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
365	0x16D	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
366	0x16E	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-

SN	Interrupt Event Number	Modules	Interrupt Name	Interrupt Type*1	Interrupt Source	Trigger source	NVIC Interrupt Request (IRQ) Number Mapping	
							Type A (INTC_INTSEL Write Event Number * 2)	Type B (Fixed)
367	0x16F	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
368	0x170	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
369	0x171	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
370	0x172	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
371	0x173	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
372	0x174	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
373	0x175	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
374	0x176	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
375	0x177	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
376	0x178	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
377	0x179	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
378	0x17A	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
379	0x17B	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
380	0x17C	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
381	0x17D	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
382	0x17E	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
383	0x17F	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
384	0x180	USART_3	USART_3_BRKWKPI	L	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[132]
385	0x181		USART_3_EI	L	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
386	0x182		USART_3_RI	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
387	0x183		USART_3_TI	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
388	0x184		USART_3_RTO	L	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
389	0x185		USART_3_TENDI	L	✓	-	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
390	0x186		USART_3_TCI	L	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[133]
391	0x187	USART_4	USART_4_BRKWKPI	L	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[134]
392	0x188		USART_4_EI	L	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
393	0x189		USART_4_RI	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
394	0x18A		USART_4_TI	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
395	0x18B		USART_4_RTO	L	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
396	0x18C		USART_4_TENDI	L	✓	-	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
397	0x18D		USART_4_TCI	L	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[135]
398	0x18E	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
399	0x18F	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-

SN	Interrupt Event Number	Modules	Interrupt Name	Interrupt Type*1	Interrupt Source	Trigger source	NVIC Interrupt Request (IRQ) Number Mapping	
							Type A (INTC_INTSEL Write Event Number * 2)	Type B (Fixed)
400	0x190	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
401	0x191	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
402	0x192	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
403	0x193	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	
404	0x194	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	
405	0x195	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	
406	0x196	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	
407	0x197	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	
408	0x198	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	
409	0x199	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	
410	0x19A	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	
411	0x19B	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	
412	0x19C	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
413	0x19D	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
414	0x19E	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
415	0x19F	AOS	AOS_STRG*3	P	-	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	-
416	0x1A0	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
417	0x1A1	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
418	0x1A2	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
419	0x1A3	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
420	0x1A4	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
421	0x1A5	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
422	0x1A6	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
423	0x1A7	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
424	0x1A8	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
425	0x1A9	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
426	0x1AA	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
427	0x1AB	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
428	0x1AC	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
429	0x1AD	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
430	0x1AE	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
431	0x1AF	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
432	0x1B0	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
433	0x1B1	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-

SN	Interrupt Event Number	Modules	Interrupt Name	Interrupt Type*1	Interrupt Source	Trigger source	NVIC Interrupt Request (IRQ) Number Mapping	
							Type A (INTC_INTSEL Write Event Number * 2)	Type B (Fixed)
434	0x1B2	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
435	0x1B3	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
436	0x1B4	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
437	0x1B5	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
438	0x1B6	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
439	0x1B7	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
440	0x1B8	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
441	0x1B9	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
442	0x1BA	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
443	0x1BB	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
444	0x1BC	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
445	0x1BD	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
446	0x1BE	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
447	0x1BF	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
448	0x1C0	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
449	0x1C1	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
450	0x1C2	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
451	0x1C3	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
452	0x1C4	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
453	0x1C5	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
454	0x1C6	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
455	0x1C7	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
456	0x1C8	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
457	0x1C9	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
458	0x1CA	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
459	0x1CB	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
460	0x1CC	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
461	0x1CD	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
462	0x1CE	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
463	0x1CF	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
464	0x1D0	USART_1	USART_1_WUPI	Edge	✓	-	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[137]
465	0x1D1	PVD	PVD_PVD1	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[138]
466	0x1D2		PVD_PVD2	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[139]
467	0x1D3	-	-		-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-

SN	Interrupt Event Number	Modules	Interrupt Name	Interrupt Type*1	Interrupt Source	Trigger source	NVIC Interrupt Request (IRQ) Number Mapping	
							Type A (INTC_INTSEL Write Event Number * 2)	Type B (Fixed)
468	0x1D4	FCM	FCMFERRI	L	✓	-	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[140]
469	0x1D5		FCMMENDI	L	✓	-	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
470	0x1D6		FCMCOVFI	L	✓	-	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
471	0x1D7	WDT	WDT_REFUDF	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[141]
472	0x1D8	CTC	CTC_ERR	P	✓	-	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[142]
473	0x1D9	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
474	0x1DA	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
475	0x1DB	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
476	0x1DC	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
477	0x1DD	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
478	0x1DE	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
479	0x1DF	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
480	0x1E0	ADC_1	ADC_1_EOCA	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[143]
481	0x1E1	ADC_1	ADC_1_EOCB	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
482	0x1E2		ADC_1_CMP0	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
483	0x1E3		ADC_1_CMP1	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
484	0x1E4	ADC_2	ADC_2_EOCA	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[144]
485	0x1E5		ADC_2_EOCB	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
486	0x1E6		ADC_2_CMP0	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
487	0x1E7		ADC_2_CMP1	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
488	0x1E8	ADC_3	ADC_3_EOCA	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	IRQ[145]
489	0x1E9		ADC_3_EOCB	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
490	0x1EA		ADC_3_CMP0	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
491	0x1EB		ADC_3_CMP1	P	✓	✓	IRQ[0]~[15] (INTC_INTSEL0~15)	↑
492	0x1EC	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
493	0x1ED	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
494	0x1EE	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
495	0x1EF	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
496	0x1F0	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
497	0x1F1	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
498	0x1F2	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
499	0x1F3	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
500	0x1F4	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
501	0x1F5	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-

SN	Interrupt Event Number	Modules	Interrupt Name	Interrupt Type*1	Interrupt Source	Trigger source	NVIC Interrupt Request (IRQ) Number Mapping	
							Type A (INTC_INTSEL Write Event Number * 2)	Type B (Fixed)
502	0x1F6	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
503	0x1F7	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
504	0x1F8	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
505	0x1F9	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
506	0x1FA	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
507	0x1FB	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
508	0x1FC	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
509	0x1FD	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
510	0x1FE	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-
511	0x1FF	-	-	-	-	-	IRQ[0]~[15] (INTC_INTSEL0~15)	-

*1: “P” indicates pulse interrupt, and “L” indicates level interrupt. The interrupt flag must be cleared in the interrupt handler for level interrupts.

*2: If the interrupt event is not configured, the number selected by the interrupt selection register invalid.

*3: AOS_STRG is generated by software setting the STRG bit in the INTSFTTRG register.

*4: “↑” indicates the IRQ number is the same as above.

10.4Functional Description

10.4.1 Nomaskable Interrupt

The non-maskable interrupt events are as follows:

- Detecting the main generator stops terminating
- WDT underflow/refresh interrupt
- SWDT underflow/refresh interrupt
- Low voltage detection PVD1 interrupt
- Low voltage detection PVD2 interrupt
- SRAM parity error interrupt
- SRAM ECC parity error interrupt
- MPU bus error interrupt

Non-maskable interrupts have the highest priority and can select multiple interrupt events. The status of each interrupt event can be queried through the interrupt flag register (INTC_NMIFR). Please make sure all the flag bits are 0 before the non-maskable interrupt processing exits.

When you need to use non-maskable interrupts, please set them according to the following process:

1. Configure the corresponding function 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.

Note:

– *Once the corresponding bit of INTC_NMIER is set to 1, it cannot be changed unless reset with RESET.*

10.4.2 External Interrupt

Equipped with 16 external interrupt events, the interrupt status can be queried through the interrupt flag register (INTC_EIFR). When you need to use external interrupts, please set them according to the following process:

1. Set the external interrupt function for the corresponding external pin.
2. Write 1 to each bit of INTC_EIFCR to clear the interrupt flag and prevent malfunction.
3. Clear the INTC_EIRQCRx.EFEN bit (x=0~15) to disable the digital filter.
4. Set the INTC_EIRQCRx.EIRQTRG[1:0] bits to select the trigger edge or level; set the EISMPCLK[1:0] bits to select the digital filter sampling clock; set the EFEN bit to enable the digital filter.

10.4.3 Interrupt Event Multiplexer

The interrupt controller configures a 512-to-1 interrupt event multiplexer for each of the NIVC interrupt requests numbered 0 to 15. By writing the interrupt event number into the corresponding interrupt event selection register INTC_INTSEL0~15, one event can be arbitrarily selected from all interrupt events as the interrupt source of the interrupt request. For specific interrupt event numbers, please refer to the [Interrupt Event] section. When using NIVC interrupt requests 0 to 15, the corresponding interrupt event enable position in the Interrupt Event Enable Register INTC_INTEN can be set to 0. This can avoid the same interrupt event generating two interrupts.

10.4.4 Software Interrupt

The software interrupt function can generate an interrupt request by directly writing the software interrupt register INTC_SWIER. A total of 32 software interrupts are provided, and the corresponding NVIC interrupt request numbers are 0~31, and the permission is controlled by the interrupt request enable register INTC_IER.

10.4.5 Event Input

NIVC interrupt requests numbered 0 to 31 can also be used as event inputs (RXEV) to wake up the core (WFE).

The permission is controlled by the event enable register INTC_EVTER.

The functional block diagram of interrupt request and event input is as follows:

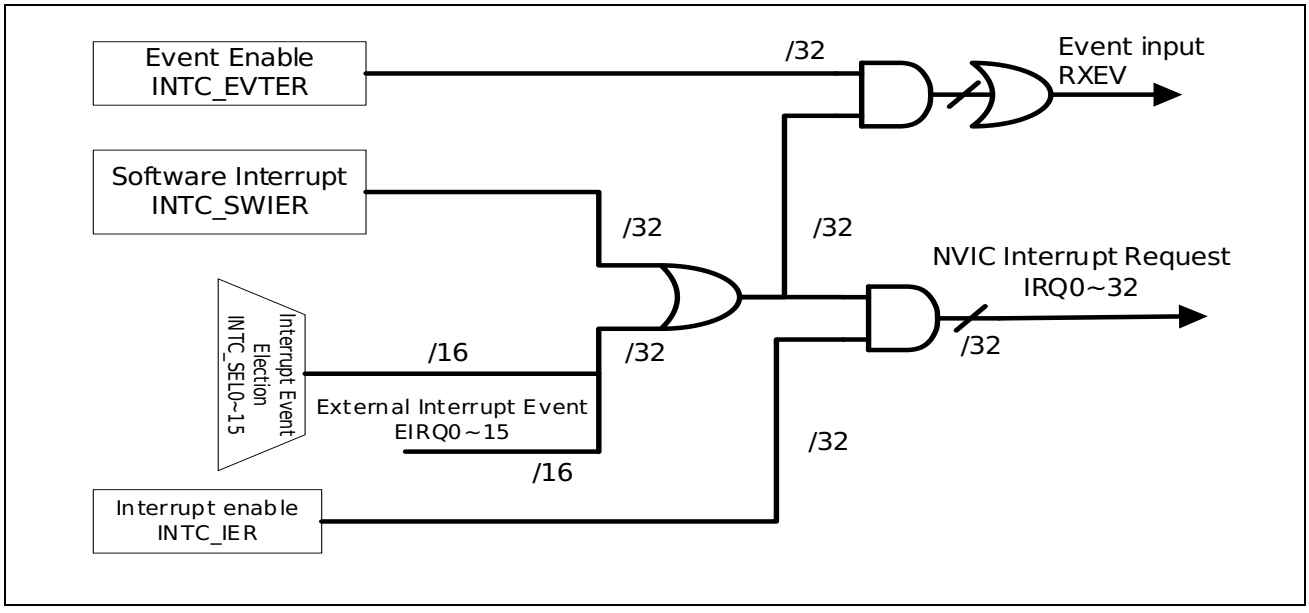


Figure 10-2 Interrupt Request/Event Input Selection

10.4.6 WFE Event Management

The interrupt controller can wake up the core (WFE) through two ways: interrupt request and event input.

- Interrupt Request Wake-up kernel

  Select an interrupt event, set INTC_INTSEL according to the selected interrupt request and enable the INT_IER register (but do not enable the interrupt request in NVIC). The INTC_EVTER register is set to disabled and the SEVONPEND bit in the Cortex-M4 core control register SCR is enabled. Executing WFE puts the core into Sleep Mode. When the selected interrupt event occurs, the core is awakened from Sleep Mode but does not enter the interrupt handler.

- Event Input Wake-up Kernel

  Select an interrupt event, set INTC_INTSEL according to the selected interrupt request and enable the INTC_EVTER register. The INTC_IER register is set to disabled. Executing WFE puts the core into Sleep Mode. When the selected interrupt event occurs, the core is woken up from Sleep Mode.

10.4.7 Noise Filter

In order to eliminate input noise, the external interrupt EIRQx (x=0~15) is configured with both digital and analog filters.

The digital filter is enabled via the EFEN bit in the INTC_EIRQCRx register. When used, PCLK3 is used as the sampling clock to sample the EIRQ input signal, and signals with a width less than 3 sampling cycles will be filtered out. The sampling period is set by the EISMPCLK[1:0] bits in the INTC_EIRQCRx register.

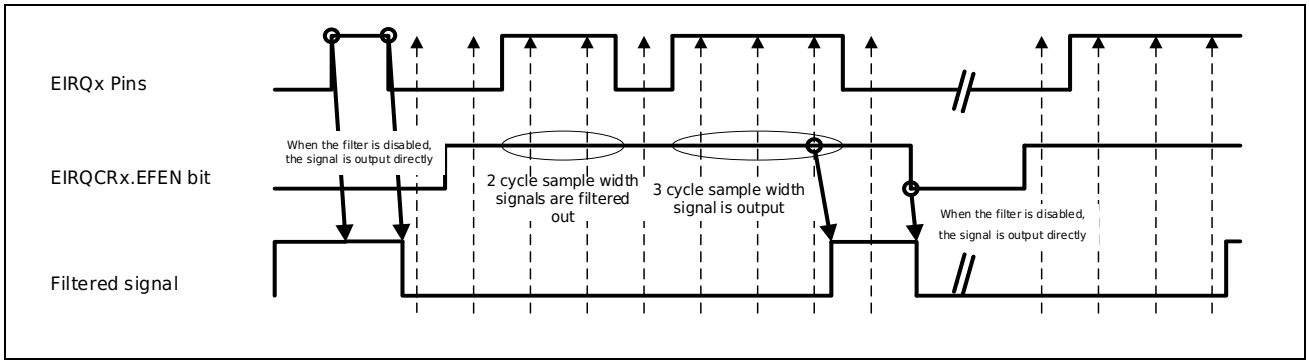


Figure 10-3 Schematic diagram of digital filter operation

The analog filter is enabled via the NOCEN bit in the INTC_EIRQCRx register. When enabled, the signal is filtered with the filter width selected by the NOCSEL bit in the INTC_EIRQCR register.

The two filters are independent of each other and can be enabled or disabled individually. When both are turned on, the signal is first filtered by analog and then by digital filtering before entering the internal circuit.

Before entering Stop Mode, the digital filter must be turned off. The digital filter can be enabled again after returning from Stop Mode. The analog filter can be operated in Stop Mode.

10.4.8 Low Power Mode Return

10.4.8.1 Dormant Mode Return

When waking up from Sleep Mode via an interrupt request, the following settings are required:

- Select an interrupt event as an interrupt request
  If you use interrupt requests 0 to 31, you need to set the INTC_IER register.
- Enable this interrupt request in NVIC
- When using non-maskable interrupts, you need to set the INTC_NMIER register

10.4.8.2 Stop Mode return

To wake up from Stop Mode by an interrupt request, the following settings are required:

- Select an interrupt event as an interrupt request
  If you use interrupt requests 0 to 31, you need to set the INTC_IER register.
- Enable this interrupt event through the INTC_WKEN register
  If non-maskable interrupts are used, the interrupt events are enabled through the INTC_NMIER register. The non-maskable interrupts that can wake up the Stop Mode are: detection of main oscillator stop interrupt and SWDT underflow/refresh interrupt.
- Enable this interrupt request in NVIC

10.4.8.3 Power-Down Mode Return

The return from Power-Down mode can be returned by the conditions described in the Power Control (PWC) section, RES# pin reset, power-on reset, and Brown-out reset conditions. After returning, the CPU enters the reset interrupt processing. For details, please refer to the [Power Control (PWC)]section.

10.4.8.4 Non-maskable Interrupt and WFI Instruction

Before executing the WFI instruction, please confirm that all status bits of the non-maskable interrupt flag register INTC_NMIFR are 0.

10.5Register Description

Table 10-4 INTC Base Address

Name	Base Address	Description
INTC	0x4005 1000	INTC Base Address

Table 10-5 INTC Register List

Register	Description	Offset Address	Bit Width	Reset Value
INTC_NMIER	Nomaskable interrupt enable register	0x0004	32	0x0000 0000
INTC_NMIFR	Nomaskable interrupt flag register	0x0008	32	0x0000 0000
INTC_NMIFCR	Nomaskable interrupt flag clear register	0x000C	32	0x0000 0000
INTC_EIRQCR0	External Interrupt Control Register 0	0x0010	32	0x0000 0000
INTC_EIRQCR1	External Interrupt Control Register 1	0x0014	32	0x0000 0000
INTC_EIRQCR2	External Interrupt Control Register 2	0x0018	32	0x0000 0000
INTC_EIRQCR3	External Interrupt Control Register 3	0x001C	32	0x0000 0000
INTC_EIRQCR4	External Interrupt Control Register 4	0x0020	32	0x0000 0000
INTC_EIRQCR5	External Interrupt Control Register 5	0x0024	32	0x0000 0000
INTC_EIRQCR6	External Interrupt Control Register 6	0x0028	32	0x0000 0000
INTC_EIRQCR7	External Interrupt Control Register 7	0x002C	32	0x0000 0000
INTC_EIRQCR8	External Interrupt Control Register 8	0x0030	32	0x0000 0000
INTC_EIRQCR9	External Interrupt Control Register 9	0x0034	32	0x0000 0000
INTC_EIRQCR10	External Interrupt Control Register 10	0x0038	32	0x0000 0000
INTC_EIRQCR11	External Interrupt Control Register 11	0x003C	32	0x0000 0000
INTC_EIRQCR12	External Interrupt Control Register 12	0x0040	32	0x0000 0000
INTC_EIRQCR13	External Interrupt Control Register 13	0x0044	32	0x0000 0000
INTC_EIRQCR14	External Interrupt Control Register 14	0x0048	32	0x0000 0000
INTC_EIRQCR15	External Interrupt Control Register 15	0x004C	32	0x0000 0000
INTC_WKEN	Stop Mode Wakeup Event Enable Register	0x0050	32	0x0000 0000
INTC_EIFR	External Interrupt Flag Register	0x0054	32	0x0000 0000
INTC_EIFCR	External Interrupt Flag Clear Register	0x0058	32	0x0000 0000
INTC_INTSEL0	Interrupt Event Select Register 0	0x005C	32	0x0000 01FF
INTC_INTSEL1	Interrupt Event Select Register 1	0x0060	32	0x0000 01FF
INTC_INTSEL2	Interrupt Event Select Register 2	0x0064	32	0x0000 01FF

Register	Description	Offset Address	Bit Width	Reset Value
INTC_INTSEL3	Interrupt Event Select Register 3	0x0068	32	0x0000 01FF
INTC_INTSEL4	Interrupt Event Select Register 4	0x006C	32	0x0000 01FF
INTC_INTSEL5	Interrupt Event Select Register 5	0x0070	32	0x0000 01FF
INTC_INTSEL6	Interrupt Event Select Register 6	0x0074	32	0x0000 01FF
INTC_INTSEL7	Interrupt Event Select Register 7	0x0078	32	0x0000 01FF
INTC_INTSEL8	Interrupt Event Select Register 8	0x007C	32	0x0000 01FF
INTC_INTSEL9	Interrupt Event Select Register 9	0x0080	32	0x0000 01FF
INTC_INTSEL10	Interrupt Event Select Register 10	0x0084	32	0x0000 01FF
INTC_INTSEL11	Interrupt Event Select Register 11	0x0088	32	0x0000 01FF
INTC_INTSEL12	Interrupt Event Select Register 12	0x008C	32	0x0000 01FF
INTC_INTSEL13	Interrupt Event Select Register 13	0x0090	32	0x0000 01FF
INTC_INTSEL14	Interrupt Event Select Register 14	0x0094	32	0x0000 01FF
INTC_INTSEL15	Interrupt Event Select Register 15	0x0098	32	0x0000 01FF
INTC_INTEN0	Interrupt Event Enable Register 0	0x025C	32	0xFFFF FFFF
INTC_INTEN1	Interrupt Event Enable Register 1	0x0260	32	0xFFFF FFFF
INTC_INTEN2	Interrupt Event Enable Register 2	0x0264	32	0xFFFF FFFF
INTC_INTEN3	Interrupt Event Enable Register 3	0x0268	32	0xFFFF FFFF
INTC_INTEN4	Interrupt Event Enable Register 4	0x026C	32	0xFFFF FFFF
INTC_INTEN5	Interrupt Event Enable Register 5	0x0270	32	0xFFFF FFFF
INTC_INTEN6	Interrupt Event Enable Register 6	0x0274	32	0xFFFF FFFF
INTC_INTEN7	Interrupt Event Enable Register 7	0x0278	32	0xFFFF FFFF
INTC_INTEN8	Interrupt Event Enable Register 8	0x027C	32	0xFFFF FFFF
INTC_INTEN9	Interrupt Event Enable Register 9	0x0280	32	0xFFFF FFFF
INTC_INTEN10	Interrupt Event Enable Register 10	0x0284	32	0xFFFF FFFF
INTC_INTEN11	Interrupt Event Enable Register 11	0x0288	32	0xFFFF FFFF
INTC_INTEN12	Interrupt Event Enable Register 12	0x028C	32	0xFFFF FFFF
INTC_INTEN13	Interrupt Event Enable Register 13	0x0290	32	0xFFFF FFFF
INTC_INTEN14	Interrupt Event Enable Register 14	0x0294	32	0xFFFF FFFF
INTC_INTEN15	Interrupt Event Enable Register 15	0x0298	32	0xFFFF FFFF
INTC_SWIER	Software Interrupt Register	0x029C	32	0x0000 0000
INTC_EVTER	Event Enable Register	0x02A0	32	0x0000 0000
INTC_IER	Interrupt Request Enable	0x02A4	32	0xFFFF FFFF

Register	Description	Offset Address	Bit Width	Reset Value
	Register			
INTC_FPUIER	Interrupt Request Enable Register	0x02A8	32	0x0000 0000

10.5.1 Nomaskable Interrupt Enable Register (INTC _ NMIER)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res				WDTEN	BUSERRENN	RECCERREN	RPARERRENN	Res		XTALSTPEN	Res	PVD2EN	PVD1EN	SWDTEN	Res

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b12	Res	Reserved Bit	Read at “0”, write at “0”	RW
b11	WDTEN	WDT underflow/refresh interrupt selection	0: Disable interrupt as a nomaskable interrupt source 1: Choose interrupt as nomaskable interrupt source	RW
b10	BUSERREN	MPU main bus error interrupt selection	0: Disable interrupt as a nomaskable interrupt source 1: Choose interrupt as nomaskable interrupt source	RW
b9	RECCERREN	SRAM ECC check error interrupt selection	0: Disable interrupt as a nomaskable interrupt source 1: Choose interrupt as nomaskable interrupt source	RW
b8	RPARERREN	Cache RAM parity error interrupt selection	0: Disable interrupt as a nomaskable interrupt source 1: Choose interrupt as nomaskable interrupt source	RW
b7~b6	Res	Reserved Bit	Read at “0”, write at “0”	RW
b5	XATLSTPEN	Detect Master Oscillator Stop Interrupt Selection	0: Disable interrupt as a nomaskable interrupt source 1: Choose interrupt as nomaskable interrupt source	RW
b4	Res	Reserved Bit	Read at “0”, write at “0”	RW
b3	PVD2EN	Low voltage detection PVD2 interrupt selection	0: Disable interrupt as a nomaskable interrupt source 1: Choose interrupt as nomaskable interrupt source	RW
b2	PVD1EN	Low voltage detection PVD1 interrupt selection	0: Disable interrupt as a nomaskable interrupt source 1: Choose interrupt as nomaskable interrupt source	RW
b1	SWDTEN	SWDT underflow/refresh interrupt selection	0: Disable interrupt as a nomaskable interrupt source 1: Choose interrupt as nomaskable interrupt source	RW
b0	Res	Reserved Bit	Read at “0”, write at “0”	RW

10.5.2 Nomaskable Interrupt Flag Register (INTC _ NMIFR)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res				WDTF	BUSERRF	RECCERRF	RPARERRF	Res		XTALSTPF	Res	PVD2F	PVD1F	SWDTF	Res

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b12	Res	Reserved Bit	Read at “0”, write at “0”	RW
b11	WDTF	WDT underflow/refresh interrupt flag	0: No WDT underflow/refresh application occurs 1: WDT underflow/refresh request occurs	R
b10	BUSERRF	MPU main bus error interrupt flag	0: No MPU master bus error application occurs 1: MPU main bus error application occurs	R
b9	RECCERRF	SRAM DED parity error interrupt flag	0: SRAM DED verification error application does not occur 1: SRAM DED verification error application occurs	R
b8	RPARERRF	SRAM parity error interrupt flag	0: No SRAM parity error application occurs 1: SRAM parity error application occurs	R
b7~b6	Res	Reserved Bit	Read at “0”, write at “0”	RW
b5	XTALSTPF	Detect main oscillator stop interrupt flag	0: No detection occurs Main oscillator stop application 1: Occurrence detection main oscillator stop application	R
b4	Res	Reserved Bit	Read at “0”, write at “0”	RW
b3	PVD2F	Low voltage detection PVD2 interrupt flag	0: No low voltage detection occurs PVD2 application 1: PVD2 application for low voltage detection	R
b2	PVD1F	Low voltage detection PVD1 interrupt flag	0: No low voltage detection occurs PVD1 application 1: PVD1 application for low voltage detection	R
b1	SWDTF	SWDT underflow/refresh interrupt flag	0: No SWDT underflow/refresh request occurs 1: SWDT underflow/refresh request occurs	R
b0	Res	Reserved Bit	Read at “0”, write at “0”	RW

10.5.3 Non-Maskable Interrupt Flag Clear Register (INTC_NMIFCR)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res				WDTFCLR	BUSERRFCLR	RECCERRFCLR	RPARERRFCLR	Res		XTALSTPFCLR	Res	PVD2FCLR	PVD1FCLR	SWDTFCLR	Res

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b12	Res	Reserved Bit	Read at “0”, write at “0”	RW
b11	WDTFCLR	WDT underflow/refresh interrupt flag clear	0: Invalid 1: Clear WDT underflow/refresh flag	RW
b10	BUSERRFCLR	MPU main bus error interrupt flag cleared	0: Invalid 1: Clear the MPU main bus error flag	RW
b9	RECCERRFCLR	SRAM DED parity error interrupt flag clear	0: Invalid 1: Clear the SRAM DED verification error flag	RW
b8	RPARERRFCLR	SRAM parity error interrupt flag cleared	0: Invalid 1: Clear the SRAM parity error flag	RW
b7~b6	Res	Reserved Bit	Read at “0”, write at “0”	RW
b5	XTALSTPFCLR	Detect main oscillator stop interrupt flag clear	0: Invalid 1: Clear the detection of the stop flag of the main oscillator	RW
b4	Res	Reserved Bit	Read at “0”, write at “0”	RW
b3	PVD2FCLR	Low voltage detection PVD2 interrupt flag clear	0: Invalid 1: Clear low voltage detection PVD2 flag	RW
b2	PVD1FCLR	Low voltage detection PVD1 interrupt flag clear	0: Invalid 1: Clear low voltage detection PVD1 flag	RW
b1	SWDTFCLR	SWDT underflow/refresh interrupt flag cleared	0: Invalid 1: Clear SWDT underflow/refresh flag	RW
b0	Res	Reserved Bit	Read at “0”, write at “0”	RW

Note:

This register is read as 0.

10.5.4 External Interrupt Control Register (INTC_EIRQCRx, x=0~15)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
NOCEN	Res	NOCSEL[1:0]		Res				EFEN	Res	EISMPCLK[1:0]		Res		EIRQTRG[1:0]	

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b16	Res	Reserved Bit	Read at “0”, write at “0”	RW
b15	NOCEN	EIRQ analog filter enable	0: Disable Analog Filter Function 1: Enable Analog Filter Function The filter width is determined by INTC_NOCCR.NOCSEL.	RW
b14	Res	Reserved Bit	Read at “0”, write at “0”	RW
b13~b12	NOCSEL[1:0]	Analog filter filter width selection	00: Filter Width Level 1 01: Filter Width Level 2 10: Filter Width Level 3 11: Filter Width Level 4 For specific values of each gear, please refer to the EIRQ Filter features section in the **Electrical Features of Data Sheet**.	RW
b11~b8	Res	Reserved Bit	Read at “0”, write at “0”	RW
b7	EFEN	EIRQ digital filter enable	0: Disable digital filter function 1: Allow digital filter function	RW
b6	Res	Reserved Bit	Read at “0”, write at “0”	RW
b5~b4	EISMPCLK[1:0]	Digital filter sampling period selection	00: 1*PCLK3 Cycle 01: 8*PCLK3 Cycles 10: 32*PCLK3 Cycles 11: 64*PCLK3 Cycles	RW
b3~b2	Res	Reserved Bit	Read at “0”, write at “0”	RW
b1~b0	EIRQTRG[1:0]	Trigger selection	00: Falling edge 01: Rising edge 10: Double edge 11: low level	RW

10.5.5 External Interrupt Flag Register (INTC_EIFR)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
EIF[15:0]															

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b16	Res	Reserved Bit	Read at “0”, write at “0”	RW
b15~b0	EIF	EIRQ flag	0: EIRQ event did not occur, or write EIFCR bit to clear the bit 1: The selected EIRQ event occurs	R

10.5.6 External Interrupt Flag Clear Register (INTC_EIFCR)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
EIFCLR[15:0]															

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b16	Res	Reserved Bit	Read at “0”, write at “0”	RW
b15~b0	EIFCLR	EIRQ Flag Clear Bit	0: Writing "0" is invalid 1: Write "1" to clear the EIFR register	RW

10.5.7 Interrupt Event Selection Register (INTC_INTSEL0~15)

Reset value: 0x0000 01FF

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res							INTSEL[8:0]								

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b9	Res	Reserved Bit	Read at “0”, write at “0”	RW
b8~b0	INTSEL[8:0]	Interrupt Event Request Selection	0x000~0x1FE: [Interrupt Event List] Corresponding Event	RW

10.5.8 Interrupt Event Enable Register (INTC_INTEN0~15)

Reset value: 0xFFFF FFFF

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
INTEN[x*32+31:x*32+16]															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
INTEN[x*32+15:x*32]															

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b0	INTEN [x*32+31:x*32] x=0~15	Interrupt Event Enable	INTEN[511:0] corresponds to the interrupt event number one by one. An interrupt event set to "0" will not be received by the NVIC interrupt request numbered 15 or later. 0: Interrupt event invalid 1: Interrupt event valid The bit corresponding to the sequence number of the unconfigured interrupt event is a read-only bit and is "1" when read out.	RW

10.5.9 Stop Mode Wake-up Event Enable Register (INTC_WKEN)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res	CMP3WKEN	CMP2WKEN	Res		RXWKEN	Res		TMR0CMPWKEN	RTCPRDWKEN	RTCALMWKEN	WKTMWKEN	CMP1WKEN	Res		SWDTWKEN

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
EIRQWKEN[15:0]															

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31	Res	Reserved Bit	Read at “0”, write at “0”	RW
b30	CMP3WKEN	CMP3 Stop Mode wake-up enable	0: Arousal prohibition 1: Arousal permit	RW
b29	CMP2WKEN	CMP2 Stop Mode wake-up enable	0: Arousal prohibition 1: Arousal permit	RW
b28~b27	Res	Reserved Bit	Read at “0”, write at “0”	RW
b26	RXWKEN	USART_1 RX line Stop Mode wake-up enable	0: Arousal prohibition 1: Arousal permit	RW
b25~b24	Res	Reserved Bit	Read at “0”, write at “0”	RW
b23	TMR0CMPWKEN	TMR0_1_CMPA Stop Mode wake-up enable	0: Arousal prohibition 1: Arousal permit	RW
b22	RTCPRDWKEN	RTC_PRD Stop Mode wakeup enable	0: Arousal prohibition 1: Arousal permit	RW
b21	RTCALMWKEN	RTC_ALM Stop Mode wake-up enable	0: Arousal prohibition 1: Arousal permit	RW
b20	WKTMWKEN	WKTM_CMP Stop Mode wake-up enable	0: Arousal prohibition 1: Arousal permit	RW
b19	CMP1WKEN	CMP1 Stop Mode wake-up enable	0: Arousal prohibition 1: Arousal permit	RW
b18~b17	Res	Reserved Bit	Read at “0”, write at “0”	RW
b16	SWDTWKEN	SWDT_REFUDF Stop Mode wakeup enable	0: Arousal prohibition 1: Arousal permit	RW
b15~b0	EIRQWKEN[15:0]	EIRQ Stop Mode wakeup enable	0: Arousal prohibition 1: Arousal permit	RW

10.5.10 Software Interrupt Register (INTC_SWIER)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
SWIE[31:16]															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
SWIE[15:0]															

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b0	SWIE	Software Interrupt	0: Invalid 1: Software interrupt occurs Note: A software interrupt occurs after writing "1" from "0". If it happens again, you must write "0" to clear it.	RW

10.5.11 Event Enable Register (INTC_EVTER)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
EVTE[31:16]															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
EVTE[15:0]															

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b0	EVTE	Event Enable	0: Event selection disabled 1: Event selection permission	RW

10.5.12 Interrupt Request Enable Register (INTC_IER)

Reset value: 0xFFFF FFFF

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
IEN[31:0]															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
IEN[15:0]															

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b0	IEN	Interrupt Request Enable	Register bits 0 to 31 correspond to NVIC interrupt requests 0 to 31 respectively. When disabled, the interrupt events corresponding to each interrupt request will not be received by NVIC. 0: Interrupt Disable 1: Interrupt Permission	RW

10.5.13 FPU Interrupt Enable Register (INTC_FPUIER)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res										IDCIEN	IXCIEN	UFCIEN	OFCIEN	DZCIEN	IOCIEN

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b6	Res	Reserved Bit	Read at “0”, write at “0”	RW
b5	IDCIEN	FPU IDCI Interrupt Enable Bit	0: FPU IDCI Interrupt Disabled 1: FPU IDCI Interrupt Enabled	RW
b4	IXCIEN	FPU IXCI Interrupt Enable Bit	0: FPU IXCI Interrupt Disabled 1: FPU IXCI Interrupt Enabled	RW
b3	UFCIEN	FPU UFCI Interrupt Enable Bit	0: FPU UFCI Interrupt Disabled 1: FPU UFCI Interrupt Enabled	RW
b2	OFCIEN	FPU OFCI Interrupt Enable Bit	0: FPU OFCI Interrupt Disabled 1: FPU OFCI Interrupt Enabled	RW
b1	DZCIEN	FPU DZCI Interrupt Enable Bit	0: FPU DZCI Interrupt Disabled 1: FPU DZCI Interrupt Enabled	RW
b0	IOCIEN	FPU IOCI Interrupt Enable Bit	0: FPU IOCI Interrupt Disabled 1: FPU IOCI Interrupt Enabled	RW

10.6 Precautions

For a description of ARM core interrupts, please refer to the ARM manual ARM Processor Cortex®-M4 Technical Reference Manual (ARM DDI 0439D).

11 Automatic Operation System (AOS)

11.1 Overview

The automatic operation system is used to realize the linkage between peripheral hardware circuits without the help of the CPU. Use the events generated by peripherals as AOS sources, such as timer compare match, timer count overflow, RTC periodic signal, various states of communication module receiving and sending data (idle, receiving data full, sending data end, sending data empty), ADC conversion end and other events to trigger other peripheral actions. The triggered peripheral action is called AOS Target.

11.1.1 Function Overview

AOS has multiple trigger sources, see the [AOS Source Events] section for details. Except for special restrictions, each AOS target can select one or more of them as trigger sources. In addition, two common trigger sources can be selected through the common trigger source selection register 1 and the common trigger source selection register 2. All AOS targets share these two common trigger sources. Thus, for an AOS target, up to three trigger sources can be selected, and any one of them can trigger the AOS target action when a trigger event occurs.

It can be triggered by the hardware of the peripheral circuit, or it can be triggered by the software by writing the register.

The peripherals and quantities that can be targeted by AOS are as follows:

- 9 DMA trigger targets
- 4 advanced control timers (Timer6) trigger targets
- 4 high-resolution PWM (HRPWM) trigger targets
- 1 Universal-purpose control timer 4 (Timer4) trigger target
- 1 universal-purpose timer 0 (Timer0) trigger target
- 4 universal-purpose timer A (TimerA) trigger targets
- 2 Event Port trigger targets
- 3 groups of 2 ADC trigger targets in each group, used for ADC1~ADC3 sequence trigger

11.2Functional Description

11.2.1 AOS Block Diagram

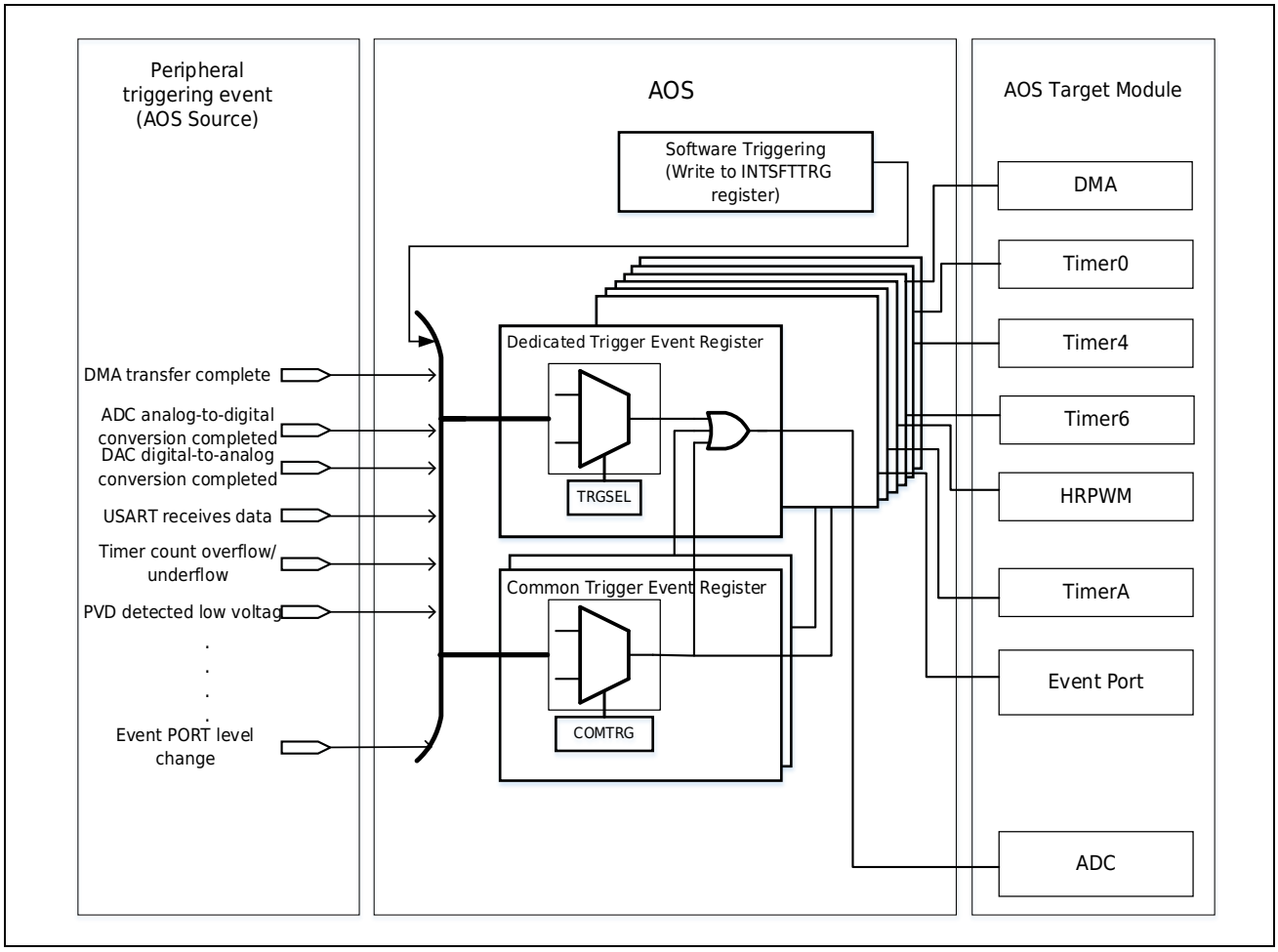


Figure 11-1 AOS Module Block Diagram

11.2.2 AOS Source Eventsand Targets

The AOS source event numbers are shown in [Interrupt Event List]. The events marked with "✓" in the "Trigger Source" column in the table can be used as AOS sources. The AOS targets are shown inTable 11-1.

Table 11-1 AOS Target List

Modules	SN	Action
DMA	0	Trigger channel 0 to start transmission
	1	Trigger channel 1 to start transmission
	2	Trigger channel 2 to start transmission
	3	Trigger channel 3 to start transmission
	4	Trigger channel 4 to start transmission
	5	Trigger channel 5 to start transmission
	6	Trigger channel 6 to start transmission
	7	Trigger channel 7 to start transmission
DMA	8	Trigger Event Triggers Channel Reset
Timer6	9	Trigger counter start/stop/clear/refresh/capture/increment/decrement
	10	Trigger counter start/stop/clear/refresh/capture/increment/decrement
	11	Trigger counter start/stop/clear/refresh/capture/increment/decrement
	12	Trigger counter start/stop/clear/refresh/capture/increment/decrement
HRPWM	13	Trigger counter start/clear/capture
	14	Trigger counter start/clear/capture
	15	Trigger counter start/clear/capture
	16	Trigger counter start/clear/capture
Timer4	17	Trigger counter start/stop/clear/capture
Event Port	18	Triggering Event Port1/2 Action
	19	Trigger Event Port3/4 Action
Timer0	20	Trigger counter start/stop/clear/capture
TimerA	21	Trigger counter unit 1/5 start/stop/clear/increment/decrement
		Trigger counter unit 2 input capture
	22	Trigger counter unit 2 start/stop/clear/increment/decrement
		Trigger counter unit 1/5 input capture
	23	Trigger counter unit 3 start/stop/clear/increment/decrement
		Trigger Counter Unit 4 Input Capture
	24	Trigger counter unit 4 start/stop/clear/increment/decrement
		Trigger Counter Unit 3 Input Capture
ADC_1	25，26	Trigger sequence A/B to start analog-to-digital conversion

Modules	SN	Action
ADC_2	27，28	Trigger sequence A/B to start analog-to-digital conversion
ADC_3	29，30	Trigger sequence A/B to start analog-to-digital conversion

11.2.3 Dedicated Trigger Source

The peripheral circuit module with AOS target is equipped with a dedicated peripheral trigger source selection register for each AOS target. When this register is written into the event number corresponding to the AOS source, the AOS target will select this AOS source as the trigger source. When an event from the AOS source occurs, the event will be delivered to the AOS target through the AOS, and the peripheral circuit as the AOS target starts to act according to its own settings.

11.2.4 Common Trigger Source

In addition to the dedicated peripheral trigger source selection registers for each AOS target, AOS also configures two common trigger source selection registers (AOS_COMTRGSEL1, AOS_COMTRGSEL2). It is used to realize the function of triggering one or more AOS targets with the same trigger source, and it can also realize the function of multiple AOS trigger sources triggering the same AOS target. When using, first enable the common trigger source enable position in the AOS target-specific peripheral trigger source selection register, and then write the event number corresponding to the AOS source in the common trigger source selection register. When an event of the AOS source occurs, the event will be passed to the AOS target through the common trigger source of the AOS, and the peripheral circuit as the AOS target starts to act according to its own setting. When the dedicated trigger source and public trigger source are set at the same time, a maximum of 3 AOS sources can trigger the same AOS target at the same time, and when any trigger event occurs in any of the 3 AOS sources, the AOS target will be triggered.

All AOS targets share these two common trigger sources. Therefore, when other AOS targets do not use the event selected by the public trigger source selection register, it is necessary to disable the public trigger source enable position in its dedicated peripheral trigger source selection register to prevent wrong trigger actions.

11.3Register Description

Table 11-2 AOS Base Address

Name	Base Address	Description
AOS	0x4001 0800	AOS Base Address

Table 11-3 AOS Register List

Register	Description	Offset Address	Bit Width	Reset Value
INTSFTTRG	Peripheral Trigger Event Register	0x00	32	0x0000 0000
DMA_TRGSELx (x=0~7)	DMA Transfer Start Trigger Source Selection Register	0x04+0x04*x	32	0x0000 01FF
DMA_RC_TRGSEL	DMA channel reset trigger source select register	0x24	32	0x0000 01FF
TMR6_TRGSELx (x=0~3)	Timer6 hardware trigger event selection register	0x28+0x04*x	32	0x0000 01FF
HRPWM_TRGSELx (x=0~3)	HRPWM Hardware Trigger Event Selection Register	0x38+0x04*x	32	0x0000 01FF
TMR4_TRGSEL	Timer4 hardware trigger event selection register	0x48	32	0x0000 01FF
PEVNT_TRGSEL12	Event Port1/2 trigger source selection register	0x4C	32	0x0000 01FF
PEVNT_TRGSEL34	Event Port3/4 trigger source selection register	0x50	32	0x0000 01FF
TMR0_TRGSEL	Timer0 hardware trigger event selection register	0x54	32	0x0000 01FF
TMRA_TRGSELx (x=0~3)	TimerA Hardware Trigger Event Selection Register	0x58+0x04*x	32	0x0000 01FF
ADC1_TRGSELx (x=0~1)	ADC1 Conversion Start On-chip Trigger Source Selection Register	0x68+0x04*x	32	0x0000 01FF
ADC2_TRGSELx (x=0~1)	ADC2 Conversion Start On-chip Trigger Source Selection Register	0x70+0x04*x	32	0x0000 01FF
ADC3_TRGSELx (x=0~1)	ADC3 Conversion Start On-chip Trigger Source Selection Register	0x78+0x04*x	32	0x0000 01FF
AOS_COMTRGSELx (x=1~2)	Common trigger source selection register	0x80+0x04*x	32	0x0000 01FF

11.3.1 Peripheral Trigger Event Register (INTSFTTRG)

Each time a 1 is written to this register, a trigger event will be generated.

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res															STRG

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b1	Res	Reserved Bit	Read at “0”, write at “0”	RW
b0	STRG	Software triggering	0: No software trigger event is generated 1: Generate a software trigger event Set this bit to 1 to generate a peripheral trigger event, writing 0 by software is invalid	W

11.3.2 DMA Transfer Start Trigger Source Selection Register (DMA_TRGSELx, x=0~7)

After DMA is enabled, use this register to select the trigger event source. When a peripheral event corresponding to the number occurs, DMA will be triggered by the event to start the transmission.

Each data transmission channel corresponds to a register.

Reset value: 0x0000 01FF

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
COMEN[1:0]		Res													
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res							TRGSEL[8:0]								

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31	COMEN[1]	Common Trigger Enable	0: Disable the common trigger event of AOS_COMTRGSEL2 from triggering this AOS target 1: Allow the common trigger event of AOS_COMTRGSEL2 to trigger this AOS target	RW
b30	COMEN[0]	Common Trigger Enable	0: Disable the common trigger event of AOS_COMTRGSEL1 from triggering this AOS target 1: Allow the common trigger event of AOS_COMTRGSEL1 to trigger this AOS target	RW
b29~b9	Res	Reserved Bit	Read at “0”, write at “0”	RW
b8~b0	TRGSEL[8:0]	Trigger Source Selection	Write the number of the peripheral circuit event to be selected	RW

11.3.3 DMA Channel Reset Trigger Source Selection Register (DMA_RC_TRGSEL)

After DMA is enabled, use this register to select the trigger event source and write the trigger event number into the TRGSEL[8:0] bits of this register. When a peripheral event corresponding to the number occurs, DMA will be triggered by the event to reset the channel.

Reset value: 0x0000 01FF

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
COMEN[1:0]		Res													
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res							TRGSEL[8:0]								

Bit/Bitfield	Marking	Bit Name		Function	Read and Write
b31	COMEN[1]	Common Enable	Trigger	0: Disable the common trigger event of AOS_COMTRGSEL2 from triggering this AOS target 1: Allow the common trigger event of AOS_COMTRGSEL2 to trigger this AOS target	RW
b30	COMEN[0]	Common Enable	Trigger	0: Disable the common trigger event of AOS_COMTRGSEL1 from triggering this AOS target 1: Allow the common trigger event of AOS_COMTRGSEL1 to trigger this AOS target	RW
b29~b9	Res	Reserved Bit		Read at “0”, write at “0”	RW
b8~b0	TRGSEL[8:0]	Trigger Selection	Source	Write the number of the peripheral circuit event to be selected	RW

11.3.4 Timer6 Hardware Trigger Event Selection Register (TMR6_TRGSELx) (x=0~3)

After selecting internal trigger hardware synchronization in Timer6, please use this register to select the trigger event source and write the trigger event number into the TRGSEL[8:0] bits of this register. When a peripheral circuit event corresponding to the number occurs, Timer6 will be triggered by the event to perform hardware synchronization (including start/stop/clear/refresh/increment/decrement/input capture).

All Timer6 units share these 4 registers.

Reset value: 0x0000 01FF

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
COMEN[1:0]		Res													
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res							TRGSEL[8:0]								

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31	COMEN[1]	Common Trigger Enable	0: Disable the common trigger event of AOS_COMTRGSEL2 from triggering this AOS target 1: Allow the common trigger event of AOS_COMTRGSEL2 to trigger this AOS target	RW
b30	COMEN[0]	Common Trigger Enable	0: Disable the common trigger event of AOS_COMTRGSEL1 from triggering this AOS target 1: Allow the common trigger event of AOS_COMTRGSEL1 to trigger this AOS target	RW
b29~b9	Res	Reserved Bit	Read at “0”, write at “0”	RW
b8~b0	TRGSEL[8:0]	Trigger Source Selection	Write the number of the peripheral circuit event to be selected	RW

11.3.5 HRPWM Hardware Trigger Event Selection Register (HRPWM_TRGSELx, x=0~3)

After HRPWM selects internal trigger hardware synchronization, please use this register to select the trigger event source and write the trigger event number into the TRGSEL[8:0] bits of this register. When a peripheral circuit event corresponding to the number occurs, HRPWM will be triggered by the event to perform hardware synchronization (including start/stop/clear/refresh/increment/decrement/input capture).

All HRPWM units share these 4 registers.

Reset value: 0x0000 01FF

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
COMEN[1:0]		Res													
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res							TRGSEL[8:0]								

Bit/Bitfield	Marking	Bit Name		Function	Read and Write
b31	COMEN[1]	Common Trigger Enable		0: Disable the common trigger event of AOS_COMTRGSEL2 from triggering this AOS target 1: Allow the common trigger event of AOS_COMTRGSEL2 to trigger this AOS target	RW
b30	COMEN[0]	Common Trigger Enable		0: Disable the common trigger event of AOS_COMTRGSEL1 from triggering this AOS target 1: Allow the common trigger event of AOS_COMTRGSEL1 to trigger this AOS target	RW
b29~b9	Res	Reserved Bit		Read at “0”, write at “0”	RW
b8~b0	TRGSEL[8:0]	Trigger Source Selection		Write the number of the peripheral circuit event to be selected	RW

11.3.6 Timer4 Hardware Trigger Event Selection Register (TMR4_TRGSEL)

After Timer4 selects internal trigger hardware synchronization, please use this register to select the trigger event source and write the trigger event number into the TRGSEL[8:0] bits of this register. When the peripheral circuit event corresponding to the number occurs, Timer4 will be triggered by the event to start hardware synchronization.

Reset value: 0x0000 01FF

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
COMEN[1:0]		Res													
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res							TRGSEL[8:0]								

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31	COMEN[1]	Common Trigger Enable	0: Disable the common trigger event of AOS_COMTRGSEL2 from triggering this AOS target 1: Allow the common trigger event of AOS_COMTRGSEL2 to trigger this AOS target	RW
b30	COMEN[0]	Common Trigger Enable	0: Disable the common trigger event of AOS_COMTRGSEL1 from triggering this AOS target 1: Allow the common trigger event of AOS_COMTRGSEL1 to trigger this AOS target	RW
b29~b9	Res	Reserved Bit	Read at “0”, write at “0”	RW
b8~b0	TRGSEL[8:0]	Trigger Source Selection	Write the number of the peripheral circuit event to be selected	RW

11.3.7 Event Port Trigger Source Select Register (PEVNT_TRGSEL12, PEVNT_TRGSEL34)

Write the trigger event number into this register. Use this register to select the trigger event source. Write the number of the trigger event into the TRGSEL[8:0] bits of this register. Event Port12 shares the PEVNT_TRGSEL12 register, and EventPort34 shares the PEVNT_TRGSEL34 register.

Reset value: 0x0000 01FF

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
COMEN[1:0]		Res													
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res							TRGSEL[8:0]								

Bit/Bitfield	Marking	Bit Name		Function	Read and Write
b31	COMEN[1]	Common Enable	Trigger	0: Disable the common trigger event of AOS_COMTRGSEL2 from triggering this AOS target 1: Allow the common trigger event of AOS_COMTRGSEL2 to trigger this AOS target	RW
b30	COMEN[0]	Common Enable	Trigger	0: Disable the common trigger event of AOS_COMTRGSEL1 from triggering this AOS target 1: Allow the common trigger event of AOS_COMTRGSEL1 to trigger this AOS target	RW
b29~b9	Res	Reserved Bit		Read at “0”, write at “0”	RW
b8~b0	TRGSEL[8:0]	Trigger Selection	Source	Write the number of the peripheral circuit event to be selected	RW

11.3.8 Timer0 Hardware Trigger Event Selection Register (TMR0_TRGSEL)

After selecting hardware trigger in Timer0, please use this register to select the trigger event source and write the trigger event number into the TRGSEL[8:0] bits of this register. When a peripheral event corresponding to the number occurs, Timer0 will be triggered by the event to take actions (including start/stop/clear/input capture).

All Timer0 units and their channels share this register.

Reset value: 0x0000 01FF

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
COMEN[1:0]		Res													
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res							TRGSEL[8:0]								

Bit/Bitfield	Marking	Bit Name		Function	Read and Write
b31	COMEN[1]	Common Enable	Trigger	0: Disable the common trigger event of AOS_COMTRGSEL2 from triggering this AOS target 1: Allow the common trigger event of AOS_COMTRGSEL2 to trigger this AOS target	RW
b30	COMEN[0]	Common Enable	Trigger	0: Disable the common trigger event of AOS_COMTRGSEL1 from triggering this AOS target 1: Allow the common trigger event of AOS_COMTRGSEL1 to trigger this AOS target	RW
b29~b9	Res	Reserved Bit		Read at “0”, write at “0”	RW
b8~b0	TRGSEL[8:0]	Trigger Selection	Source	Write the number of the peripheral circuit event to be selected	RW

11.3.9 TimerA Hardware Trigger Event Selection Register (TMRA_TRGSELx, x = 0 ~ 3)

After selecting the internal hardware trigger in TimerA, use this register to select the trigger event source and write the trigger event number into the TRGSEL[8:0] bits of this register. When a peripheral event corresponding to the number occurs, TimerA will be triggered by the event to take actions (including start/stop/clear/input capture).

All TimerA units share these 4 registers.

Reset value: 0x0000 01FF

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
COMEN[1:0]		Res													
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res							TRGSEL[8:0]								

Bit/Bitfield	Marking	Bit Name		Function	Read and Write
b31	COMEN[1]	Common Enable	Trigger	0: Disable the common trigger event of AOS_COMTRGSEL2 from triggering this AOS target 1: Allow the common trigger event of AOS_COMTRGSEL2 to trigger this AOS target	RW
b30	COMEN[0]	Common Enable	Trigger	0: Disable the common trigger event of AOS_COMTRGSEL1 from triggering this AOS target 1: Allow the common trigger event of AOS_COMTRGSEL1 to trigger this AOS target	RW
b29~b9	Res	Reserved Bit		Read at “0”, write at “0”	RW
b8~b0	TRGSEL[8:0]	Trigger Selection	Source	Write the number of the peripheral circuit event to be selected	RW

11.3.10 ADC1 Conversion Start On-chip Trigger Source Selection Register (ADC1_TRGSELx, x = 0~1)

After selecting the on-chip trigger mode for ADC1, use this register to select the trigger event source and write the trigger event number into the TRGSEL[8:0] bits of this register. When the peripheral event corresponding to the number occurs, ADC1 will be triggered by the event to perform analog-to-digital conversion.

Sequence A and sequence B of ADC1 share these 2 registers.

Reset value: 0x0000 01FF

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
COMEN[1:0]		Res													
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res							TRGSEL[8:0]								

Bit/Bitfield	Marking	Bit Name		Function	Read and Write
b31	COMEN[1]	Common Enable	Trigger	0: Disable the common trigger event of AOS_COMTRGSEL2 from triggering this AOS target 1: Allow the common trigger event of AOS_COMTRGSEL2 to trigger this AOS target	RW
b30	COMEN[0]	Common Enable	Trigger	0: Disable the common trigger event of AOS_COMTRGSEL1 from triggering this AOS target 1: Allow the common trigger event of AOS_COMTRGSEL1 to trigger this AOS target	RW
b29~b9	Res	Reserved Bit		Read at “0”, write at “0”	RW
b8~b0	TRGSEL[8:0]	Trigger Selection	Source	Write the number of the peripheral circuit event to be selected	RW

11.3.11 ACD2 Conversion Start On-chip Trigger Source Selection Register (ADC2_TRGSELx, x = 0~1)

After selecting the on-chip trigger mode for ADC2, use this register to select the trigger event source and write the trigger event number into the TRGSEL[8:0] bits of this register. When the peripheral event corresponding to the number occurs, ADC2 will be triggered by the event to perform analog-to-digital conversion.

ADC2 sequence A and sequence B share these two registers.

Reset value: 0x0000 01FF

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
COMEN[1:0]		Res													
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res							TRGSEL[8:0]								

Bit/Bitfield	Marking	Bit Name		Function	Read and Write
b31	COMEN[1]	Common Enable	Trigger	0: Disable the common trigger event of AOS_COMTRGSEL2 from triggering this AOS target 1: Allow the common trigger event of AOS_COMTRGSEL2 to trigger this AOS target	RW
b30	COMEN[0]	Common Enable	Trigger	0: Disable the common trigger event of AOS_COMTRGSEL1 from triggering this AOS target 1: Allow the common trigger event of AOS_COMTRGSEL1 to trigger this AOS target	RW
b29~b9	Res	Reserved Bit		Read at “0”, write at “0”	RW
b8~b0	TRGSEL[8:0]	Trigger Selection	Source	Write the number of the peripheral circuit event to be selected	RW

11.3.12 ADC3 Conversion Start On-chip Trigger Source Selection Register (ADC3_TRGSELx, x=0, 1)

After selecting the on-chip trigger mode for ADC3, use this register to select the trigger event source and write the trigger event number into the TRGSEL[8:0] bits of this register. When the peripheral event corresponding to the number occurs, ADC3 will be triggered by the event to perform analog-to-digital conversion.

Sequence A and sequence B of ADC3 share these two registers.

Reset value: 0x0000 01FF

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
COMEN[1:0]		Res													
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res							TRGSEL[8:0]								

Bit/Bitfield	Marking	Bit Name		Function	Read and Write
b31	COMEN[1]	Common Trigger Enable		0: Disable the common trigger event of AOS_COMTRGSEL2 from triggering this AOS target 1: Allow the common trigger event of AOS_COMTRGSEL2 to trigger this AOS target	RW
b30	COMEN[0]	Common Trigger Enable		0: Disable the common trigger event of AOS_COMTRGSEL1 from triggering this AOS target 1: Allow the common trigger event of AOS_COMTRGSEL1 to trigger this AOS target	RW
b29~b9	Res	Reserved Bit		Read at “0”, write at “0”	RW
b8~b0	TRGSEL[8:0]	Trigger Selection	Source	Write the number of the peripheral circuit event to be selected	RW

11.3.13 Common Trigger Source Selection Register (AOS_COMTRGSELx, x = 1~2)

Write the trigger event number in AOS_COMTRGSEL1/2. When the peripheral circuit event corresponding to the number occurs, if the COMEN bit value of 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 one or more AOS targets.

Reset value: 0x0000 01FF

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res							COMTRG[8:0]								

Bit/Bitfield	Marking	Bit Name		Function	Read and Write
b31~b9	Res	Reserved Bit		Read at “0”, write at “0”	RW
b8~b0	COMTRG[8:0]	Trigger Selection	Source	Write the number of the peripheral circuit event to be selected	RW

12 Memory Protection Unit (MPU)

12.1 Overview

The MPU can provide protection to the memory, and can improve the security of the system by preventing unauthorized access.

This chip has built-in 1 MPU unit for CPU, 1 MPU unit for CPU main stack pointer, 1 MPU unit for CPU thread stack pointer, 1 MPU unit for DMA and 1 MPU unit for IP.

Modules	Content
ARM MPU	Memory Protection Unit of the CPU 8 areas, see ARM MPU description for details
MSPMPU	CPU Main Stack Pointer Storage Protection Unit
PSPMPU	CPU Thread Stack Pointer Storage Protection Unit
System DMA MPU: SMPU1	Memory protection unit for system DMA 16 areas, 8 areas dedicated to system DMA, 8 areas shared by all DMAs
IPMPU	Access protection unit for system IP and security-related IP

Among them, the ARM MPU provides the access control of the CPU to the entire 4G address space.

MSPMPU/PSPMPU respectively provide protection for the CPU's main stack pointer/thread stack pointer. When the pointer exceeds the set range, the MPU action can be set to non-maskable interrupt/reset.

SMPU1 provides system DMA with read and write access rights control for the entire 4G address space. When accessing the prohibited space, the MPU action can be set to ignore/bus error/non-maskable interrupt/reset.

The IPMPU provides access control to system IP and security-related IP in non-privileged mode.

12.2 Functional Description

12.2.1 Locale Setting

The MPU manages the authority of the storage space in units of regions. Each area can independently set the base address and area size, the range of which can be set is 32Bytes~4GBytes, the size must be 2^n Bytes (n=5~32), and the corresponding lower n bits of the base address are 0.

The address space not covered by any region is called the background region.

12.2.2 Permission Settings

Each area, including the background area, can be independently set for each DMA to allow reading/prohibit reading and allow writing/prohibit writing. If address overlap occurs between different areas, the set prohibition takes precedence.

12.2.3 MPU Action Selection

When a prohibited access occurs, the access is ignored (read access reads 0, write access is ignored), and the corresponding action can be set, which can be set as:

- Ignore
- Bus error
- Nomaskable interrupt
- Reset

12.2.4 Start MPU

MSPMPU/ PSPMPU/ SMPU1 can be enabled independently.

It is recommended to enable the MPU after setting the area range/permission setting/action selection.

12.3Application Examples

12.3.1 Only Allow Partial Space Access

Example: Set the permission of the background area to prohibit reading/writing, set area 0 to allow reading/allow writing, set area 1 to allow reading and prohibit writing, and the range of area 0 covers area 1.

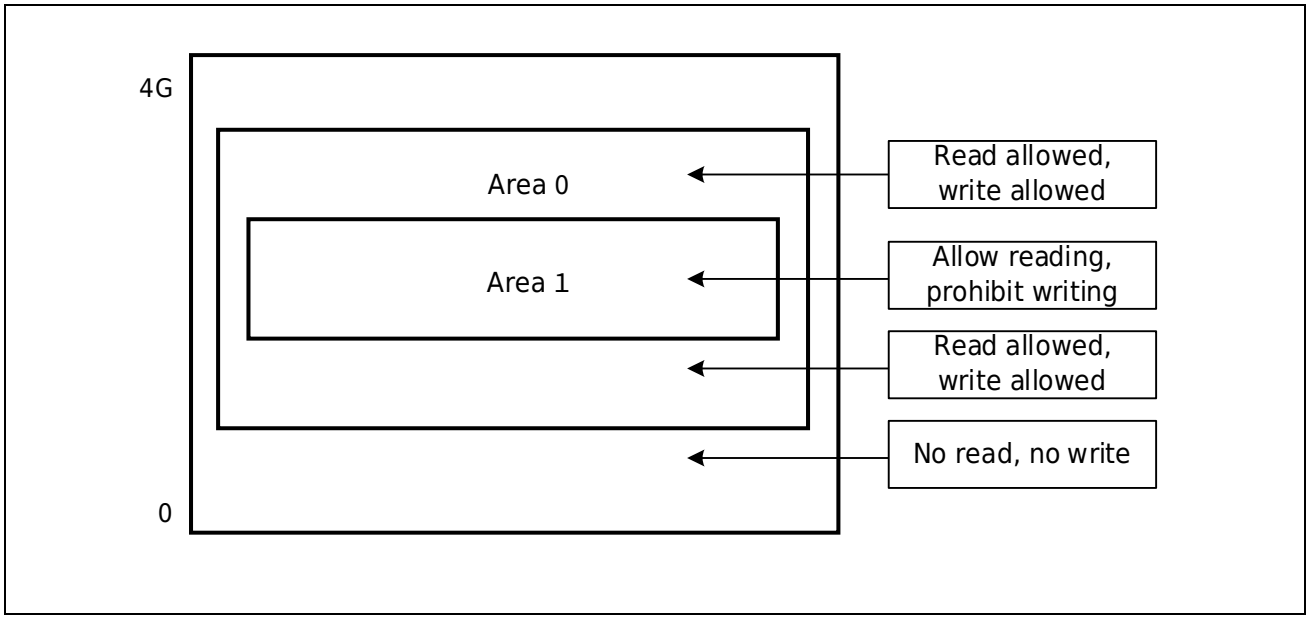


Figure 12-1 Only Allow Partial Space Access to Schematic Diagram

12.3.2 Only Some Space Access Prohibited

Example: Set the permission of the background area to allow reading/prohibiting writing, set area 0 to allow reading/allow writing, set area 1 to allow reading/prohibit writing, area 0 covers area 1, and set area 2 to prohibit reading/ prohibit writing.

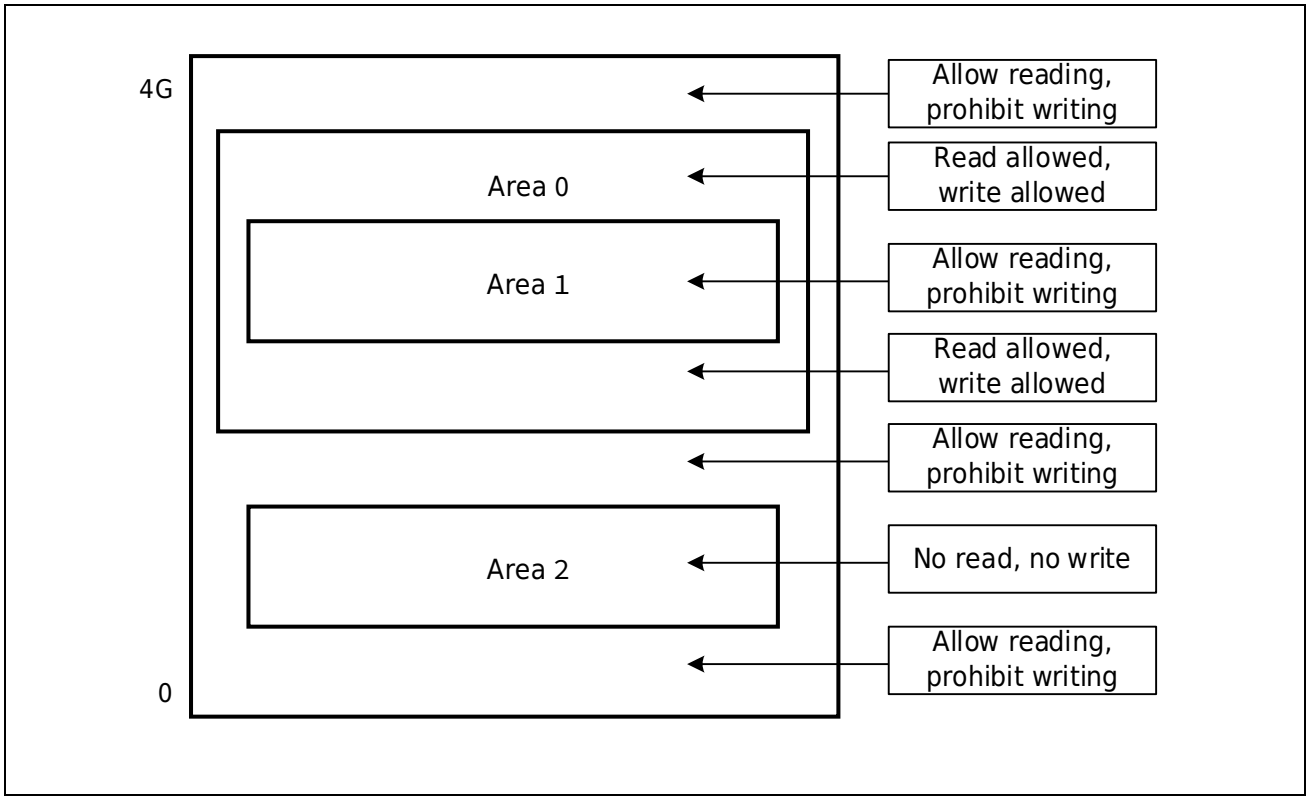


Figure 12-2 Schematic Diagram of Only Prohibiting Access to Part of Space

12.4Register Description

The registers of this module can only be set by the CPU.

Table 12-1 MPU Base Address

Name	Base Address	Description
MPU	0x4005 0000	MPU Base Address

Table 12-2 MPU Register List

Register	Description	Offset Address	Bit Width	Reset Value	Write protection
MPU_RGD0~15	Area 0~15 range description register	0x00~0x3C	32	0x0000 0000	MPUWE
MPU_SR	Status flag register	0x40	32	0x0000 0000	no
MPU_ECLR	Flag Clear Register	0x44	32	0x0000 0000	no
MPU_WP	Write protection register	0x48	32	0x0000 0000	WKEY
MPU_IPPR	IP access protection register	0x4C	32	0x0000 0000	MPUWE
MPU_MSPPBA	MSP Protection Start Address Register	0x50	32	0x0000 0000	MPUWE
MPU_MSPPCTL	MSP Protection Control Register	0x54	32	0x0000 0003	MPUWE
MPU_PSPPBA	PSP Protection Start Address Register	0x58	32	0x0000 0000	MPUWE
MPU_PSPPCTL	PSP Protection Control Register	0x5C	32	0x0000 0003	MPUWE
MPU_S1RGE	SMPU1 Region Enable Register	0x60	32	0x0000 0000	MPUWE
MPU_S1RGWP	SMPU1 Region Write Permission Register	0x64	32	0x0000 0000	MPUWE
MPU_S1RGRP	SMPU1 Region Read Permission Register	0x68	32	0x0000 0000	MPUWE
MPU_S1CR	SMPU1 Control Register	0x6C	32	0x0000 0000	MPUWE

12.4.1 Region Range Description Register (MPU_RGDn, n=0~15)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
MPURGnADDR [31:16]															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
MPURGnADDR[15:5]											MPURGnSIZE[4:0]				

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b5	MPURGnADDR[31:5]	Zone base address	Set the base address of area n, the effective number of digits is related to the area size, and the low (MPURGnSIZE+1) bit is fixed to 0	RW
b4~b0	MPURGnSIZE[4:0]	Area size	Set the size of the region n 00000~ 00011: Reserved, setting is prohibited 00100：32Bytes 00101：64Bytes … 11110：2GBytes 11111：4GBytes	RW

12.4.2 Status Flag Register (MPU_SR)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res													MSPEF	PSPEF	SMPU1EAF

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b3	Res	Reserved Bit	Reserved bit, read as 0, write 0 when writing	RW
b2	MSPEF	MSP error flag	0: MSP pointer is within the set range 1: The MSP pointer is out of the setting range	R
b1	PSPEF	PSP error flag	0: The PSP pointer does not exceed the setting range 1: The PSP pointer is out of the setting range	R
b0	SMPU1EAF	SMPU1 error flag	0: No system DMA access error occurs 1: System DMA has an access error.	R

Note:

The write operation to this register will be ignored, please use MPUECLR to clear the error flag.

12.4.3 Flag Clear Register (MPU_ECLR)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res													MSP ECLR	PSP ECLR	SMPU1E CLR

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b3	Res	Reserved Bit	Reserved bit, read as 0, write 0 when writing	RW
b2	MSPECLR	MSP error flag cleared	Writing 1 can clear MSPEF to 0	RW
b1	PSPECLR	PSP error flag cleared	Writing 1 can clear PSPEF to 0	RW
b0	SMPU1ECLR	SMPU1 error flag cleared	Writing 1 can clear SMPU1EAF to 0	RW

Note:

The read value of this register is fixed as 0x0000 0000.

12.4.4 Write Protection Register (MPU_WP)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
WKEY[15:1]															MPUWE

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b6	Res	Reserved Bit	Reserved bit, read as 0, write 0 when writing	RW
b15~b1	WKEY[15:1]	Write code	When writing to MPUWE, you must also write 0x96A5 to WKEY, which reads as 0	W
b0	MPUWE	MPU register write enable	0: MPU address registers/control registers are not allowed to be written 1: MPU address register/control register allows writing	RW

Note:

Writing 0x96A5 to this register can set MPUWE to 1, writing 0x96A4 can clear MPUWE to 0, and writing other values cannot change MPUWE.

12.4.5 IP Access Protection Register (MPU_IPPR)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
BUSERRE	Res	MSTPWRP	MSTPRDP	SYSCWRP	SYSCRDP	INTCWRP	INTCRDP	SRAMCWRP	SRAMCRDP	DMPUWRP	DMPURDP	RTCWRP	RTCRDP	BKSRAMWRP	BKSRAMRDP
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
SWDTWRP	SWDTRDP	WDTWRP	WDTRDP	Res		EFMWRP	EFMRDP	CRCWRP	CRCRDP	Res					

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31	BUSERRE	Bus Error Enable	0: Ignore access to protected IP 1: When an access to the protected IP occurs, a bus error is returned	RW
b30	Res	Reserved Bit	Reserved bit, read as 0, write 0 when writing	RW
b29	MSTPWRP	MSTP protection write	0: Allow write operations to PWC_FCG0/1/2/3 and PWC_FCG0PC 1: Disable write operations to PWC_FCG0/1/2/3 and PWC_FCG0PC	RW
b28	MSTPRDP	MSTP protection read	0: Allow read operations on PWC_FCG0/1/2/3 and PWC_FCG0PC 1: Disable the read operation of PWC_FCG0/1/2/3 and PWC_FCG0PC	RW
b27	SYSCWRP	SYSC protection write	0: Allow write operation to RMU/CMU/PWC 1: Disable writing to RMU/CMU/PWC **Note:** When BUSERRE is set to 1, write operations to the RMU/PWC will not cause bus errors	RW
b26	SYSCRDP	SYSC protection read	0: Allow read operation of RMU/CMU/PWC 1: Disable the read operation of RMU/CMU/PWC **Note:** When BUSERRE is set to 1, no bus error will occur when reading RMU/PWC	RW
b25	INTCWRP	INTC protection write	0: Allow write operation to INTC 1: Prohibit writing to INTC	RW
b24	INTCRDP	INTC protection read	0: Allow read operation on INTC 1: Disable the read operation of INTC	RW
b23	SRAMCWRP	SRAMC protection write	0: Allow writing to SRAMC 1: Forbid writing to SRAMC	RW
b22	SRAMCRDP	SRAMC protection read	0: Allow read operations on SRAMC 1: Disable read operation on SRAMC	RW
b21	DMPUWRP	DMPU protection write	0: Allow writing to DMPU 1: Forbid writing to DMPU	RW
b20	DMPURDP	DMPU protection read	0: Allow read operations on DMPU 1: Disable DMPU read operation	RW
b19	RTCWRP	RTC protection write	0: Allow writing to RTC 1: Disable writing to RTC	RW
b18	RTCRDP	RTC read protection	0: Allow read operations on RTC 1: Disable RTC read operation	RW
b17	BKSRAMWRP	BKSRAM protection write	0: Allow writing to Ret SRAM 1: Disable writing to Ret SRAM	RW
b16	BKSRAMRDP	BKSRAM protection read	0: Allow read operations on Ret SRAM 1: Disable read operation on Ret SRAM	RW
b15	SWDTWRP	SWDT protection write	0: Allow writing to SWDT 1: Forbid writing to SWDT	RW
b14	SWDTRDP	SWDT protection read	0: Allow read operations on SWDT 1: Disable read operation on SWDT	RW
b13	WDTWRP	WDT protection write	0: Allow writing to WDT 1: Forbid writing to WDT	RW
b12	WDTRDP	WDT protection read	0: Allow read operations on WDT 1: Disable WDT read operation	RW
b11-b10	Res	Reserved Bit	Reserved bit, read as 0, write 0 when writing	RW
b9	EFMWRP	EFM protection write	0: Allow writing to EFM 1: Forbid writing to EFM	RW
b8	EFMRDP	EFM read protection	0: Allow read operations on EFM 1: Disable EFM read operation	RW
b7	CRCWRP	CRC protection write	0: Allow writing to CRC 1: Forbid writing to CRC	RW
b6	CRCRDP	CRC read protection	0: Allow reading of CRC 1: Disable CRC read operation	RW
b5~b0	Res	Reserved Bit	Reserved bit, read as 0, write 0 when writing	RW

Note:

In the privileged mode, the object IP can be read and written without being affected by this register.

12.4.6 MSP Protection Start Address Register (MPU_MSPPBA)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
MSPPBA[31:16]															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
MSPPBA[15:2]														Res	

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b2	MSPPBA[31:2]	Allow range start address	Set the MSP pointer to allow the start address of the range	RW
b1~b0	Res	Reserved Bit	Reserved bit, read as 0, write 0 when writing	RW

12.4.7 MSP Protection Control Register (MPU_MSPPCTL)

Reset value: 0x0000 0003

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
MSPPE	MSPPACT	Res													
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
MSPPSIZE[15:2]														Res	

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31	MSPPE	MSP protection enable	0: MSP pointer protection invalid 1: MSP pointer protection valid	RW
b30	MSPPACT	MSP protection action selection	0: Nomaskable interrupt 1: Reset	
b29~b16	Res	Reserved Bit	Reserved bit, read as 0, write 0 when writing	RW
b15~b2	MSPPSIZE[15:2]	Allowable range size	Set the allowed range of the MSP pointer, the setting value is 4Bytes to 64KBytes When MSPPSIZE=0x0, the MSP pointer allows a range of 4 bytes. When MSPPSIZE=0x3FFF, the MSP pointer allows a range of 64 KBytes.	RW
b1~b0	Res	Reserved Bit	Reserved bit, read as 1, write 1 when writing	RW

12.4.8 PSP Protection Start Address Register (MPU_PSPPBA)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
PSPPBA[31:16]															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
PSPPBA[15:2]														Res	

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b2	PSPPBA[31:2]	Allow range start address	Set the PSP pointer to allow the start address of the range	RW
b1~b0	Res	Reserved Bit	Reserved bit, read as 0, write 0 when writing	RW

12.4.9 PSP Protection Control Register (MPU_PSPPCTL)

Reset value: 0x0000 0003

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
PSPPE	PSPPACT	Res													
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
PSPPSIZE[15:2]														Res	

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31	PSPPE	PSP protection enable	0: PSP pointer protection disable 1: PSP pointer protection valid	RW
b30	PSPPACT	PSP protection action selection	0: Nomaskable interrupt 1: Reset	-
b29~b16	Res	Reserved Bit	Reserved bit, read as 0, write 0 when writing	RW
b15~b2	PSPPSIZE[15:2]	Allowable range size	Set the PSP pointer allowed range, setting value 4Bytes～64KBytes When PSPPSIZE=0x0, the PSP pointer allows a range of 4 bytes. When PSPPSIZE=0x3FFF, the PSP pointer allows a range of 64 KBytes.	RW
b1~b0	Res	Reserved Bit	Reserved bit, read as 1, write 1 when writing	RW

12.4.10 SMPU1 Region Enable Register (MPU_S1RGE)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
S1RG15E	S1RG14E	S1RG13E	S1RG12E	S1RG11E	S1RG10E	S1RG9E	S1RG8E	S1RG7E	S1RG6E	S1RG5E	S1RG4E	S1RG3E	S1RG2E	S1RG1E	S1RG0E

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b16	Res	Reserved Bit	Reserved bit, read as 0, write 0 when writing	RW
b15	S1RG15E	SMPU1 region 15 enable	0: Area 15 of SMPU1 invalid 1: Area 15 of SMPU1 valid	RW
b14	S1RG14E	SMPU1 region 14 enable	0: Area 14 of SMPU1 invalid 1: Area 14 of SMPU1 valid	RW
b13	S1RG13E	SMPU1 region 13 enable	0: Area 13 of SMPU1 invalid 1: Area 13 of SMPU1 valid	RW
b12	S1RG12E	SMPU1 region 12 enable	0: Area 12 of SMPU1 invalid 1: Area 12 of SMPU1 valid	RW
b11	S1RG11E	SMPU1 region 11 enable	0: Area 11 of SMPU1 invalid 1: Area 11 of SMPU1 valid	RW
b10	S1RG10E	SMPU1 region 10 enable	0: Area 10 of SMPU1 invalid 1: Area 10 of SMPU1 valid	RW
b9	S1RG9E	SMPU1 region 9 enable	0: Area 9 of SMPU1 invalid 1: Area 9 of SMPU1 valid	RW
b8	S1RG8	SMPU1 region 8 enable	0: Area 8 of SMPU1 invalid 1: Area 8 of SMPU1 valid	RW
b7	S1RG7E	SMPU1 region 7 enable	0: Area 7 of SMPU1 invalid 1: Area 7 of SMPU1 valid	RW
b6	S1RG6E	SMPU1 region 6 enable	0: Area 6 of SMPU1 invalid 1: Area 6 of SMPU1 valid	RW
b5	S1RG5E	SMPU1 region 5 enable	0: Area 5 of SMPU1 invalid 1: Area 5 of SMPU1 valid	RW
b4	S1RG4E	SMPU1 region 4 enable	0: Area 4 of SMPU1 invalid 1: Area 4 of SMPU1 valid	RW
b3	S1RG3E	SMPU1 region 3 enable	0: Area 3 of SMPU1 invalid 1: Area 3 of SMPU1 valid	RW
b2	S1RG2E	SMPU1 region 2 enable	0: Area 2 of SMPU1 invalid 1: Area 2 of SMPU1 valid	RW
b1	S1RG1E	SMPU1 region 1 enable	0: Area 1 of SMPU1 invalid 1: Area 1 of SMPU1 valid	RW
b0	S1RG0E	SMPU1 region 0 enable	0: Area 0 of SMPU1 invalid 1: Area 0 of SMPU1 valid	RW

12.4.11 SMPU1 Region Write Permission Register (MPU_S1RGWP)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
S1RG15WP	S1RG14WP	S1RG13WP	S1RG12WP	S1RG11WP	S1RG10WP	S1RG9WP	S1RG8WP	S1RG7WP	S1RG6WP	S1RG5WP	S1RG4WP	S1RG3WP	S1RG2WP	S1RG1WP	S1RG0WP

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b16	Res	Reserved Bit	Reserved bit, read as 0, write 0 when writing	RW
b15	S1RG15WP	SMPU1 area 15 write permission	0: SMPU1 region 15 allows system DMA write 1: Disable system DMA write in SMPU1 region 15	RW
b14	S1RG14WP	SMPU1 area 14 write permission	0: SMPU1 region 14 allows system DMA write 1: Disable system DMA write in SMPU1 region 14	RW
b13	S1RG13WP	SMPU1 area 13 write permission	0: SMPU1 region 13 allows system DMA write 1: Disable system DMA write in SMPU1 region 13	RW
b12	S1RG12WP	SMPU1 area 12 write permission	0: SMPU1 region 12 allows system DMA write 1: Disable system DMA write in SMPU1 region 12	RW
b11	S1RG11WP	SMPU1 area 11 write permission	0: SMPU1 region 11 allows system DMA write 1: Disable system DMA write in SMPU1 region 11	RW
b10	S1RG10WP	SMPU1 area 10 write permission	0: SMPU1 region 10 allows system DMA write 1: Disable system DMA write in SMPU1 region 10	RW
b9	S1RG9WP	SMPU1 area 9 write permission	0: SMPU1 region 9 allows system DMA write 1: Disable system DMA write in SMPU1 region 9	RW
b8	S1RG8WP	SMPU1 area 8 write permission	0: SMPU1 region 8 allows system DMA write 1: Disable system DMA write in SMPU1 region 8	RW
b7	S1RG7WP	SMPU1 area 7 write permission	0: SMPU1 region 7 allows system DMA write 1: Disable system DMA write in SMPU1 region 7	RW
b6	S1RG6WP	SMPU1 area 6 write permission	0: SMPU1 region 6 allows system DMA write 1: Disable system DMA write in SMPU1 region 6	RW
b5	S1RG5WP	SMPU1 area 5 write permission	0: SMPU1 region 5 allows system DMA write 1: Disable system DMA write in SMPU1 region 5	RW
b4	S1RG4WP	SMPU1 area 4 write permission	0: SMPU1 region 4 allows system DMA write 1: Disable system DMA write in SMPU1 region 4	RW
b3	S1RG3WP	SMPU1 area 3 write permission	0: SMPU1 region 3 allows system DMA write 1: Disable system DMA write in SMPU1 region 3	RW
b2	S1RG2WP	SMPU1 area 2 write permission	0: SMPU1 region 2 allows system DMA write 1: Disable system DMA write in SMPU1 region 2	RW
b1	S1RG1WP	SMPU1 area 1 write permission	0: SMPU1 region 1 allows system DMA write 1: Disable system DMA write in SMPU1 region 1	RW
b0	S1RG0WP	SMPU1 area 0 write permission	0: SMPU1 region 0 allows system DMA write 1: Disable system DMA write in SMPU1 region 0	RW

12.4.12 SMPU1 Region Read Permission Register (MPU_S1RGRP)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
S1RG15RP	S1RG14RP	S1RG13RP	S1RG12RP	S1RG11RP	S1RG10RP	S1RG9RP	S1RG8RP	S1RG7RP	S1RG6RP	S1RG5RP	S1RG4RP	S1RG3RP	S1RG2RP	S1RG1RP	S1RG0RP

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b16	Res	Reserved Bit	Reserved bit, read as 0, write 0 when writing	RW
b15	S1RG15RP	SMPU1 area 15 read permission	0: SMPU1 region 15 allows system DMA to read 1: SMPU1 region 15 disables system DMA to read	RW
b14	S1RG14RP	SMPU1 area 14 read permission	0: SMPU1 region 14 allows system DMA read 1: SMPU1 region 14 prohibits system DMA read	RW
b13	S1RG13RP	SMPU1 area 13 read permission	0: SMPU1 region 13 allows system DMA read 1: SMPU1 region 13 prohibits system DMA read	RW
b12	S1RG12RP	SMPU1 area 12 read permission	0: SMPU1 region 12 allows system DMA read 1: SMPU1 region 12 prohibits system DMA read	RW
b11	S1RG11RP	SMPU1 area 11 read permission	0: SMPU1 region 11 allows system DMA read 1: SMPU1 region 11 disables system DMA to read	RW
b10	S1RG10RP	SMPU1 area 10 read permission	0: SMPU1 region 10 allows system DMA read 1: SMPU1 region 10 prohibits system DMA read	RW
b9	S1RG9RP	SMPU1 area 9 read permission	0: SMPU1 region 9 allows system DMA read 1: SMPU1 region 9 prohibits system DMA read	RW
b8	S1RG8RP	SMPU1 area 8 read permission	0: SMPU1 region 8 allows system DMA read 1: SMPU1 region 8 prohibits system DMA read	RW
b7	S1RG7RP	SMPU1 area 7 read permission	0: SMPU1 region 7 allows system DMA read 1: SMPU1 region 7 prohibits system DMA read	RW
b6	S1RG6RP	SMPU1 area 6 read permission	0: SMPU1 region 6 allows system DMA read 1: SMPU1 region 6 prohibits system DMA read	RW
b5	S1RG5RP	SMPU1 area 5 read permission	0: SMPU1 region 5 allows system DMA read 1: SMPU1 region 5 prohibits system DMA read	RW
b4	S1RG4RP	SMPU1 area 4 read permission	0: SMPU1 region 4 allows system DMA read 1: SMPU1 region 4 prohibits system DMA read	RW
b3	S1RG3RP	SMPU1 area 3 read permission	0: SMPU1 region 3 allows system DMA read 1: SMPU1 region 3 prohibits system DMA read	RW
b2	S1RG2RP	SMPU1 area 2 read permission	0: SMPU1 region 2 allows system DMA read 1: SMPU1 region 2 prohibits system DMA read	RW
b1	S1RG1RP	SMPU1 area 1 read permission	0: SMPU1 region 1 allows system DMA read 1: SMPU1 region 1 prohibits system DMA read	RW
b0	S1RG0RP	SMPU1 area 0 read permission	0: SMPU1 region 0 allows system DMA read 1: SMPU1 region 0 prohibits system DMA read	RW

12.4.13 SMPU1 Control Register (MPU_S1CR)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res								SMPU1E	Res			SMPU1ACT[1:0]		SMPU1BWP	SMPU1BRP

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b8	Res	Reserved Bit	Reserved bit, read as 0, write 0 when writing	RW
b7	SMPU1E	SMPU1 enable	0: SMPU1 invalid 1: SMPU1 effective	RW
b6~b4	Res	Reserved Bit	Reserved bit, read as 0, write 0 when writing	RW
b3~b2	SMPU1ACT[1:0]	SMPU1 action selection	Set the action when the system DMA access is prohibited 00: Read access reads 0, write access ignores 01: Read access reads 0, write access is ignored; bus error occurs 10: Read access reads 0, write access is ignored; non-maskable interrupt is generated 11: Reset	RW
b1	SMPU1BWP	SMPU1 background write permission setting	0: SMPU1 background space allows system DMA to write 1: Disable system DMA writing in SMPU1 background space	RW
b0	SMPU1BRP	SMPU1 background read permission setting	0: SMPU1 background space allows system DMA to read 1: SMPU1 background space prohibits system DMA read	RW

Note:

When multiple area settings overlap, the priority is: setting prohibited > setting allowed.

13 Internal Clock Calibrator (CTC)

13.1 Overview

The Clock Trimming Controller (CTC) automatically calibrates the internal high-speed oscillator (HRC). Because the influence of working environment on HRC frequency may cause deviation, CTC can automatically adjust HRC frequency by hardware based on external high precision reference clock to obtain an accurate HRC clock.

13.2 Main Features

- Hardware automatic calibration without software involvement
- Three external reference clock sources: XTAL, XTAL32, CTCREF
- Two calibration modes: continuous calibration and single calibration
- 16-bit calibration counter for frequency measurement with heavy load function
- 8-bit calibration target deviation value and 6-bit calibration value for frequency calibration
- Error interrupt to prompt calibration failure

13.3 Functional Description

13.3.1 CTC Block Diagram

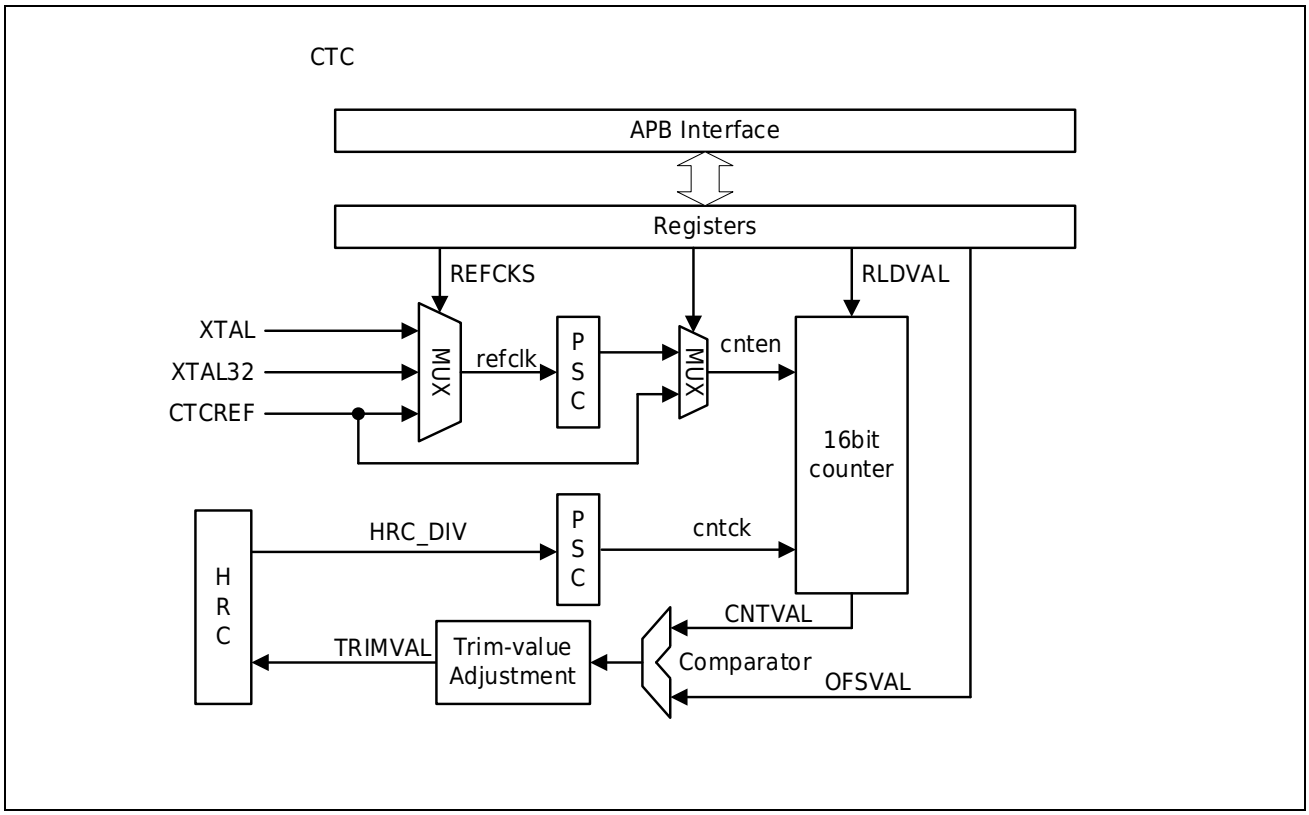


Figure 13-1 Basic Block Diagram of CTC

13.3.2 Reference Clock

There are three reference clocks for CTC continuous calibration, which are selected by the REFCKS[1:0] bits in the CTC_CR1 register.

- External high-speed clock (XTAL)
- External low-speed clock (XTAL32)
- External reference clock (CTCREF)

The jitter of HRC and the counting error during the calibration process will affect the calibration accuracy. The error can be reduced by setting the REFPSC[2:0] and HRCPSC[2:0] bits in the CTC_CR1 register to select appropriate frequency division for the reference clock and HRC respectively. The error calculation formula is as follows.

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

Note:

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 number of reference clock dividers;

Table 13-1 and Table 13-2 respectively list the error values when using reference clocks of different frequencies and calibrating under different frequency division conditions when the HRC target frequency is 16 MHz and 20 MHz, for quick reference.

Table 13-1 Measurement Error when HRC Target Frequency of 20MHz

F_{ref}		1KHz							
PSC_{ref}		2	4	8	16	64	256	1024	4096
PSC_{hrc}	1	0.167%	Irreducible	Irreducible	Irreducible	Irreducible	Irreducible	Irreducible	Irreducible
	2	0.177%	0.121%	Irreducible	Irreducible	Irreducible	Irreducible	Irreducible	Irreducible
	4	0.197%	0.131%	0.088%	Irreducible	Irreducible	Irreducible	Irreducible	Irreducible
	8	0.237%	0.151%	0.098%	0.065%	Irreducible	Irreducible	Irreducible	Irreducible
	32	0.477%	0.271%	0.158%	0.095%	0.038%	Irreducible	Irreducible	Irreducible
	128	1.437%	0.751%	0.398%	0.215%	0.068%	0.024%	Irreducible	Irreducible
	512	Not recommended	Not recommended	1.358%	0.695%	0.188%	0.054%	0.017%	Irreducible
	2048	Not recommended	Not recommended	Not recommended	Not recommended	0.668%	0.174%	0.047%	0.013%
F_{ref}		32.768KHz							

PSC_{ref}		2	4	8	16	64	256	1024	4096
PSC_{hrc}	1	1.224%	0.797%	0.530%	0.358%	0.169%	Irreducible	Irreducible	Irreducible
	2	1.551%	0.961%	0.612%	0.399%	0.179%	Irreducible	Irreducible	Irreducible
	4	Not recommended	1.289%	0.776%	0.481%	0.199%	0.089%	Irreducible	Irreducible
	8	Not recommended	1.944%	1.103%	0.644%	0.240%	0.100%	Irreducible	Irreducible
	32	Not recommended	Not recommended	Not recommended	1.628%	0.486%	0.161%	0.060%	Irreducible
	128	Not recommended	Not recommended	Not recommended	Not recommended	1.469%	0.407%	0.122%	0.040%
	512	Not recommended	Not recommended	Not recommended	Not recommended	Not recommended	1.390%	0.367%	0.102%
	2048	Not recommended	Not recommended	Not recommended	Not recommended	Not recommended	Not recommended	1.350%	0.347%
F_{ref}		100KHz							
PSC_{ref}		2	4	8	16	64	256	1024	4096
PSC_{hrc}	1	Not recommended	1.607%	1.033%	0.678%	0.308%	0.146%	不可设	不可设
	2	Not recommended	Not recommended	1.283%	0.803%	0.339%	0.154%	不可设	不可设
	4	Not recommended	Not recommended	1.783%	1.053%	0.402%	0.170%	0.077%	不可设
	8	Not recommended	Not recommended	Not recommended	1.553%	0.527%	0.201%	0.085%	不可设
	32	Not recommended	Not recommended	Not recommended	Not recommended	1.277%	0.388%	0.132%	0.050%
	128	Not recommended	Not recommended	Not recommended	Not recommended	Not recommended	1.138%	0.319%	0.097%
	512	Not recommended	Not recommended	Not recommended	Not recommended	Not recommended	Not recommended	1.069%	0.285%
	2048	Not recommended	Not recommended	Not recommended	Not recommended	Not recommended	Not recommended	Not recommended	1.035%
F_{ref}		1MHz							
PSC_{ref}		2	4	8	16	64	256	1024	4096
PSC_{hrc}	1	Not recommended	Not recommended	Not recommended	Not recommended	1.188%	0.516%	0.238%	Irreducible
	2	Not recommended	Not recommended	Not recommended	Not recommended	1.500%	0.594%	0.258%	0.119%
	4	Not recommended	Not recommended	Not recommended	Not recommended	Not recommended	0.750%	0.297%	0.129%

	8	Not recommended	Not recommended	Not recommended	Not recommended	Not recommended	1.063%	0.375%	0.148%
	32	Not recommended	Not recommended	Not recommended	Not recommended	Not recommended	Not recommended	0.844%	0.266%
	128	Not recommended	Not recommended	Not recommended	Not recommended	Not recommended	Not recommended	Not recommended	0.734%
	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	1.863%	0.775%	0.348%
	2	Not recommended	Not recommended	Not recommended	Not recommended	Not recommended	Not recommended	0.931%	0.388%
	4	Not recommended	Not recommended	Not recommended	Not recommended	Not recommended	Not recommended	1.244%	0.466%
	8	Not recommended	Not recommended	Not recommended	Not recommended	Not recommended	Not recommended	1.869%	0.622%
	32	Not recommended	Not recommended	Not recommended	Not recommended	Not recommended	Not recommended	Not recommended	1.559%
	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
F_{ref}		20MHz							
PSC_{ref}		2	4	8	16	64	256	1024	4096
PSC_{hrc}	1	Not recommended	Not recommended	Not recommended	Not recommended	Not recommended	Not recommended	1.369%	0.587%
	2	Not recommended	Not recommended	Not recommended	Not recommended	Not recommended	Not recommended	1.760%	0.684%
	4	Not recommended	Not recommended	Not recommended	Not recommended	Not recommended	Not recommended	Not recommended	0.880%
	8	Not recommended	Not recommended	Not recommended	Not recommended	Not recommended	Not recommended	Not recommended	1.270%
	32	Not recommended	Not recommended	Not recommended	Not recommended	Not recommended	Not recommended	Not recommended	Not recommended
	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

Note:

"Cannot be set" means that the setting is out of range for calibration counter to complete calibration.

"Not recommended" means that the setting error exceeds 2%, so it is not recommended.

Table 13-2 Measurement Error when HRC Target Frequency of 16MHz

F_{ref}		1KHz							
PSC_{ref}		2	4	8	16	64	256	1024	4096
PSC_{hrc}	1	0.188%	0.130%	Irreducible	Irreducible	Irreducible	Irreducible	Irreducible	Irreducible
	2	0.200%	0.136%	0.094%	Irreducible	Irreducible	Irreducible	Irreducible	Irreducible
	4	0.225%	0.149%	0.100%	0.068%	Irreducible	Irreducible	Irreducible	Irreducible
	8	0.275%	0.174%	0.113%	0.074%	Irreducible	Irreducible	Irreducible	Irreducible
	32	0.575%	0.324%	0.188%	0.112%	0.043%	Irreducible	Irreducible	Irreducible
PSC_{hrc}	128	1.775%	0.924%	0.488%	0.262%	0.081%	0.028%	Irreducible	Irreducible
	512	Not recommended	Not recommended	1.688%	0.862%	0.231%	0.065%	0.020%	Irreducible
	2048	Not recommended	Not recommended	Not recommended	Not recommended	0.831%	0.215%	0.058%	0.016%
F_{ref}		32.768KHz							
PSC_{ref}		2	4	8	16	64	256	1024	4096
PSC_{hrc}	1	1.411%	0.913%	0.603%	0.405%	0.190%	Irreducible	Irreducible	Irreducible
	2	1.821%	1.118%	0.706%	0.457%	0.203%	0.095%	Irreducible	Irreducible
	4	Not recommended	1.528%	0.911%	0.559%	0.228%	0.101%	Irreducible	Irreducible
	8	Not recommended	Not recommended	1.320%	0.764%	0.279%	0.114%	0.051%	Irreducible
	32	Not recommended	Not recommended	Not recommended	1.993%	0.587%	0.191%	0.070%	0.029%
	128	Not recommended	Not recommended	Not recommended	Not recommended	1.815%	0.498%	0.147%	0.048%
	512	Not recommended	Not recommended	Not recommended	Not recommended	Not recommended	1.727%	0.454%	0.125%
	2048	Not recommended	Not recommended	Not recommended	Not recommended	Not recommended	Not recommended	1.683%	0.432%
F_{ref}		100KHz							
PSC_{ref}		2	4	8	16	64	256	1024	4096

PSC_{hrc}	1	Not recommended	1.863%	1.188%	0.775%	0.348%	0.164%	Irreducible	Irreducible
	2	Not recommended	Not recommended	1.500%	0.931%	0.388%	0.174%	Irreducible	Irreducible
	4	Not recommended	Not recommended	Not recommended	1.244%	0.466%	0.194%	0.087%	Irreducible
	8	Not recommended	Not recommended	Not recommended	1.869%	0.622%	0.233%	0.097%	Irreducible
	32	Not recommended	Not recommended	Not recommended	Not recommended	1.559%	0.467%	0.155%	0.058%
	128	Not recommended	Not recommended	Not recommended	Not recommended	Not recommended	1.405%	0.390%	0.117%
	512	Not recommended	Not recommended	Not recommended	Not recommended	Not recommended	Not recommended	1.327%	0.351%
	2048	Not recommended	Not recommended	Not recommended	Not recommended	Not recommended	Not recommended	Not recommended	1.289%
F_{ref}		1MHz							
PSC_{ref}		2	4	8	16	64	256	1024	4096
PSC_{hrc}	1	Not recommended	Not recommended	Not recommended	Not recommended	1.369%	0.587%	0.269%	Irreducible
	2	Not recommended	Not recommended	Not recommended	Not recommended	1.760%	0.684%	0.293%	0.134%
	4	Not recommended	Not recommended	Not recommended	Not recommended	Not recommended	0.880%	0.342%	0.147%
	8	Not recommended	Not recommended	Not recommended	Not recommended	Not recommended	1.270%	0.440%	0.171%
	32	Not recommended	Not recommended	Not recommended	Not recommended	Not recommended	Not recommended	1.026%	0.318%
	128	Not recommended	Not recommended	Not recommended	Not recommended	Not recommended	Not recommended	Not recommended	0.904%
	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	Not recommended	0.887%	0.395%
	2	Not recommended	Not recommended	Not recommended	Not recommended	Not recommended	Not recommended	1.082%	0.444%
	4	Not recommended	Not recommended	Not recommended	Not recommended	Not recommended	Not recommended	1.473%	0.541%

	8	Not recommended	Not recommended	Not recommended	Not recommended	Not recommended	Not recommended	Not recommended	0.737%
PSC_{hrc}	32	Not recommended	Not recommended	Not recommended	Not recommended	Not recommended	Not recommended	Not recommended	1.908%
	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
F_{ref}		20MHz							
PSC_{ref}		2	4	8	16	64	256	1024	4096
PSC_{hrc}	1	Not recommended	Not recommended	Not recommended	Not recommended	Not recommended	Not recommended	1.582%	0.669%
	2	Not recommended	Not recommended	Not recommended	Not recommended	Not recommended	Not recommended	Not recommended	0.791%
	4	Not recommended	Not recommended	Not recommended	Not recommended	Not recommended	Not recommended	Not recommended	1.035%
	8	Not recommended	Not recommended	Not recommended	Not recommended	Not recommended	Not recommended	Not recommended	1.523%
	32	Not recommended	Not recommended	Not recommended	Not recommended	Not recommended	Not recommended	Not recommended	Not recommended
	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

Note:

"*Cannot be set*" *means that the setting is out of range for calibration counter to complete calibration.*
"Not recommended" means that the setting error exceeds 2%, so it is not recommended.

During a single CTC calibration, the reference signal can only be input from CTCREF, and the effective edge of the reference signal can be selected through the REFEDG[1:0] bits in the CTC_CR1 register. At this time, the setting of REFPSC[2:0] invalid, and each calibration is only performed between two valid edges of the reference signal input (such as rising edge to rising edge, which is called a calibration cycle). The error of a single calibration is calculated using the following formula.

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

Note:

F_{hrc} represents the target frequency of HRC, in Hz;

T_{trim} represents the calibration period, unit is S;

PSC_{hrc} represents the HRC frequency division number;

13.3.3 Frequency Calibration

After the CTCEN bit in the CTC_CR1 register is written to 1, the 16-bit calibration counter reloads RLDVAL from the CTC_CR2 register under the trigger of the reference clock and starts counting down. The counting clock is provided by the HRC. After counting starts, if the reference clock triggers counting 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.

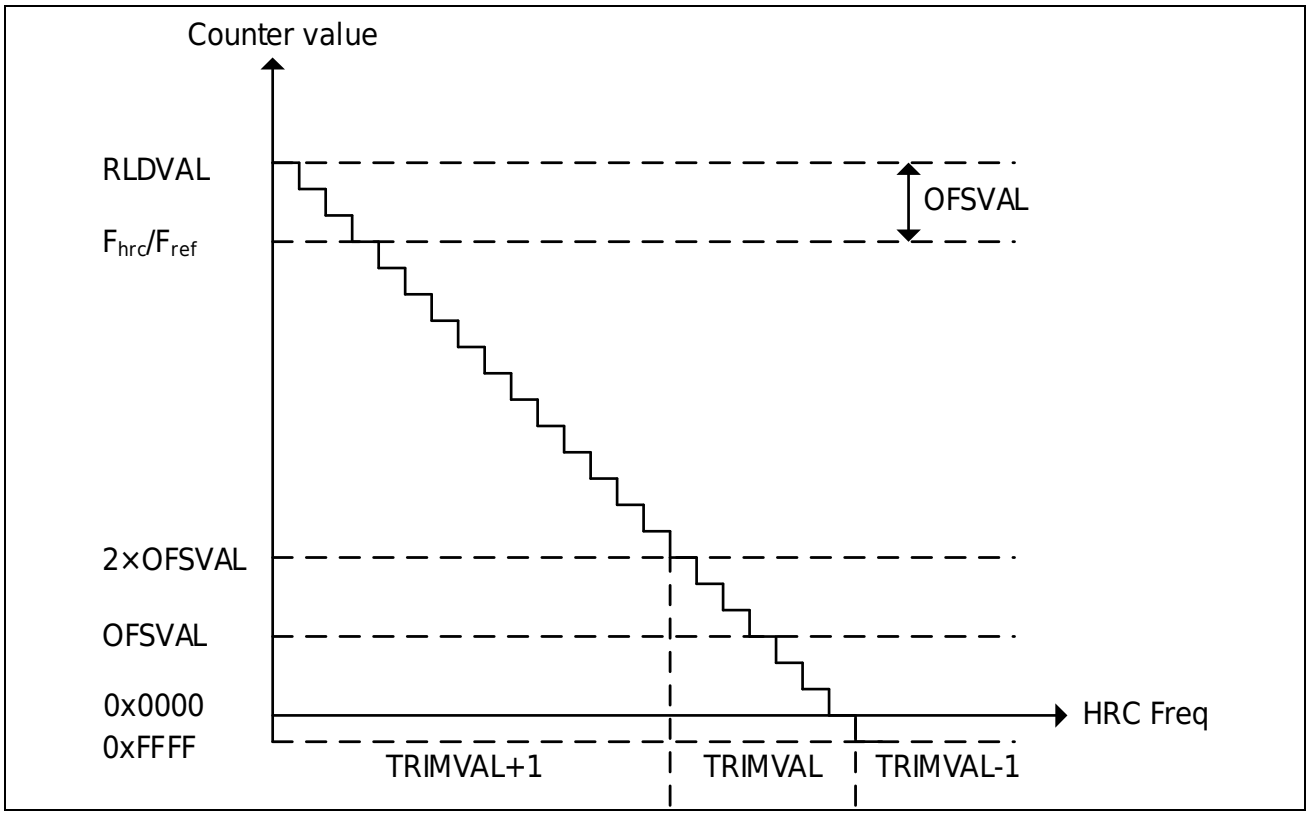


Figure 13-2 Schematic Diagram of CTC Calibration

After the 16-bit calibration counter stops counting, the HRC frequency calibration starts. Since the calibration target deviation value OFSVAL is added when setting the calibration count reload value RLDVAL, if the value of the 16-bit calibration counter is greater than 2×OFSVAL and no underflow occurs, it means that the current frequency is slower than the expected frequency, and the value of TRMVAL[5:0] in the CTC_CR1 register needs to be increased to increase the frequency. On the contrary, if the 16-bit calibration counter value underflows, the current frequency is faster than expected, and the TRMVAL value needs to be reduced to reduce the frequency. If the value of the 16-bit calibration counter is less than 2×OFSVAL, it means that the current frequency is within the expected frequency range, and there is no need to adjust the value of TRMVAL.

- When Counter > (OFSVAL×2), counting stops and the value of TRMVAL automatically increases by 1.

- Counting stops when 0 ≤ Counter ≤ (OFSVAL×2), the value of TRMVAL remains unchanged, and the TRIMOK bit of the CTC_STR register automatically changes to 1.
- Counter = 0xFFFF, the value of TRMVAL in the CTC_CR1 register is automatically reduced by 1.

Continuous calibration automatically starts the next calibration after each TRMVAL adjustment and stops when the software clears CTCEN or a calibration error occurs. Single calibration will automatically stop after each TRMVAL adjustment, and the software needs to write 1 to CTCEN to start the next calibration.

13.4 Interrupt

The median value of TRMVAL is 0x00. If the result of automatic addition is greater than 0x1F, the value of TRMVAL will remain 0x1F and a calibration overflow error will occur. At this time, the TRMOVF bit of the CTC_STR register automatically becomes 1. On the contrary, if the result of automatic subtraction is less than 0x20, the value of TRMVAL will remain 0x20 and a calibration underflow error will occur. At this time, the TRMUDF bit of the CTC_STR register automatically becomes 1. When a calibration overflow error or underflow error occurs, the calibration will automatically stop (CTCEN is automatically cleared). If the ERRIE bit in the CTC_CR1 register is 1, a calibration error interrupt request will occur. For specific setting methods, please refer to the [Interrupt controller (INTC)] section.

13.5 Application Examples

Take the example of using XTAL32 as the reference clock to calibrate the HRC to 16MHz with a calibration target deviation of less than 0.5%. Here, XTAL32 is set to 16 division and HRC is set to no division. By viewing Table 13-2, it can be seen that the calibration error is 0.405%, which is less than the calibration target deviation of 0.5%. The solution is feasible.

1) Set the register.

   CTC_CR2 = 0x1EAB 0027, where

   OFSVAL= (16÷0.032768)×16×0.5%≈39 (0x27);

   RLDVAL= (16÷0.032768)×16+39≈7851 (0x1EAB);

2) Set the register.

   CTC_CR1 = 0x0000 00E3, where
   TRMVAL = 0x00 (Calibrate from median)
   HRCPSC=0b000 (select HRC without frequency division)
   CTCEN = 1 (start CTC)
   ERRIE = 1 (Allow incorrect interrupt)
   REFCKS = 0b10 (select XTAL32 as reference clock for continuous calibration)
   REFPSC = 0b011 (Select 8 frequency division of reference clock)

3) After a while, confirm the status of the CTC_STR.TRIMOK bit.

   If TRIMOK=1, write 0 to CTCEN to stop CTC;

   If TRIMOK=0, return to (3);

4) Confirm that CTC_STR.CTCBSY=0, and proceed with the following operations.

If an error interrupt occurs during the journey, the CTCEN bit is cleared automatically and the calibration stops due to a calibration overflow or a calibration underflow outside the calibrable range. In this case, please correct OFSVAL, RLDVAL and reference clock settings and try calibration again.

Note: Frequency calibration is initiated or stopped by CTCEN bits in CTC _ CR1register. Therefore, do not change any other register settings after frequency calibration is started. To modify the settings, write the CTCEN bit to 0 and confirm that the CTCBSY bit in the CTC _ STRregister is 0.

13.6 Register Description

Table 13-3 CTC Base Address

Name	Base Address	Description
CTC	0x4004 9C00	CTC Base Address

Table 13-4 CTC Register List

Register	Description	Offset Address	Bit Width	Reset Value
CTC_CR1	Clock calibration control register1	0x00	32	0x0000 0000
CTC_CR2	Clock calibration control register2	0x04	32	0x0000 0000
CTC_STR	Clock Calibration Status Register	0x08	32	0x0000 0000
CTC_CNT	Clock Calibration Counter	0x0C	16	0x0000

13.6.1 Clock Calibration Control Register1 (CTC _ CR1)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res										TRMVAL[5:0]					
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res		REFEDG[1:0]		Res	HRCPSC[2:0]			CTC EN	ERR IE	REFCKS[1:0]		Res	REFPSC[2:0]		

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b22	Res	Reserved Bit	Read at “0”, write at “0”	R
b21~b16	TRMVAL[5:0]	HRC calibration value	TRMVAL is a 6-bit signed calibration value with a median value of 0. 0x20：-32 0x21：-31 …… 0x3F：-1 0x00: 0 (median) 0x01：+1 …… 0x1E：+30 0x1F：+31 When the CTCEN bit is written as 1, TRMVAL is read-only and is automatically modified by hardware. If modification is required, please write 0 to the CTCEN bit and confirm that the CTC_STR.CTCBSY bit is 0 before modification. In any case, the final frequency calibration value of HRC is the result of adding the factory calibration value to TRMVAL. Therefore, if the frequency under the factory calibration value meets the requirements and no calibration is required, please set TRMVAL to 0x00.	RW
b15~b14	Res	Reserved Bit	Read at “0”, write at “0”	R
b13~b12	REFEDG[1:0]	Reference signal edge selection	Used to select the valid edge of CTCREF when REFCKS = 0b01. 00: Select the rising edge to the rising edge of CTCREF for calibration 01: Select the rising edge to the falling edge of CTCREF for calibration 10: Select the falling edge to the rising edge of CTCREF for calibration 11: Select the falling edge to the falling edge of CTCREF for calibration When CTCEN is 1, it is prohibited to modify the value of REFEDG. If modification is required, please write CTCEN to 0 and confirm that CTC_STR.CTCBSY is 0 before modification.	RW
b11	Res	Reserved Bit	Read at “0”, write at “0”	R
b10~b8	HRCPSC[2:0]	HRC frequency division selection	000: No division 001: 2 frequency division 010: 4 frequency division 011: 8 frequency division 100: 32 frequency division 101: 128 frequency division 110:512 frequency division 111:2048 frequency division When CTCEN is 1, it is prohibited to modify the value of HRCPSC. If modification is required, please write CTCEN to 0 and confirm that CTC_STR.CTCBSY is 0 before modification.	RW
b7	CTCEN	Calibration enable	Used to start frequency calibration. 0: Stop frequency calibration 1: Start-up frequency calibration Setting conditions: Software Set; Clearing Conditions: (1) Software reset; (2) Automatically clear to zero when overflow or underflow occurs in calibration;	RW

			(3) Automatically reset to zero at the end of a single calibration;	
b6	ERRIE	Error interrupt permission	Used to select whether error interrupt occurs when a calibration overflow or underflow occurs. 0: Prevent error interrupt 1: Allow error interrupt	RW
b5~b4	REFCKS[1:0]	Reference Clock selection	Used to select a reference timer for frequency calibration. 00: Select CTCREF as reference clock (continuous calibration) 01: Select CTCREF as the reference signal (single calibration) 10: Select XTAL32 as reference clock (continuous calibration) 11: Select XTAL as reference clock (continuous calibration) When CTCEN is 1, it is prohibited to modify the value of REFCKS. If modification is required, please write CTCEN to 0 and confirm that CTC_STR.CTCBSY is 0 before modification.	RW
b3	Res	Reserved Bit	Read at “0”, write at “0”	R
b2~b0	REFPSC[2:0]	Reference Clock frequency division selection	When REFCKS=0b00/0b10/0b11, select the reference clock division number. 000: 2 frequency division 001: 4 frequency division 010: 8 frequency division 011:16 frequency division 100: 64 frequency division 101:256 frequency division 110: 1024 frequency division 111:4096 frequency division When CTCEN is 1, it is prohibited to modify the value of REFPSC. If modification is required, please write CTCEN to 0 and confirm that CTC_STR.CTCBSY is 0 before modification.	RW

13.6.2 Clock Calibration Control Register 2 (CTC _ CR2)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
RLDVAL[15:0]															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res								OFSVAL[7:0]							

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b16	RLDVAL[15:0]	Calibration count overload	RLDVAL sets the reload value of the 16-bit calibration counter. During continuous calibration, RLDVAL = (F_{hrc}÷PSC_{hrc})÷ (F_{ref}÷PSC_{ref})+OFSVAL For single calibration, RLDVAL = ((F_{hrc}÷PSC_{hrc})×T_{trim})+OFSVAL F_{hrc}: HRC target frequency; F_{ref}: reference clock frequency; T_{trim}: single calibration cycle; OFSVAL: Calibration count offset; PSC_{hrc}: HRC frequency division number; PSC_{ref}: reference clock frequency division number; When CTCEN is 1, it is prohibited to modify the value of RLDVAL. If modification is required, please write CTCEN to 0 and confirm that CTC_STR.CTCBSY is 0 before modification.	RW
b15~b8	Res	Reserved Bit	Read at “0”, write at “0”	R
b7~b0	OFSVAL[7:0]	Calibration target deviation value	OFSVAL sets the permissible deviation value for HRC calibration. During continuous calibration, OFSVAL = (F_{hrc}÷PSC_{hrc})÷ (F_{ref}÷PSC_{ref})× A For single calibration, OFSVAL = ((F_{hrc}÷PSC_{hrc})×T_{trim})×T_A F_{hrc}: HRC target frequency; F_{ref}: reference clock frequency; T_{trim}: single calibration cycle; TA: Calibration accuracy, e.g. 0.5%; PSC_{hrc}: HRC frequency division number; PSC_{ref}: reference clock frequency division number; When CTCEN is 1, it is prohibited to modify the value of OFSVAL. If modification is required, please write CTCEN to 0 and confirm that CTC_STR.CTCBSY is 0 before modification.	RW

13.6.3 Clock Calibration Status Register (CTC _ STR)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res												CTCBSY	TRMUDF	TRMOVF	TRIMOK

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b4	Res	Reserved Bit	Read at “0”, write at “0”	R
b3	CTCBSY	CTC status flag	0: CTC is stopped 1: CTC is working	R
b2	TRMUDF	Calibration underflow mark	Calibration underflow indicates that TRMVAL has been reduced to 0x20 but has not yet been calibrated to the target range. Auto stop calibration when calibration underflow, error interrupt when ERRIE is 1. 0: No calibration underflow 1: Calibration underflow occurred TRMUDF is cleared when CTCEN is written with 1.	R
b1	TRMOVF	Calibration overflow mark	Calibration overflow indicates that TRMVAL has increased to 0x1F but has not yet been calibrated to the target range. Auto stop calibration when calibration overflow, error interrupt when ERRIE is 1. 0: No calibration overflow 1: Calibration overflow occurred TRMOVF is cleared when CTCEN is written with 1.	R
b0	TRIMOK	Calibration success mark	0: HRC frequency not calibrated to target range 1: HRC frequency calibrated to target range TRIMOK is cleared when CTCEN is written with 1.	R

13.6.4 Clock Calibration Counter (CTC_CNT)

Reset value: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
CTCCNT[15:0]															

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b15~b0	CTCCNT[15:0]	Calibration counter	The calibration counter reloads the value of RLDVAL at the beginning of each calibration and counts down from this value. After calibration, the counter value will be retained until the next RLDVAL reload. Please write 0 to CTCEN and confirm that CTC_STR.CTCBSY is 0 before reading.	R

14 Direct Memory Access Controller (DMA)

14.1 Overview

DMA is used to transfer data between memory and peripheral functional modules. It , can exchange data between memory, between memory and peripheral functional modules and between peripheral functional modules without CPU involvement.

14.2 Main Features

- DMA bus is independent of CPU bus and transmitted according to AMBA AHB-Lite bus protocol.
- With 1 DMA control units, a total of 8 independent channels, which can independently operate different DMA transfer functions
- For each channel, the source of the boot request is configured via a separate trigger source
- One block per request
- The data block is a minimum of one data and can be up to 1024 data
- The width of each data can be configured as 8bit, 16bit or 32bit
- You can configure 1~65535 transmissions or unlimited transmissions
- The source address and target address can be independently configured as fixed, auto-increment, auto-decrement, loop or jump with a specified offset
- Three types of interrupts can be generated: block transport complete interrupt, transport complete interrupt and transport error interrupt. Each interrupt can be configured with a mask or not. Among them, the block transmission is completed, and the transmission completion can be used as an event output, which can be used as a trigger source for other peripheral modules
- Support for linked transmission to enable multiple blocks of data to be transmitted at a time
- Support channel reset triggered by external events
- You can set the module stop state to reduce Power Consumption when not in use
- Support software start and software triggered channel reset function

14.3Functional Description

14.3.1 DMA Block Diagram

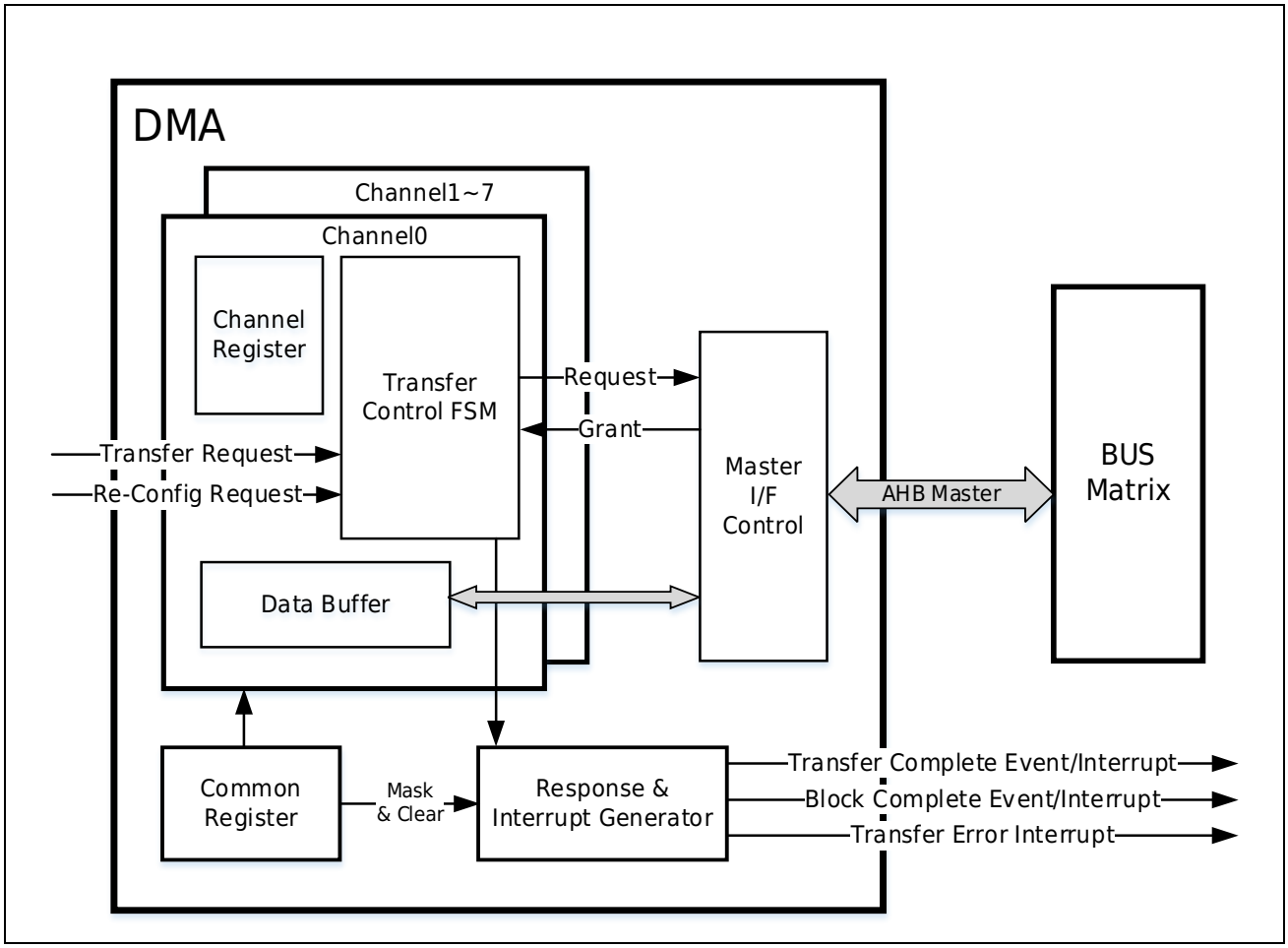


Figure 14-1 DMA Structure Diagram

14.3.2 Enable DMA Controller

When using DMA, you need to first write 0 to the function clock control 0 register FCG0.DMAx bit to allow the DMA clock supply, and then write 1 to the DMA enable register DMA_EN.EN bit. At the same time, DMA requires peripheral circuits to start transmission, so the FCG0.AOS peripheral circuit trigger function bit also needs to be written to 0.

When DMA is not used, or the chip needs to enter Stop Mode, please set DMA_EN.EN to 0 and write FCG0 to turn off the DMA clock. En Please confirm that registerDMA _ CHSTAT.DMAACT is 0 before writing 0 to ensure that DMA has completed all transfers.

14.3.3 Channel Selection and Channel Priority

Each DMA control unit contains 8 channels, and each channel can configure the transfer function independently.

The priority order of the 8 channels is: Channel 0>channel 1>channel 2>channel 7.

When a DMA unit has multiple channel requests to transfer, it 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. Theserequests are configured through the trigger source selection register DMA_TRGSELx (x=0~7), and each channel independently configures the start request source. When the peripheral circuit generates a start request or the software writes the register to generate a start request, and the DMA transmission is enabled DMA_EN.EN=1, and the transmission channel is in the permitted state DMA_CHEN.CHEN[x]=1, Then the channel x transmission is started.

Before use, it is necessary to enable the peripheral circuit trigger function and DMA function enable position of the function clock control 0 register (FCG0).

14.3.5 Data Block

The amount of data transmitted by the DMA each time is represented by a block. The size of the block is set by the data control register DMA_DTCTLx.BLKSIZE, and a maximum of 1024 data can be set. The data width of each is determined by DMA_CHCTLx.HSIZE and can be 8 bits, 16 bits or 32 bits.

14.3.6 Transmission Address Control

The source and destination addresses can be fixed, incremented, decremented, overloaded or discontinuous.

Fixed: The source address and destination address will be fixed during the transmission process.

Increment and decrement: The source address and destination address will increase or decrease according to the value of HSIZE after each data transmission is completed. For example, when HSIZE is 8bit, the address will increase/decrease by 1 each time, when it is 16bit, it will increase/decrease by 2 each time, and when it is 32bit, it will increase/decrease by 4 each time.

Overload: After transferring the specified amount of data, the source and destination addresses will return to the original address setting value. The amount of data that needs to be transferred before the address is reloaded, that is, the size of the repeated area is set by the register DMA_RPT.

Discontinuous address transfer: After transferring the specified amount of data, the source and destination addresses will skip the specified offset. The offset of the address jump and the amount of data to be transferred before the jump, that is, the size of the discontinuous area, are set by the register DMA_SNSEQCTL/DMA_DNSEQCTL. When the conditions of address overloading and discontinuous jump are met at the same time, address overloading is performed.

14.3.7 Number of Transmissions

The number of total data blocks transferred by DMA is set by the CNT bit of the data control register DMA_DTCTLx. The maximum number of transmissions can be set to 65535. Every time a data block is transmitted, the register value is reduced by 1. When the register value is reduced to 0, it means that all data transmission of this channel is completed, and the channel transmission permission bit DMA_CHEN.CHEN[x] is automatically cleared, and a transmission completion interrupt is generated. If DMA_DTCTLx.CNT is set to 0 when the transfer starts, it means unlimited transfers. Each time a request is initiated to transfer a block of data, the channel transfer permission bit is not cleared, and no transfer completion interrupt is generated.

Note:

– *After CHEN[x] is automatically cleared to 0, if you need to start this channel again, you need to reset the channel configuration register and then write CHEN[x] to 1 to enable this channel. Otherwise, the subsequent transmission will start according to the state at the end of the last transmission, that is, the source/destination address is the next address corresponding to the last transmission, and CNT is 0 for unlimited transmission.*

14.3.8 Chain Transmission

DMAcontroller has the function of linkage transmission. The chain transmission needs to configure the following eight registers, a total of eight words, called a descriptor (descriptor), which contains the source address, destination address, data control information, address control information, chain pointer and transmission control information of the chain transmission.

DMA_SARx
DMA_DARx
DMA_DTCTLx
DMA_RPTx
DMA_SNSEQCTLx
DMA_DNSEQCTLx
DMA_LLPx
DMA_CHCTLx

Where LLP is called a link-list Pointer, where the value represents the next descriptor's first address in the memory. When using chain transfer, first write LLPEN of the channel control register DMA_CHxCTLx to enable chain transfer, and write the descriptor information of the first transfer into the corresponding register. Descriptors for subsequent transfers are then initialized in memory in sequence. When it is necessary to end the chain transfer, set the LLPEN of DMA_CHxCTLx in the last descriptor of the channel to invalid, and the DMA controller will end the chain transfer after the transfer is completed.

When the last transfer of a descriptor ends, the next descriptor specified by the LLP is loaded from the memory into the channel configuration register. Wait for the next transfer request to be entered to start the first transfer of the new descriptor. Or, depending on the setting of register DMA_CHCTLx.LLPRUN, the first transfer can be started 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 from memory into the channel registers, BTC and TC interrupts are generated as usual, and then it 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 for the last block transfer of the old descriptor, neither BTC nor TC interrupts are generated.

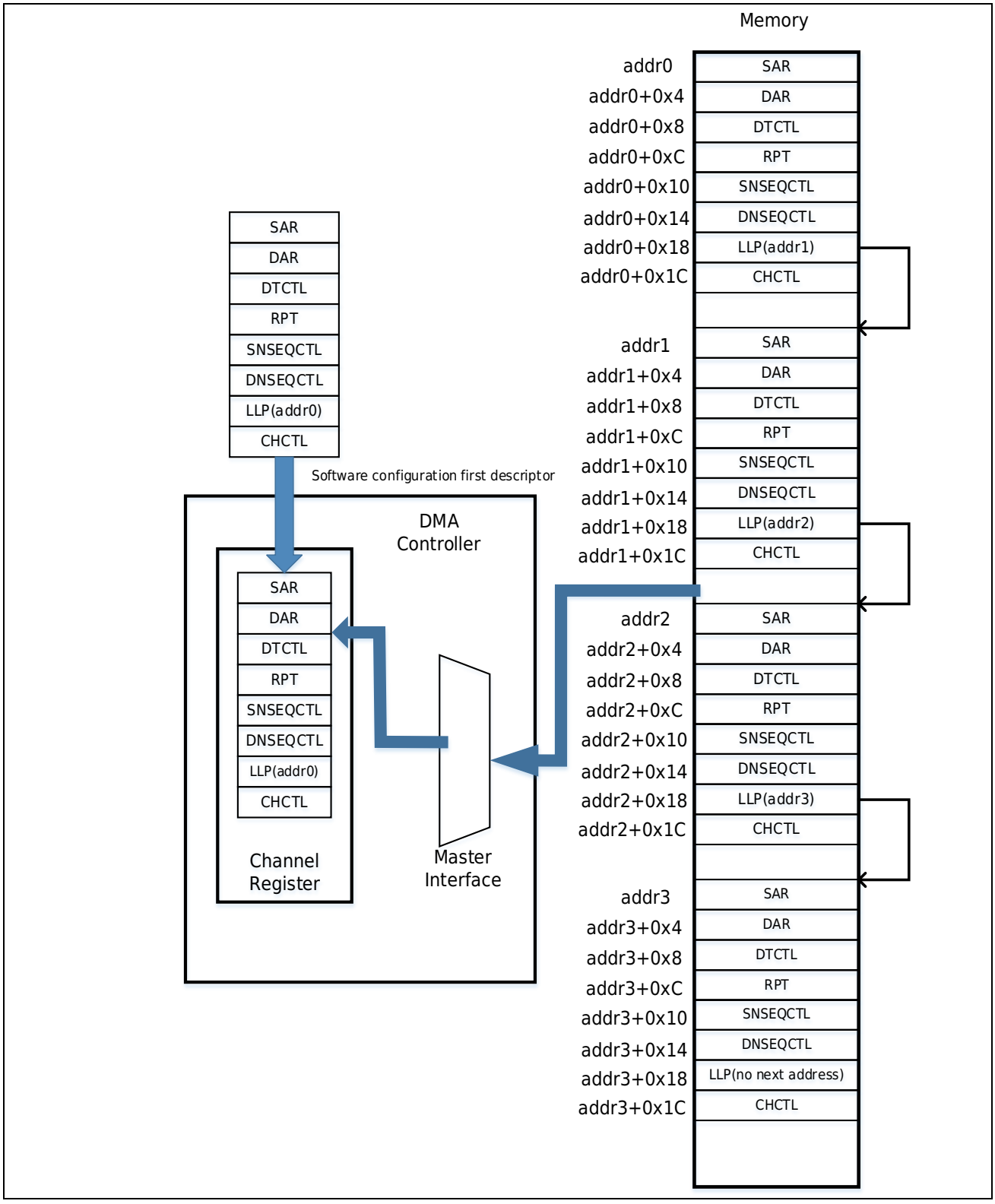


Figure 14-2 Chain Transmission diagram

14.3.9 Discontinuous Address Transmission

Using discontinuous address transmission can realize source address and destination address to jump according to a certain offset after a certain amount of data is transmitted. Take the discontinuous source address transmission as an example: first, set the channel control register

DMA_CHCTLx.SNSEQEN to 1 as needed to make the discontinuous address transmission effective. The jump direction is forward or backward according to the setting of DMA_CHCTLx.SINC, and then configure the register DMA_SNSEQCTLx to set the specific jump method. The control of the target address is similar to that of the source address, and the corresponding control registers are DMA_CHCTLx.DNSEQEN, DMA_CHCTLx.DINC, and DMA_DNSEQCTLx. The transmission process is as shown in the following figure.

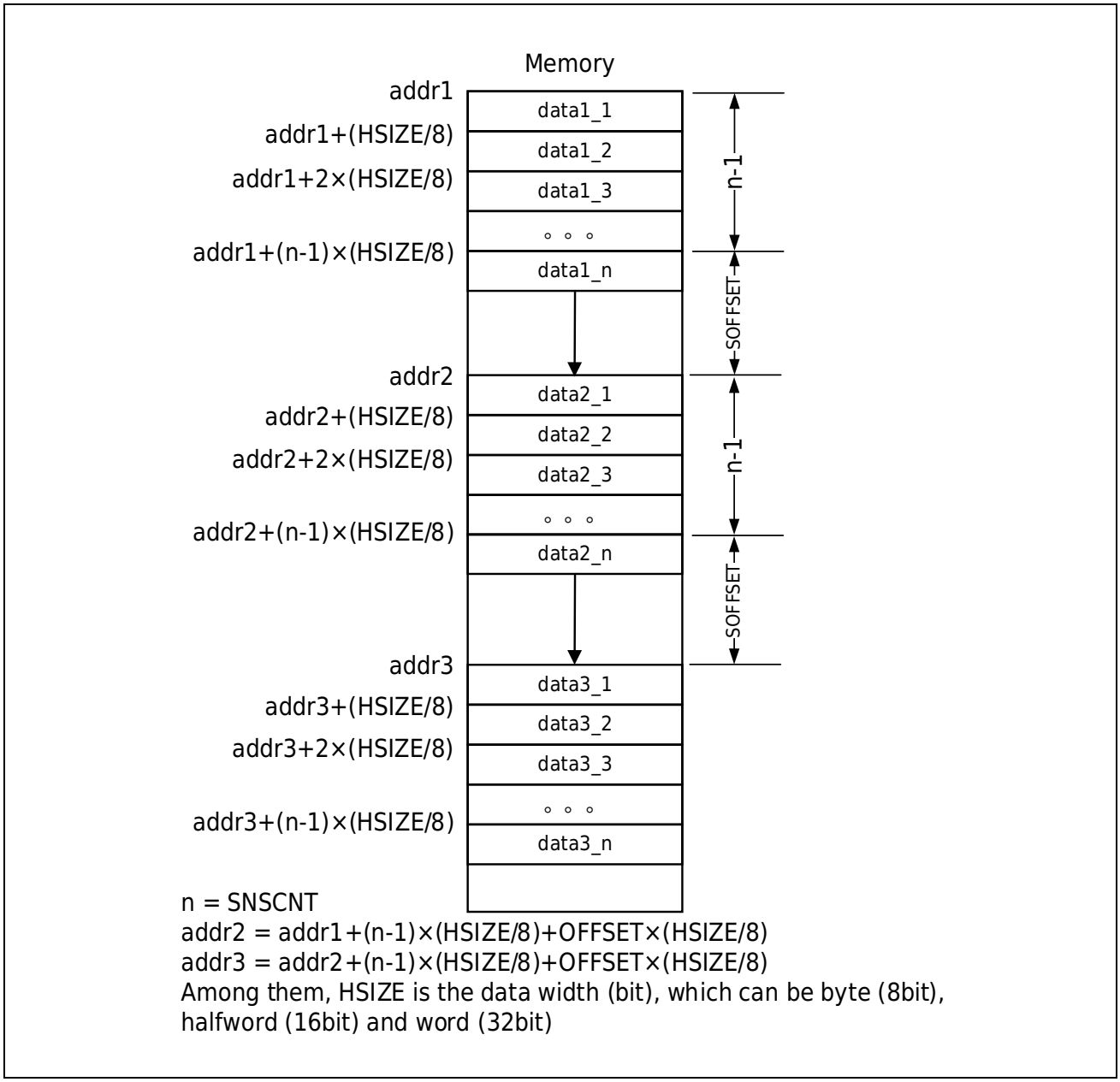


Figure 14-3 Schematic Diagram of Discontinuous Address Transmission (Taking Source Address as an Example)

14.3.10 Channel Reset

The channel reset function refers to modifying the internal status register of the channel through the event request of the peripheral circuit, and reconfiguring the next data transmission mode. Set register DMA_RCFGCTL.RCFGEN to 1 to allow channel reset. Select the reset request source through the trigger source selection register DMA_RC_TRGSEL. When the selected reset request source is input, the channel selected by the register DMA_RCFGCTL.RCFGCHS will be updated according to the specified method. A reset request only updates internal state and does not initiate the actual data transfer.

There are three ways to reset channels: chain pointer, discontinuous, and repeat.

When the chain pointer reset is selected, the descriptor and internal state of the channel are all updated to the new descriptor pointed to by the chain pointer LLP. Subsequent transfer requests are transferred according to the new descriptor.

When discontinuous or repeated reset is selected, the channel internal status is updated as described in the following table.

Table 14-1 Channel Reset Description

<table>
<tr><th colspan="3">Reset method (Register DMA_RCFGCTL)</th><th colspan="3">Channel internal state</th></tr>
<tr><th>CNTMD</th><th>SARMD</th><th>DARMD</th><th>Remaining Transfers Counter</th><th>The source address of the next transmission</th><th>The destination address of the next transmission</th></tr>
<tr><td>0x0</td><td>Arbitrarily</td><td>Arbitrarily</td><td>Keep it the same</td><td>Updated according to SARMD</td><td>Updated by DARMD</td></tr>
<tr><td rowspan="3">0x1</td><td>0x0</td><td rowspan="3">Arbitrarily</td><td>Keep it the same</td><td>Keep it the same</td><td rowspan="3">Updated by DARMD</td></tr>
<tr><td>0x1</td><td>Updated to the value after the next discontinuous jump of the source address in normal state</td><td>Update to the first address of the next source address discontinuous transmission area</td></tr>
<tr><td>0x2, 0x3</td><td>Updated to the value after the next source address reload occurs in normal state</td><td>Update to the initial setting value of register DMA_SARx</td></tr>
<tr><td rowspan="3">0x2, 0x3</td><td rowspan="3">Arbitrarily</td><td>0x0</td><td>Keep it the same</td><td rowspan="3">Updated according to SARMD</td><td>Keep it the same</td></tr>
<tr><td>0x1</td><td>Updated to the value after the next discontinuous jump of the target address in normal state</td><td>Update to the first address of the next target address discontinuous transfer area</td></tr>
<tr><td>0x2, 0x3</td><td>Updated to the value after the next target address reload occurs in normal state</td><td>Update to the initial setting value of register DMA_DARx</td></tr>
</table>

Note:

– *When the reset function is valid, the channel uses the registers DMA_RPTBx and DMA_SNSEQCTLBx, and DMA_DNSEQCTLBx controls the reload and discontinuous jump of the transfer address. Registers DMA_RPTx, DMA_SNSEQCTLx, DMA_DNSEQCTLx are invalid.*

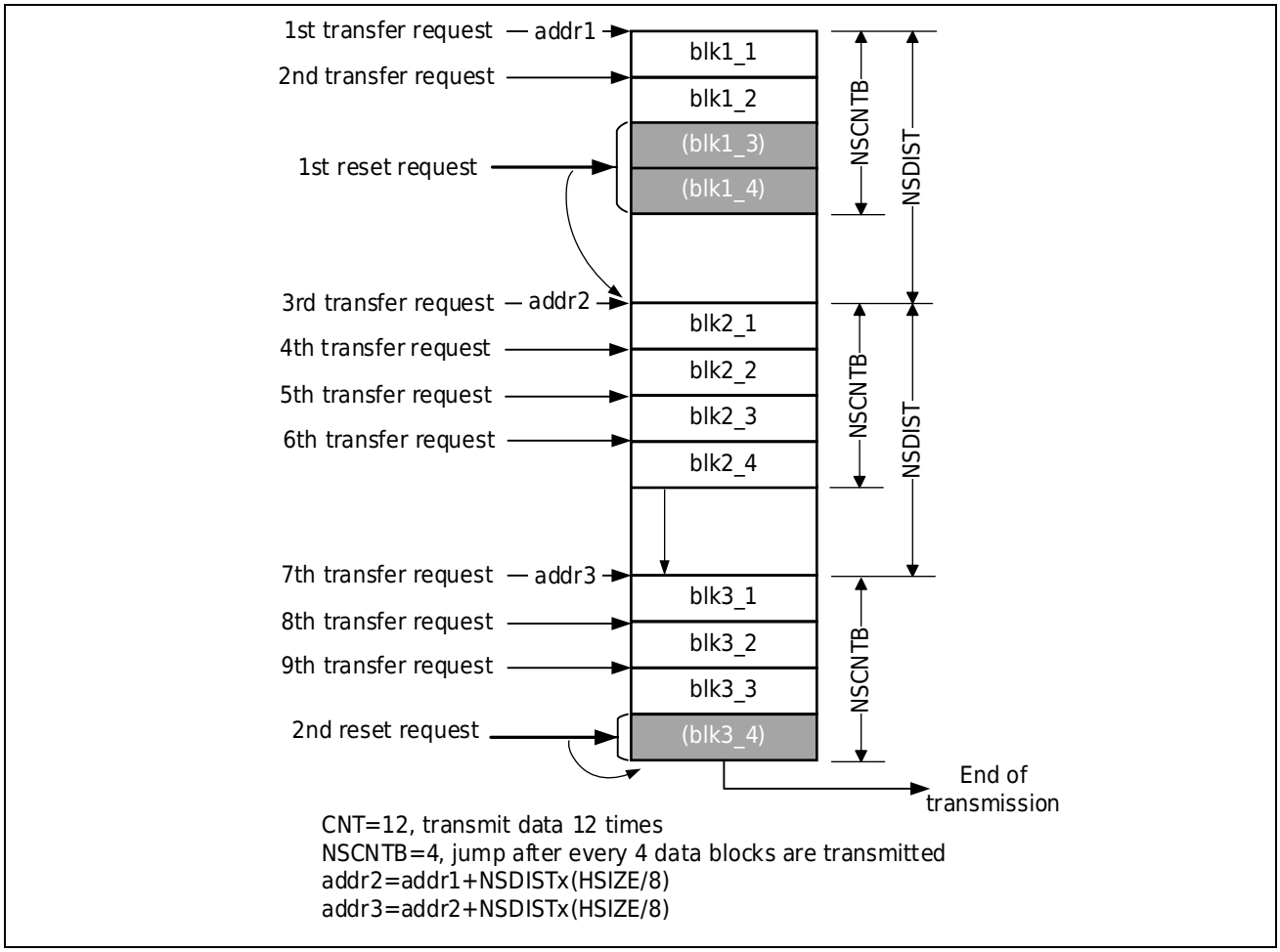


Figure 14-4 Schematic Diagram of Discontinuous Reset

Figure 14-4 in the DMA operation shown, each transfer request initiates the transfer of one block of data. After the first reset request occurs, the controller skips the data blocks blk1_3 and blk1_4, and the transmission address is updated to the first address of the next discontinuous area, which is addr2. After the second reset request occurs, the remaining number of transfers is updated to 0, that is, all data transfers are completed, the channel permission bit is automatically cleared to 0, and a transfer completion interrupt and event are generated.

14.3.11 Premature Interrupt of Transmission

During the transmission process, the channel enable register DMA_CHEN.CHENx remains valid. During non-chain transmission, the data control register DMA_DTCTLx is automatically set to invalid after the number of transmissions set by the data control register DMA_DTCTLx is complete ed. invalid. If the software writes 1 to DMA_CHENCLR.CHENCLRx during the transmission process, DMA will terminate the transmission after completing the current data reading and writing, and the DMA_CHEN.CHENx bit will be cleared to 0.

Note:

– *When the software writes 1 to the CHENCLRx bit to terminate the transfer early, the DMA will not save the transfer status when it is terminated. Without resetting the state of the channel*

configuration register (descriptor), write CHENx 1 to enable the channel again. After the transmission request is input, the DMA will retransmit the terminated data block instead of resuming the breakpoint.

14.4Interrupt

DMAcontroller can produce the following three interrupts:

Data Block Complete interrupt DMA _ BTCx: Generated after completing a data block transfer.

Transmission complete interrupt DMA_TCx: Generated after completing the number of transmissions set by the register DMA_DTCTLx.CNT.

Transmission error interrupt DMA_ERR: When the start request overflows (that is, the channel triggers the start request again when the last request of the channel has not been responded), or when a bus error occurs during the transmission (such as accessing an illegal address or a protected address) An interrupt is generated, where in a bus error terminates the transfer immediately.

Except for the start request overflow error, the above interrupts can be enabled or disabled by setting the register DMA_CHCTLx.IE. In addition, all interrupts are equipped with independent MASK registers to mask the interrupts.

The above-mentioned DMA_BTCx，DMA_TCx interrupt can also be used as an event signal output,

which can be used as a trigger source for other peripheral circuits. The event output is controlled by the MASK register, but not controlled by the interrupt per mission bit DMA_CHxCTL .IE The occurrence of DMA_BTCx, DMA_TCx, and DMA_ERR events will set the corresponding status bit registers and will not be affected by the DMA_CHCTLx.IE or MASK registers.

14.5Application Examples

14.5.1 Memory-to-memory transmission

Goal: Transfer 12 data from RAM address 0x2000 0000 to 0x2000 1000, with a data width of 32 bits.

1. Register Setting

1) Dma _ EN.EN Write 1 Enables DMAcontroller
2) Select a channel, such as channel 0, and configure the channel register to:
   a. Write DMA_SAR0 to configure the source address to SRAM area 0x2000 0000
   b. Write DMA_DAR0 to configure the target address to SRAM area 0x2000 1000
   c. Write the DMA_DTCTL0 configuration data block size is 4, the number of transfers is 3, after each transfer one data block to produce block transfer complete interrupt, after 3 transfers complete one transfer complete interrupt
   d. Write the DMA_RPT register to configure the size of the source address repetition area to

be 6, that is, reload the initial source address after the transfer of 6 addresses is completed

e. Configure channel control register DMA_CHCTL0 to achieve:
   a) Invalid link between source and destination
   b) Source address overload is valid, and destination address update is self-increasing
   c) Data width is 32 bits
   d) Interrupt enable is valid
f. Configure trigger source controllerDMA _ TRGSEL0 and select software trigger as start request for DMA channel 0
g. Channel enable bit DMA_CHEN.CHEN0 write 1, enable channel 0

3) Write peripheral event software triggers registerINTSFTTRG.STRG to 1, sends the first software startup request, DMA starts transmitting data

2. Transmission process

Since the size of the data block is 4, the first transmission begins when INTSFTTRG.STRG is 1, and the number of transmissions is DMA_MONDTCTL0.CNT CH0CTL0 when a data block is transmitted.CNT minus 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. In the second transmission, since the size of the source address repetition area is set to 6, the source address will be reloaded into the initial address 0x2000 0000 after the transmission of 2 addresses and continue to transmit the remaining 2 addresses. After the second transfer is completed, the number of transfers DMA_MONDTCTL0.CNT is reduced by 1, and a block transfer completion interrupt is generated, and the software can continue to write INTSFTTRG.STRG in the interrupt subroutine to start the third transfer. After the third transfer is completed, the number of transfers DMA_MONDTCTL0.CNT is reduced to 0, that is, this transfer is all completed, DMA generates a block transfer completion interrupt and a transfer completion interrupt, and the channel enable bit DMA_CHEN.CHEN0 will be automatically cleared.

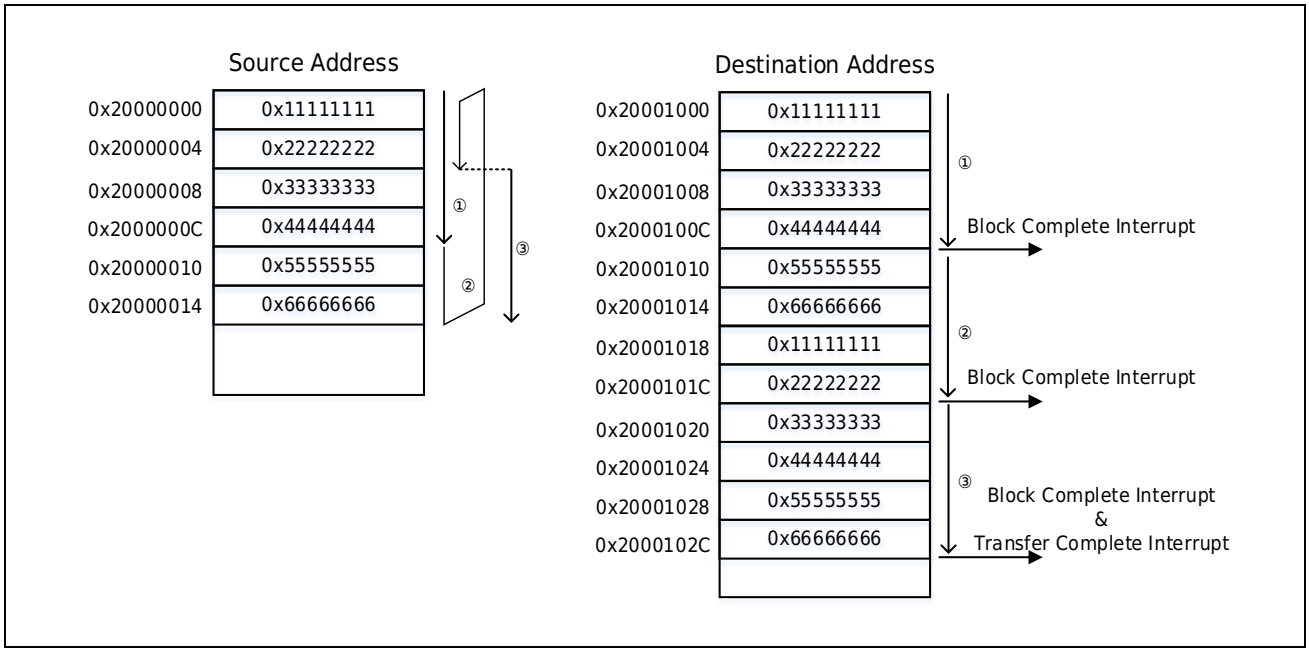


Figure 14-5 Application Example 1: Memory-to-Memory Transfer

14.5.2 Memory-to-Peripheral Transmission

Objective: Transfer 10 16-bit data from RAM address 0x2000 0000 to the communication module's Tx Buffer register. The communication module generates a send request each time it sends a data. When the last data is sent, DMA generates a transfer complete interrupt.

1. Register setting

1) Dma _ EN.EN Write 1 Enables DMAcontroller
2) Configure the DMA_INTMSK register, mask the block transfer completion interrupt, and enable the transfer completion interrupt
3) Select a channel to configure the channel register, for example, select channel 0
   a. Write DMA_SAR0 to configure the source address to SRAM area 0x2000 0000
   b. Write DMA_DAR0 to configure the target address to be the register address of the peripheral circuit 0x4000 0000
   c. Write DMA_DTCTL0Configuration data block size is 1, number of transfers is 10, each transfer request is delivered once, one data at a time.
   d. Configure channel control register DMA_CHCTL0 to achieve:
      a) Invalid link between source and destination
      b) Source address update mode is self-increment, target address is fixed
      c) The source address and destination address data width is 16 bits
      d) Interrupt enable is valid
   e. Configure trigger source controller DMA_TRGSEL0, select send register null of communication module as start request of DMA channel 0
   f. Channel enable bit DMA_CHEN.CHEN0 write 1, enable channel 0

2. Transmission 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.

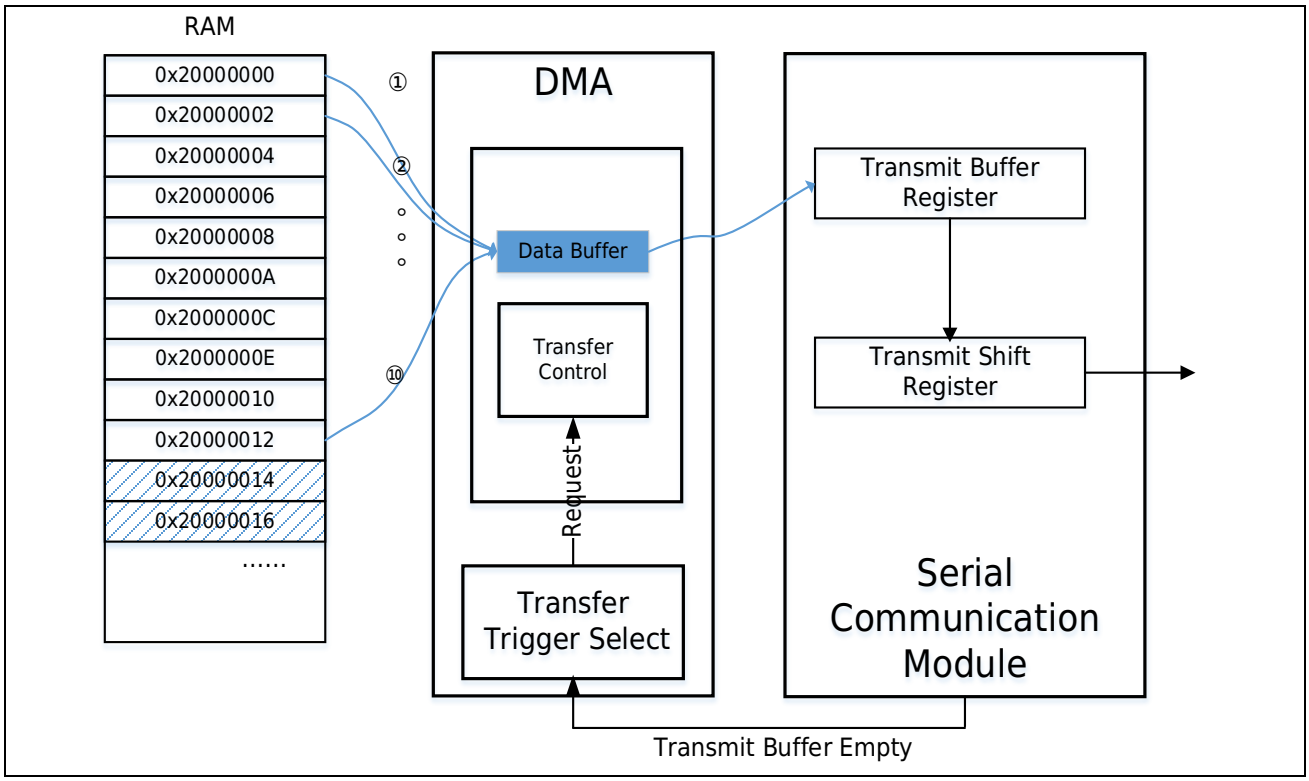


Figure 14-6 Application example 2: Transfer from memory to peripheral circuits

14.5.3 Memory-to-Memory Linkage

Objective: Transfer 3 blocks of data in RAM to 3 different addresses through chained transfers. The first block of data is 10 32-bit data, which are sent from 0x2000 0000 to 0x2000 1000. The second block of data is 20 16-bit data, from 0x2000 0100 to 0x2000 1100. The third block of data is 40 8-bit data, from 0x2000 0200 to 0x2000 1200.

1. Register Setting

1) Dma _ EN.EN Write 1 Enables DMAcontroller
2) Select a channel, configure the channel register, for example, select channel 0, configure the descriptor for the first transmission (descriptor0)
   a. Write DMA_SAR0 to configure the source address to SRAM area 0x2000 0000
   b. Write DMA_DAR0 to configure the target address to SRAM area 0x2000 1000

c. Write DMA_DTCTL0 configuration data block size is 10
d. Write the address of the second descriptor (descriptor1) in the link pointer register DMA_LLP0 0x2000 2000
e. Configure the channel control register DMA_CHCTL0, and configure the transmission parameters of the first data block to achieve:
   a) Linkage transmission efficiency
   b) Chain Transmission mode is to start the next transmission directly
   c) Source and destination address updates are self-incremented
   d) Data width is 32 bits
   e) Interrupt enable invalid

3) Configure the descriptor (descriptor1) for the second transfer at address 0x2000 2000 in the RAM space, including:
   a. 0x2000 2000 writes 32-bit data 0x2000 0100, which is the source address of the second transmission
   b. 0x2000 2004 writes 32-bit data 0x2000 1100, which is the target address of the second transmission
   c. 0x2000 The size of the configuration data block in 2008 is 20
   d. 0x2000 2018 writes 32-bit data 0x2000 2020, which is the address of the descriptor (descriptor2) of the third transmission
   e. 0x2000 201C writes the control data of the second transmission, which implements:

a) Linkage transmission efficiency
   b) Chain Transmission mode is to start the next transmission directly
   c) Source and destination address updates are self-incremented
   d) The data width is 16 bits
   e) Interrupt enable invalid

4) Configure the descriptor (descriptor2) for the third transfer at address 0x2000 2020 in the RAM space, including:
   a. 0x2000 2020 writes 32-bit data 0x2000 0200, which is the source address of the third transmission
   b. 0x2000 2024 writes 32-bit data 0x2000 1200, which is the target address of the third transmission
   c. 0x2000 2028 configuration data block size is 40
   d. Write 32-bit data 0x0 in 0x2000 2038, indicating that this transmission is the last transmission of the chain transmission.
   e. The control data for the third transmission is written in 0x2000 203C:
      a) Chain Transmission invalid
      b) Source and destination address updates are self-incremented
      c) The data width is 8 bits

d) Interrupt enable is valid

5) Configure the transfer start trigger source selection register DMA_TRGSEL0, and select the software trigger as the start request of DMA channel 0

6) Channel enable bit DMA_CHEN.CHEN0 write 1, enable channel 0

7) Write software triggers registerINTSFTTRG.STRG to 1, sends a startup request, DMA starts transmitting data

2. Transmission process

The software starts DMA and starts transmission. After the first transmission is complete, the DMA reads the descriptor (descriptor 1) of the second transmission into the channel register because the linked transmission mode is set to start the next transmission directly and the interrupt is invalid. Start the second transfer directly according to the parameters configured by the descriptor. After the second transmission is complete, read the descriptor (descriptor 2) of the third transmission into the channel register. Starts the third transfer according to the parameters configured by the descriptor. After the third transmission is complete, the last transmission is linked according to the configuration information, and because the interrupt enable is valid, DMA will generate a transmission finish interrupt and clear the channel enable bit DMA _ CHEN.CHEN0.

14.6Register Description

Table 14-2 DMA Base Address

名称	基地址	Description
DMA	0x4005 3000	DMA Base Address

Table 14-3 DMA Register List

Register	Description	Offset Address	Bit Width	Reset Value
DMA_EN	DMA enable register	0x00	32	0x0000 0000
DMA_INTSTAT0	Interrupt state register0	0x04	32	0x0000 0000
DMA_INTSTAT1	Interrupt state register1	0x08	32	0x0000 0000
DMA_INTMASK0	Interrupt shielded register0	0x0C	32	0x0000 0000
DMA_INTMASK1	Interrupt shield register1	0x10	32	0x0000 0000
DMA_INTCLR0	Interrupt reset register0	0x14	32	0x0000 0000
DMA_INTCLR1	Interrupt reset register1	0x18	32	0x0000 0000
DMA_CHEN	Channel enable register	0x1c	32	0x0000 0000
DMA_CHENCLR	Channel enable reset register	0x34	32	0x0000 0000
DMA_REQSTAT	Transfer Request Status Register	0x20	32	0x0000 0000
DMA_CHSTAT	Channel status observation register	0x24	32	0x0000 0000
DMA_RCFGCTL	Channel Reset Control Register	0x2c	32	0x0000 0000
DMA_SWREQ	Software Start Register	0x30	32	0x0000 0000
DMA_SARx (1)	Transmission source address register	0x40+0x40*x (1)	32	0x0000 0000
DMA_DARx	Transmission destination address register	0x44+0x40*x	32	0x0000 0000
DMA_DTCTLx	Data control register	0x48+0x40*x	32	0x0000 0001
DMA_RPTx	Repeat Region Size Register	0x4C+0x40*x	32	0x0000 0000
DMA_RPTBx	Repeat Region Size Register B			
DMA_SNSEQCTLx	Source address discontinuous transfer control register	0x50+0x40*x	32	0x0000 0000
DMA_SNSEQCTLBx	Source address discontinuous transfer control register B			
DMA_DNSEQCTLx	Destination address discontinuous transfer control register	0x54+0x40*x	32	0x0000 0000
DMA_DNSEQCTBx	Target address discontinuous transfer control register B			
DMA_LLPx	Chain pointer register	0x58+0x40*x	32	0x0000 0000
DMA_CHCTLx	Channel control register	0x5C+0x40*x	32	0x0000 1000
DMA_MONSARx	Transfer Source Address Monitor Register	0x60+0x40*x	32	0x0000 0000
DMA_MONDARx	Transfer Destination Address Monitor Register	0x64+0x40*x	32	0x0000 0000
DMA_MONDTCTLx	Data Control Monitor Register	0x68+0x40*x	32	0x0000 0001
DMA_MONRPTx	Repeat Region Counter Monitor Register	0x6C+0x40*x	32	0x0000 0000
DMA_MONSNSEQCTLx	Source address discontinuous transfer counter monitoring register	0x70+0x40*x	32	0x0000 0000

Register	Description	Offset Address	Bit Width	Reset Value
DMA_MONDNSEQCTLx	Destination Address Discontinuous Transfer Counter Monitor Register	0x74+0x40*x	32	0x0000 0000

Note: *(1) x = 0 to 7.*

14.6.1 DMA Enable Register (DMA _ EN)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res															EN

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b1	Res	Reserved Bit	Read at “0”, write at “0”	RW
b0	EN	DMA enable bit	0: DMA invalid 1: DMA enable	RW

14.6.2 Interrupt Status Register 0 (DMA_INTSTAT0)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res								REQERR[7:0]							
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res								TRNERR[7:0]							

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b24	Res	Reserved Bit	Read at “0”, write at “0”	RW
b23~b16	REQERR[7:0]	Transmission request overflow error interrupt bit	0: No transmission request overflow error occurred on the channel 1: The transmission request overflow error occurred on the channel, that is, the transmission request came again when the last request was still waiting	R
b15~b8	Res	Reserved Bit	Read at “0”, write at “0”	RW
b7~b0	TRNERR[7:0]	Transmission error interrupt bit	0: No transmission error occurred on the channel 1: The channel has a transmission error	R

14.6.3 Interrupt Status Register 1 (DMA_INTSTAT1)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res								BTC[7:0]							
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res								TC[7:0]							

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b24	Res	Reserved Bit	Read at “0”, write at “0”	RW
b23~b16	BTC[7:0]	Block transport complete interrupt bit	The interrupt occurs after a block of data is transmitted 0: No block transport interrupt occurs on the channel 1: Block transport interrupt of this channel	R
b15~b8	Res	Reserved Bit	Read at “0”, write at “0”	RW
b7~b0	TC[7:0]	Transport completion interrupt bit	The interrupt occurs after the number of transmissions set by registerMA _ CNTx is completed 0: The channel was not interruptd 1: The channel is interruptd	R

14.6.4 Interrupt Mask Register 0 (DMA_INTMASK0)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res								MSKREQERR[7:0]							
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res								MSKTRNERR[7:0]							

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b24	Res	Reserved Bit	Read at “0”, write at “0”	RW
b23~b16	MSKREQERR[7:0]	Transmission request overflow interrupt mask	0: Nomaskable transport request overflow interrupt 1: Shielded transport request overflow interrupt	RW
b15~b6	Res	Reserved Bit	Read at “0”, write at “0”	RW
b7~b0	MSKTRNERR[7:0]	Transmission error interrupt mask	0: Nomaskable transport error interrupt 1: Mask transmission error interrupt	RW

14.6.5 Interrupt Mask Register 1 (DMA_INTMASK1)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res								MSKBTC[7:0]							
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res								MSKTC[7:0]							

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b24	Res	Reserved Bit	Read at “0”, write at “0”	RW
b23~b16	MSKBTC[7:0]	Block transport complete interrupt mask	0: Nomaskable block transport complete interrupt 1: Block transport complete interrupt	RW
b15~b8	Res	Reserved Bit	Read at “0”, write at “0”	RW
b7~b0	MSKTC[7:0]	Transport complete interrupt mask	0: Nomaskable transport complete interrupt 1: Shield transport complete interrupt	RW

14.6.6 Interrupt Reset Register 0 (DMA_INTCLR0)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res								CLRREQERR[7:0]							
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res								CLRTRNERR[7:0]							

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b24	Res	Reserved Bit	Read at “0”, write at “0”	RW
b23~b16	CLRREQERR[7:0]	Transmission request overflow error interrupt reset	Write 0 has no effect. Write 1 Resets the transmission request overflow error interrupt status bit. Read always 0	W
b15~b8	Res	Reserved Bit	Read at “0”, write at “0”	RW
b7~b0	CLRTRNERR[7:0]	Transmission error interrupt reset	Write 0 has no effect, write 1 reset transmission error interrupt status bit Read always 0	W

14.6.7 Interrupt Reset Register 1 (DMA_INTCLR1)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res								CLRBTC[7:0]							
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res								CLRTC[7:0]							

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b24	Res	Reserved Bit	Read at “0”, write at “0”	RW
b23~b16	CLRBTC[7:0]	Block transport complete interrupt reset	Write 0 has no effect, write 1 reset block transport complete interrupt status bit Read always 0	W
b15~b8	Res	Reserved Bit	Read at “0”, write at “0”	RW
b7~b0	CLRTC[7:0]	Transmission complete interrupt reset	Write 0 has no effect. Write 1 resets the transmission to finish terminating the status bit. Read always 0	W

14.6.8 Channel Enable Register (DMA _ CHEN)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res								CHEN[7:0]							

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b8	Res	Reserved Bit	Read at “0”, write at “0”	RW
b7~b0	CHEN[7:0]	Channel enable bit	Each bit corresponds to a channel. 0: Invalid channel 1: The channel is valid Write 1 for CHEN [x] and set 1 for CHEN [x]. Invalid write 0 for CHEN [x]. 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 transmission, and it will be automatically cleared to zero after the number of transmissions set by the transmission times register DMA_DTCTLx.CNT is completed. If DMA_DTCTLx.CNT is set to 0, it will not be automatically cleared after the transfer is completed, that is, unlimited transfers. **Note:** After all 8 descriptor registers of channel x are set, please write 1 to CHEN[x] to enable this channel. When CHEN[x] is 1, write operations to the descriptor register of channel x have no effect.	RW

14.6.9 Channel Enable Reset Register (DMA _ CHENCLR)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res								CHENCLR[7:0]							

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b8	Res	Reserved Bit	Read at “0”, write at “0”	RW
b7~b0	CHENCLR[7:0]	Channel enable bit reset	Write 0 has no effect, write 1 reset channel enable bit CHEN Read always 0 If this channel is being transferred at write 1, DMA will force the transfer to interrupt after the current data read/write is completed.	W

14.6.10 Channel Reset Control Register (DMA_RCFGCTL)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res										CNTMD[1:0]		DARMD[1:0]		SARMD[1:0]	
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res				RCFGCHS[3:0]				Res						RCFGLLP	RCFGEN

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b22	Res	Reserved Bit	Read at “0”, write at “0”	RW
b21~b20	CNTMD[1:0]	Remaining transfer count counter reset method	00: remain unchanged 01: By source address 10, 11: by destination address When the source address mode is selected, and the source address selects discontinuous reset, the remaining transfer times counter is updated to the state after the specified transfer times of DMA_SNSEQCTLBx.SNSCNTB; when the source address is repeated, the remaining transfer times count er is updated to DMA_RPTBx. SRPTB The state after the specified number of transfers. When the source address selection is held, the remaining transfer times counter also remains unchanged. When the target address mode is selected, it is similar to the source address mode.	RW
b19~b18	DARMD[1:0]	Target address reset method	00: remain unchanged 01: Discontinuous reset The destination address for the next transfer is updated to addr_base + (DMA_DNSEQCTLBx.DNSDIST x DMA_CHCTLx.HSIZE (bit)/8) Among them: addr_base indicates the first address of the current discontinuous transmission area 10, 11: Repeated reset The target address of the next transfer is updated to the initial setting value of the DMA_DARx register. **Note:** When the discontinuous transfer of the target address of this channel valid (DMA_CHCTLx.DNSEQEN=1), DARMD[1:0] must be set to 0b01.	RW
b17~b16	SARMD[1:0]	Source address reset method	00: remain unchanged 01: Discontinuous reset The source address of the next transfer is updated to addr_base + (DMA_SNSEQCTLBx.SNSDIST x DMA_CHCTLx.HSIZE (bit)/8) Among them: addr_base indicates the first address of the current discontinuous transmission area 10, 11: Repeated reset The source address of the next transfer is updated to the initial setting value of the DMA_SARx register. **Note:** When the discontinuous transfer of the source address of this channel valid (DMA_CHCTLx.SNSEQEN=1), SARMD[1:0] must be set to 0b01.	RW
b15~b12	Res	Reserved Bit	Read at “0”, write at “0”	RW
b11~b8	RCFGCHS[3:0]	Reset channel selection	0x0: channel 0 0x1: channel 1 And so on 0x5: channel 5 Other: Prohibitions	RW
b7~b2	Res	Reserved Bit	Read at “0”, write at “0”	RW
b1	RCFGLLP	Chain Pointer Channel Reset	0: Chain pointer reset is invalid 1: Chain pointer reset is valid **Note:** When RCFGPLL is set to 1, the channel will reload the new descriptor in the memory, so bit16-bit25 of this register are all invalid.	RW
b0	RCFGEN	Channel Reset Permission	0: Disable event-triggered channel configuration register forced update 1: Enable event-triggered channel configuration register forced update	RW

Note:

Please set this register when DMA_EN.EN is 0, this register must be set before the first transmission of the reset channel.

14.6.11 Software Start Register (DMA_SWREQ)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
SWRCFGWP[7:0]								SWREQWP[7:0]							
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
SWRCFGREQ	Res							SWREQ[7:0]							

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b24	SWRCFGWP[7:0]	Channel reset request write protection	The software reset requests write protection, and the SWRCFGWP write value is 0xA2 to release the write protection.	W
b23~b16	SWREQWP[7:0]	Channel reset request write protection	The software triggers a write protection request, and the SWREQWP write value is 0xA1 to release the write protection.	W
b15	SWRCFGREQ	Software channel reset request	The software trigger channel reset function has the same effect as the AOS trigger. 0b0: Writing 0 invalid 0b1: When SWRCWP writes 0xA2 and SWRCFGREQ writes a value of 1, a software channel reset request is generated	W
b14~b8	Res	Reserved Bit	When reading, it is "0", when writing, please write "0"	RW
b7~b0	SWREQ[7:0]	Software channel transfer request	The software trigger starts and sends, each bit corresponds to one channel, which is the same as the AOS trigger effect. 0b0: Writing 0 invalid 0b1: When SWREQWP writes 0xA1 and SWREQ[x] writes a value of 1, the channel generates a software channel reset request	W

14.6.12 Transfer Request Status Register (DMA_REQSTAT)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															RCFGREQ
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res								CHREQ[7:0]							

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b17	Res	Reserved Bit	Read at “0”, write at “0”	RW
b16	RCFGREQ	Channel reset request flag	When the external reset request input is set to 1, Cleared to 0 when channel reset is enabled, or when channel reset is disabled. 0: No channel reset request 1: There is a channel reset request	R
b15~b8	Res	Reserved Bit	Read at “0”, write at “0”	RW
b7~b0	CHREQ[7:0]	channel transfer request flag	Each bit corresponds to one channel. When the external transmission request is input, the corresponding position is 1, This bit is cleared to 0 when a transmission error occurs after the channel transmission is started, or when the transmission permission bit (DMAEN or CHEN[x]) is written to 0. When this bit is in the 1 state and the channel transmission request is input again, a transmission request overflow error occurs and the second request is ignored, but this flag bit remains 1 and continues to wait for transmission. 0: There is no transmission request for this channel 1: The channel has a transfer request	R

14.6.13 Channel State Observation Register (DMA _ CHSTAT)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res								CHACT[7:0]							
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res														RCFGACT	DMAACT

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b24	Res	Reserved Bit	Read at “0”, write at “0”	RW
b23~b16	CHACT[7:0]	Channel monitoring bit	Each bit corresponds to one channel. 0: The channel is idle 1: The channel is in motion	R
b15~b2	Res	Reserved Bit	Read at “0”, write at “0”	RW
b1	RCFGACT	DMA channel reset action monitoring bit	0: DMA is not in channel reset action 1: DMA is in channel reset action	R
b0	DMAACT	Monitoring bit in DMA action	0: DMA not in transmission 1: DMA in transmission	R

14.6.14 Transfer Source Address Register (DMA_SARx, x=0~7)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
SAR[31:16]															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
SAR[15:0]															

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b0	SAR[31:0]	Transmission source address	Set source address **Note:** When the transmission data width is 16 bits, that is, DMA_CHCTLx.HSIZE=0b01, SAR[0] is invalid. When the transmission data width is 32 bits, that is, DMA_CHCTLx.HSIZE=0b1x, SAR[1:0] is invalid.	RW

14.6.15 Transfer Destination Address Register (DMA_DARx, x=0~7)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
DAR[31:16]															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
DAR[15:0]															

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b0	DAR[31:0]	Transmission destination address	Set transmission destination address **Note:** When the transmission data width is 16 bits, that is, DMA_CHCTLx.HSIZE=0b01, DAR[0] is invalid. When the transmission data width is 32 bits, that is, DMA_CHCTLx.HSIZE=0b1x, DAR[1:0] is invalid.	RW

14.6.16 Data Control Register (DMA_DTCTLx, x=0~7)

Reset value: 0x0000 0001

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
CNT[15:0]															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res						BLKSIZE[9:0]									

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b16	CNT[15:0]	Number of Transmissions	The total number of transfers, each request starts the transfer of a data block, the number of transfers is decremented by 1 when it is completed, and a transfer completion interrupt occurs when it is reduced to 0. If it is set to 0, it means unlimited transmission. Each time a data block is requested to be transmitted, the transmission times counter will remain at 0 when it is completed, and no transmission completion interrupt will be generated.	RW
b15~b10	Res	Reserved Bit	Read at “0”, write at “0”	RW
b9~b0	BLKSIZE[9:0]	Size of data block	Set the size of the data block, up to 1024 pieces of data can be configured. The width of each data is determined by the HSIZE bit of the DMA _ CHxCTLregister. If the register value is set to 1, 1 data is transmitted each time, and if it is set to 0, 1024 data are transmitted each time.	RW

14.6.17 Repeat Region Size Register (DMA_RPTx, x=0~7)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res						DRPT[9:0]									
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res						SRPT[9:0]									

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b26	Res	Reserved Bit	Read at “0”, write at “0”	RW
b25~b16	DRPT[9:0]	Destination address repeat area size	Set target address repeat area size The target device reloads the target address to the initial setting value of the DMA_DARx register after each DRPT data transfer. If the register is set to 10, the address will be reloaded after every 10 data transmissions, and if it is set to 0, the address will be reloaded after every 1024 data transmissions. The width of each data is determined by DMA_CHCTLx.HSIZE.	RW
b15~b10	Res	Reserved Bit	Read at “0”, write at “0”	RW
b9~b0	SRPT[9:0]	Source address duplication area size	Set the source address repeat area size The source device reloads the source address to the initial setting value of the DMA_SARx register after each transmission of SRPT data. If the register is set to 10, the address will be reloaded after every 10 data transmissions, and if it is set to 0, the address will be reloaded after every 1024 data transmissions. The width of each data is determined by DMA_CHCTLx.HSIZE.	RW

Note:

1. *This register configures the size of the source and destination address duplication region. The use of repeated address transmission needs to configure the SRPTEN/DRPREN bit of the DMA_CHCTLx register to be valid, and configure the SINC/ DINC bit of the DMA_CHCTLx register to make the address update mode self-increment or self-decrement. If it is fixed, the address reload function is invalid.*
2. *This register invalid when the reset function of this channel is turned on.*

14.6.18 Repeat Region Size Register (DMA_RPTBx, x=0~7)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res						DRPTB[9:0]									
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res						SRPTB[9:0]									

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b26	Res	Reserved Bit	Read at “0”, write at “0”	RW
b25~b16	DRPTB[9:0]	Destination address repeat area size	Set target address repeat area size The target device reloads the target address to the initial setting value of the DMA_DARx register after each transmission of DRPTB data blocks. The data block size is determined by DMA_DTCTLx.BLKSIZE and DMA_CHCTLx.HSIZE. **Note:** When the reset function of this channel valid (DMA_RCFGCTL.RCFGEN=1), the setting value of DRPTB[15:0] must be greater than 0	RW
b16~b10	Res	Reserved Bit	Read at “0”, write at “0”	RW
b9~b0	SRPTB[9:0]	Source address duplication area size	Set the source address repeat area size The source device reloads the source address to the initial setting value of the DMA_SARx register after each transmission of SRPTB data blocks. The data block size is determined by DMA_DTCTLx.BLKSIZE and DMA_CHCTLx.HSIZE. **Note:** When the reset function of this channel valid (DMA_RCFGCTL.RCFGEN=1), the setting value of SRPTB[9:0] must be greater than 0.	RW

Note:

1. *This register configures the size of the source and destination address duplication region. The use of repeated address transmission needs to configure the SRPTEN/ DRPREN bit of the DMA_CHCTLx register to be valid, and configure the SINC/ DINC bit of the DMA_CHCTLx register to make the address update mode self-increment or self-decrement. If it is fixed, the repeated address transmission function is invalid.*
2. *This register is only valid when the reset function of this channel is turned on, replacing the register DMA_RPTx. It has no effect when the channel reset function disable.*

14.6.19 Source Address Discontinuous Transfer Control Register (DMA_SNSEQCTLx, x=0~7)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
SNSCNT[11:0]												SOFFSET[19:16]			
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
SOFFSET[15:0]															

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b20	SNSCNT[11:0]	The amount of data redirected by the source address	Set the amount of data transferred before the source address jumps. After the source device transmits SNSCNT data, the source address jumps according to the offset specified by SOFFSET. If the register is set to 10, the address will jump after every 10 data transmissions, if it is set to 0, the address will jump after every 4096 data transmissions.	RW
b19~b0	SOFFSET[19:0]	The address offset of the source address jump	When discontinuous address transmission, set the offset of the source address jump. The offset is relative to the current transfer address, that is, the last transfer address before the jump. The jump direction jumps forward or backward according to the value of the channel control register DMA_CHCTLx.SINC. Reference Figure 14-3. When DMA_CHCTLx.SINC is set to fixed address, discontinuous address transfer is invalid. The jump address will be calculated according to the number of bits set by the data width (DMA_CHCTLx.HSIZE) and the value of SOFFSET. Address offset = SOFFSET × (DMA_CHCTLx.HSIZE (bit) / 8) For example, when SOFFSET is set to 10 and the data width is 32 bits (DMA_CHCTLx.HSIZE = 0b1x), the address offset is 10×4=40. If the data width is 16 bits half-word (DMA_CHCTLx.HSIZE = 0b01), the offset is 10×2=20. If the data width is 8 bits (DMA_CHCTLx.HSIZE = 0b00), the offset is 10×1=10. The source address of the next transfer = the source address of the current transfer ± address offset	RW

Note:

1. *To use discontinuous transfer of the source device, you need to configure the SNSEQEN bit of the DMA_CHCTLx register to be valid, and configure the SINC bit of the DMA_CHCTLx register to make the address update mode self-increment or self-decrement.*
2. *This register invalid when the reset function of this channel is turned on.*

14.6.20 Source Address Discontinuous Transfer Control Register B (DMA_SNSEQCTLBx, x=0~7)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
SNSCNTB[11:0]												SNSDIST[19:16]			
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
SNSDIST[15:0]															

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b20	SNSCNTB[11:0]	The amount of data redirected by the source address	Set the amount of data transferred before the source address jumps. After the source device transmits SNSCNTB data blocks, the source address jumps according to the address interval specified by SNSDIST. The data block size is determined by DMA_DTCTLx.BLKSIZE and DMA_CHCTLx.HSIZE. **Note:** When the reset function of this channel valid (DMA_RCFGCTL.RCFGEN=1), the setting value of SNSCNTB[11:0] must be greater than 0.	RW
b19~b0	SNSDIST[19:0]	Source discontinuous area address spacing	When discontinuous address transmission, set the distance between two discontinuous areas of the source device. The jump direction jumps forward or backward according to the value of the channel control register DMA_CHCTLx.SINC. Reference Figure 14-4. When DMA_CHCTLx.SINC is set to fixed address, discontinuous address transfer is invalid. The address spacing will be calculated based on the number of bits set by the data width (DMA_CHCTLx.HSIZE) and the value of SNSDIST. Address spacing = SNSDIST × (DMA_CHCTLx.HSIZE (bit) / 8) For example, when SNSDIST is set to 10 and the data width is 32 bits (DMA_CHCTLx.HSIZE = 0b1x), the address spacing is 10×4=40. If the data width is 16 bits (DMA_CHCTLx.HSIZE = 0b01), the spacing is 10×2=20. If the data width is 8 bits (DMA_CHCTLx.HSIZE = 0b00), the spacing is 10×1=10. The source address of the next transmission = the first address of the current source discontinuous area ± address distance	RW

Note:

1. *To use discontinuous transfer of the source device, you need to configure the SNSEQEN bit of the DMA_CHCTLx register to be valid, and configure the SINC bit of the DMA_CHCTLx register to make the address update mode self-increment or self-decrement.*
2. *This register is only valid when the reset function of this channel is turned on, replacing the register DMA_SNSEQCTLx. It has no effect when the channel reset function disable.*

14.6.21 Target Address Discontinuous Transfer Control Register (DMA_DNSEQCTLx, x=0~7)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
DNSCNT[11:0]												DOFFSET[19:16]			
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
DOFFSET[15:0]															

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b20	DNSCNT[11:0]	The amount of data that the target address jumps to	Set the amount of data transferred before the target address jumps. After the target device transmits DNSCNT data, the target address jumps according to the offset specified by DOFFSET. If the register is set to 10, the address will jump after every 10 data transmissions, if it is set to 0, the address will jump after every 4096 data transmissions.	RW
b9~b0	DSOFFSET[19:0]	The address offset of the target address jump	When discontinuous address transfer, set the offset of the target address jump. The offset is relative to the current transfer address, that is, the last transfer address before the jump. The direction of the jump jumps forward or backward according to the value of the channel control register DMA_CHCTLx.DINC. Reference Figure 14-3. When DMA_CHCTLx.DINC is set to fixed address, discontinuous address transfer will be invalid. The jump address will be calculated according to the number of bits set by the data width (DMA_CHCTLx.HSIZE) and the value of DOFFSET. Address offset = DOFFSET × (DMA_CHCTLx.HSIZE (bit) / 8) For example, when DOFFSET is set to 10, if the data width is 32 bits (DMA_CHCTLx.HSIZE=0b1x), the address offset is 10×4=40; if the data width is 16 bits (DMA_CHCTLx.HSIZE=0b01), the offset is 10×2=20; if the data width is 8 bits (DMA_CHCTLx.HSIZE=0b00), the offset is 10×1=10. The target address of the next transfer = the target address of the current transfer ± address offset	RW

Note:

1. *To use discontinuous transmission of the target device, you need to configure the DNSEQEN bit of the DMA_CHCTLx register to be valid, and configure the DINC bit of the DMA_CHCTLx register to make the address update mode self-increment or self-decrement.*
2. *This register invalid when the reset function of this channel is turned on.*

14.6.22 Target Address Discontinuous Transfer Control Register B (DMA_DNSEQCTLBx, x=0~7)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
DNSCNTB[11:0]												DNSDIST[19:16]			
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
DNSDIST[15:0]															

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b20	DNSCNTB[11:0]	The amount of data that the target address jumps to	Set the amount of data transferred before the target address jumps. After the target device transmits DNSCNTB data blocks, the target address jumps according to the address interval specified by DNSDIST. The data block size is determined by DMA_DTCTLx.BLKSIZE and DMA_CHCTLx.HSIZE. **Note:** When the reset function of this channel valid (DMA_RCFGCTL.RCFGEN=1), the setting value of DNSCNTB[11:0] must be greater than 0.	RW
b19~b0	DNSDIST[19:0]	Target Discontinuous Area Address Spacing	When discontinuous address transmission, set the distance between two discontinuous areas of the target device. The direction of the jump jumps forward or backward according to the value of the channel control register DMA_CHCTLx.DINC. Reference Figure 14-4. When DMA_CHCTLx.DINC is set to fixed address, discontinuous address transfer will be invalid. The address spacing will be calculated based on the number of bits set by the data width (DMA_CHCTLx.HSIZE) and the value of DNSDIST. Address spacing = DNSDIST × (DMA_CHCTLx.HSIZE (bit) / 8) For example, when DNSDIST is set to 10 and the data width is 32 bits (DMA_CHCTLx.HSIZE=0b1x), the address spacing is 10×4=40. If the data width is 16 bits (DMA_CHCTLx.HSIZE=0b01), the spacing is 10×2=20. If the data width is 8 bits (DMA_CHCTLx.HSIZE=0b00), the spacing is 10×1=10. The target address of the next transmission = the first address of the current target discontinuous area ± address distance	RW

Note:

1. *To use discontinuous transmission of the target device, you need to configure the DNSEQEN bit of the DMA_CHCTLx register to be valid, and configure the DINC bit of the DMA_CHCTLx register to make the address update mode self-increment or self-decrement.*
2. *This register is only valid when the reset function of this channel is turned on, replacing the register DMA_DNSEQCTLx. It has no effect when the channel reset function disable.*

14.6.23 Link Pointer Register (DMA_LLPx, x=0~7)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
LLP[31:16]															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
LLP[15:2]														Res	

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b2	LLP[31:2]	Chain pointer	When the chain transmission is valid, set the address of the next transmission descriptor to word alignment, that is, LLP [1: 0] is fixed to 0	RW
b1~b0	Res	Reserved Bit	Read at “0”, write at “0”	RW

14.6.24 Channel Control Register (DMA_CHCTLx, x=0~7)

Reset value: 0x0000 1000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res			IE	LLPRUN	LLPEN	HSIZE[1:0]		DNSEQEN	SNSEQEN	DRPTEN	SRPTEN	DINC[1:0]		SINC[1:0]	

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b13	Res	Reserved Bit	Read at “0”, write at “0”	RW
b12	IE	Interrupt enable bit	Configure whether the channel generates an interrupt. 0: This channel does not generate an interrupt 1: The channel generates an interrupt	RW
b11	LLPRUN	Linkage mode selection	When the chain transfer is active, set whether to start the transfer corresponding to the new descriptor immediately after loading the new descriptor specified by the chain pointer when the current transfer is complete 0: Do not transmit immediately, wait for the next transmission request to start 1: Start transmission immediately after the new descriptor is loaded	RW
b10	LLPEN	Chain Transmission enable	0: Invalid chain transmission 1: Linkage transmission is effective	RW
b9~b8	HSIZE[1:0]	Width of transmitted data	00: 8bit 01: 16bit 10~11: 32bit	RW
b7	DNSEQEN	Destination Address Discontinuous Transfer Enable	0: Do not allow discontinuous address transmission 1: Allow discontinuous address transmission	RW
b6	SNSEQEN	Source address discontinuous transfer enable	0: Do not allow discontinuous address transmission 1: Allow discontinuous address transmission	RW
b5	DRPTEN	Target repeat transfer function enable bit	Set whether to allow the target address to reload the initial value 0: no reload 1: Overload	RW
b4	SRPTEN	Source repeat transfer function enable bit	Set whether to allow the source address to reload the initial value 0: no reload 1: Overload	RW
b3~b2	DINC[1:0]	Update mode of destination address	00: Fixed 01: Incremental 10,11: decrement	RW
b1~b0	SINC[1:0]	Source address update	00: Fixed 01: Incremental 10,11: decrement	RW

14.6.25 Transfer Source Address Monitoring Register (DMA_MONSARx, x=0~7)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
SAR[31:16]															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
SAR[15:0]															

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b0	SAR[31:0]	Transmission source address	Transmission source address. **Note:** When the transmission data width is 16 bits (DMA_CHCTLx.HSIZE=0b01), SAR[0] invalid. When the transmission data width is 32 bits (DMA_CHCTLx.HSIZE=0b10 or 0b11), SAR[1:0] invalid.	R

Note:

The transfer source address register (DMA_SARx) remains unchanged before and after DMA transfer, while the transfer source address monitor register (DMA_MONSARx) is updated after DMA completes each data block transfer. The update content and method are as follows: DMA_MONSARx.SAR[31:0] is updated to the address of the next transfer according to the fixed/incrementing/decrementing/reload/discontinuous jump mode set by the register.

14.6.26 Transfer Target Address Monitoring Register (DMA_MONDARx, x=0~7)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
DAR[31:16]															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
DAR[15:0]															

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b0	DAR[31:0]	Transmission destination address	Transmission destination address **Note:** When the transmission data width is 16 bits (DMA_CHCTLx.HSIZE=0b01), DAR[0] invalid. When the transmission data width is 32 bits (DMA_CHCTLx.HSIZE=0b10 or 0b11), DAR[1:0] invalid.	R

Note:

The transfer destination address register (DMA_DARx) remains unchanged before and after DMA transfer, while the transfer destination address monitor register (DMA_MONSARx) is updated after DMA completes each data block transfer. The update content and method are as follows: DMA_MONDARx.DAR[31:0] is updated to the address of the next transfer according to the fixed/incrementing/decrementing/reload/discontinuous jump method set by the register.

14.6.27 (DMA_MONDTCTLx, x=0~7)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
CNT[15:0]															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res						BLKSIZE[9:0]									

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b16	CNT[15:0]	Number of Transmissions	The number of transfers remaining. Note: When the transfer count is 0 (DMA_DTCTLx.CNT = 0x0000), this bit field remains 0.	R
b15~b10	Res	Reserved Bit	Read at “0”, write at “0”	R
b9~b0	BLKSIZE[9:0]	Size of data block	Size of data block Note: When this bit field is 0, it means 1024 data are sent each time.	R

14.6.28 Repeat Region Counter Monitoring Register (DMA_MONRPTx, x = 0 ~ 7)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res						DRPT[9:0]									
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res						SRPT[9:0]									

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b26	Res	Reserved Bit	Read at “0”, write at “0”	RW
b25~b16	DRPT[9:0]	Destination address repeat area size	Destination address repeat area size.	R
b16~b10	Res	Reserved Bit	Read at “0”, write at “0”	RW
b9~b0	SRPT[9:0]	Source address duplication area size	Source Address Duplication Area Size.	R

Note:

The repeat region size register (DMA_RPTx) remains unchanged before and after DMA transfer, while the repeat region count control monitoring register (DMA_MONRPTx) is updated after DMA completes each data block transfer. The update content and method are as follows: DMA_MONRPTx.SRPT[9:0], DRPT[9:0]: When the channel reset invalid, subtract the block size, and reload the DMA_RPTx setting value when it is reduced to 0. When reset is effective, it decreases by 1, and when it decreases to 0, the original setting value is reloaded.

14.6.29 Source Address Discontinuous Transfer Counter Monitoring Register (DMA_MONSNSEQCTLx, x = 0 ~ 7)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
SNSCNT[11:0]												SOFFSET[19:16]			
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
SOFFSET[15:0]															

b31~b20	SNSCNT[11:0]	The amount of data redirected by the source address	The amount of data sent before the source address jumps.	R
b19~b0	SOFFSET[19:0]	The address offset of the source address jump	The offset to jump to from the source address.	R

Note:

The source device discontinuous address transfer control register (DMA_SNSEQCTLx) remains unchanged before and after DMA transfer, while the source device discontinuous transfer counter monitor register (DMA_MONSNSEQCTLx) is updated after DMA completes each data block transfer. The update content and method are as follows: When the DMA_MONSNSEQCTLx.SNSCNT[11:0] channel reset invalid, the block size is subtracted, and when it is reduced to 0, the original setting value of DMA_SNSEQCTLx is reloaded. When reset is effective, it decreases by 1, and when it decreases to 0, the original setting value is reloaded.

14.6.30 Target Address Discontinuous Transfer Counter Monitoring Register (DMA_MONDNSEQCTLx, x = 0 ~ 7)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
DNSCNT[11:0]												DOFFSET[19:16]			
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
DOFFSET[15:0]															

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b20	DNSCNT[11:0]	The amount of data that the target address jumps to	The amount of data transferred before the target address jumps.	R
b9~b0	DSOFFSET[19:0]	The address offset of the target address jump	The target address to jump to.	R

Note:

The target device discontinuous address transfer control register (DMA_DNSEQCTLx) remains unchanged before and after DMA transfer, while the target device discontinuous transfer counter monitoring register (DMA_MONDNSEQCTLx) is updated after DMA completes each data block transfer. The update content and method are as follows: When the DMA_MONDNSEQCTLx.SNSCNT[11:0] channel reset invalid, the block size is subtracted, and when it is reduced to 0, the original setting value of DMA_DNSEQCTLx is reloaded. When reset is effective, it decreases by 1, and when it decreases to 0, it reloads the original setting value

14.7 Precautions

- DMA registers only support 32bit read and write, and 8/16bit read and write operations are invalid.

15 Voltage Comparator (CMP)

15.1 Overview

A voltage comparator (CMP) is a peripheral module that compares two analog voltages and outputs the comparison result. This product is equipped with two groups of 3 comparison channels: CMP1/CMP2, CMP3.

15.2 Main Features

- 3 comparable channels can be independently compared
- The combination of CMP1/CMP2 comparison channels can realize window comparison
- Each comparison channel has multiple input sources (IO/DAC) for selection of positive/negative voltage
- Configurable hysteresis voltage
- Noise filter can filter the comparator output, 7 sampling clocks are optional
- Comparator blank window control can be performed using timer PWM
- Can generate interrupts on the changing edge of the comparison result, trigger other peripherals, and wake up the Stop Mode
- The comparison result can be monitored through the register and can also be output to the external pin VCOUT
- Comparison results can be used for emergency brake (EMB) control events
- The comparison result can be used as an external event source for HRPWM
- The comparator output keeps on when software reset and watchdog reset occur.

15.3Functional Description

15.3.1 CMP Functional Block Diagram

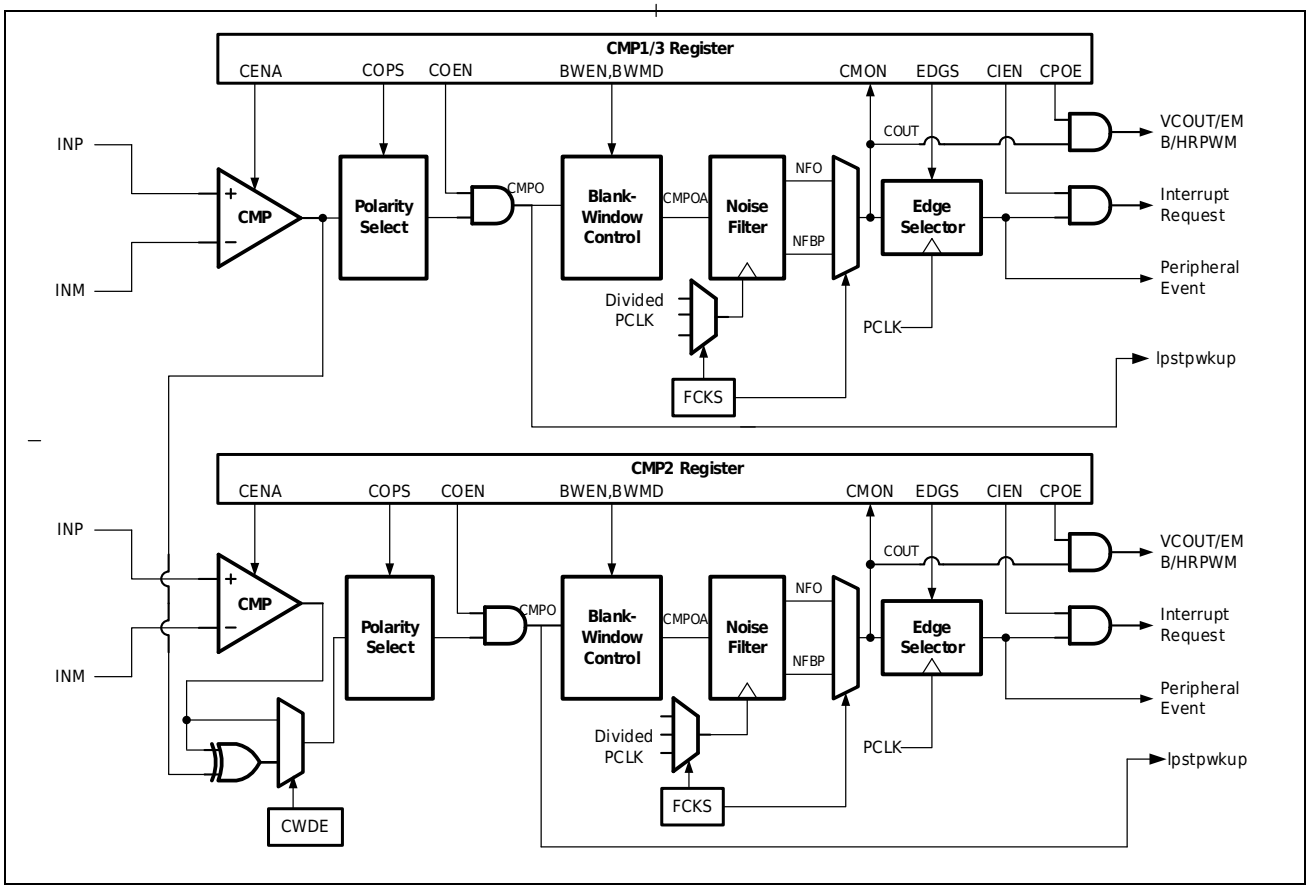


Figure 15-1 CMP Functional Block Diagram

Table 15-1 CMP Pin List

Pin Name	Input/Output	Function
AVCC	Input	Analog power
AVSS	Input	Analog ground
CMP1_INP1	Input	CMP1 Positive Analog Input
CMP123_INP2	Input	CMP1, 2, 3 Positive Analog Input
CMP123_INM1	Input	CMP1, 2, 3 Negative Analog Input
CMP1_INM2	Input	CMP1 Negative Analog Input
CMP2_INP2	Input	CMP2 Positive Analog Input
CMP2_INM2	Input	CMP2 Negative Analog Input
CMP3_INP2	Input	CMP3 Positive Analog Input
CMP3_INM2	Input	CMP3 Negative Analog Input
VCOUT1	Output	CMP1 External Port Output
VCOUT2	Output	CMP2 External Port Output

Pin Name	Input/Output	Function
VCOUT3	Output	CMP3 External Port Output
VCOUT	Output	External Port Output of CMP1~3

15.3.2 Input Selection

The analog inputs of CMP1~3 are selected according to the following table.

Table 15-2 CMP Analog Input List

CMP1				
SN	Register PMSR.CVSL	Positive Input INP	Register PMSR.RVSL	Negative Input INM
1	0b01	CMP1_INP1	0b00001	CMP123_INM1
2	0b10	CMP123_INP2	0b00010	CMP1_INM2
3	-	-	0b00100	DA1O1
4	-	-	0b01000	DA1O2
5	-	-	0b10000	DA2O1
CMP2				
SN	Register PMSR.CVSL	Positive Input INP	Register PMSR.RVSL	Negative Input INM
1	0b01	CMP2_INP1	0b00001	CMP123_INM1
2	0b10	CMP123_INP2	0b00010	CMP2_INM2
3	-	-	0b00100	DA1O1
4	-	-	0b01000	DA1O2
5	-	-	0b10000	DA2O1
CMP3				
SN	Register PMSR.CVSL	Positive Input INP	Register PMSR.RVSL	Negative Input INM
1	0b01	CMP3_INP1	0b00001	CMP123_INM1
2	0b10	CMP123_INP2	0b00010	CMP3_INM2
3	-	-	0b00100	DA1O1
4	-	-	0b01000	DA1O2
5	-	-	0b10000	DA2O1

15.3.3 Normal Comparison

CMP1~3 can perform common comparison independently. Taking CMP1 as an example, the setting steps are as follows:

1. Enable CMP1 module function (write 0 to PWC_FCG3.CMP12);
2. Set the CVSL bit of the CMP_PMSR register to select the positive terminal voltage;
3. Set the RVSL bit of the CMP_PMSR register to select the negative terminal voltage;
4. Set the CENA bit in the CMP_MDR register to 1 to enable the comparator.
5. Wait for the comparator to stabilize tCMP (please refer to the product **Datasheet** for specific values);
6. Set the FCKS bit of the CMP_FIR register to set the sampling period of the digital filter;

7. Set the EDGS and CIEN bits of the CMP_FIR register to set the edge at which the interrupt occurs;
8. Set the CPOE and COPS bits of the CMP_OCR register to set the output of VCOUT;
9. The COEN bit in the CMP_OCR register is set to 1 to enable the comparator output.

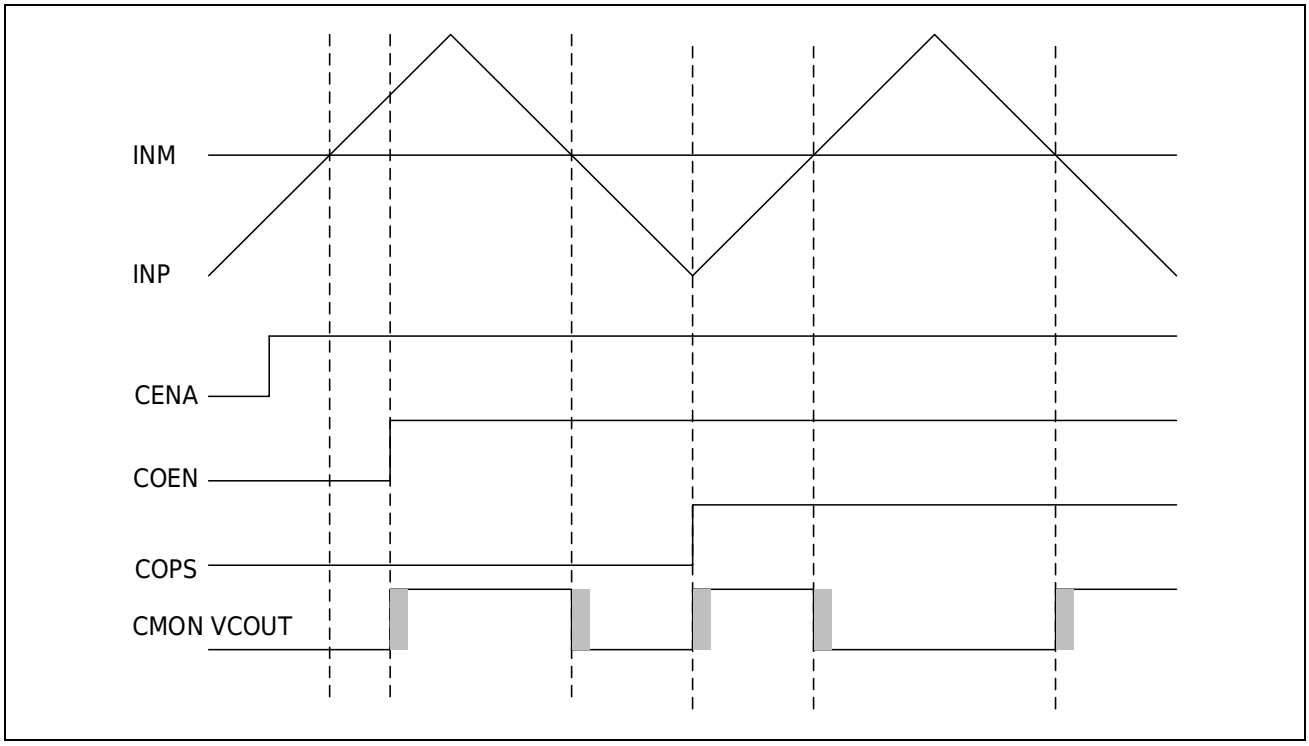


Figure 15-2 Common Comparison Schematic Diagram

As shown in Figure 15-2, in the positive phase output, if the positive terminal voltage is higher than the negative terminal voltage after CMP is enabled for output, the CMON bit of the CMP_MDR register is 1; if the positive terminal voltage is lower than the negative terminal voltage, the CMON bit is 0. The result of CMON in the inverting output is opposite to that in the non-inverting output. If a digital filter is used, the CMON edge will be at least 6 CMPCLK cycles later due to synchronization and filtered sampling.

15.3.4 Window Comparison

Window comparison can be achieved when CMP1 and CMP2 are running simultaneously. The setting steps are as follows:

1. Enable CMP1 and CMP2 module functions (PWC_FCG3.CMP12 write 0);
2. Set the CVSL bit in the CMP1_PMSR register to select the positive terminal voltage.

   Set the RVSL bit of the CMP1_PMSR register and select the negative terminal voltage as the lower limit of the window;
3. Set the CVSL bit in the CMP2_PMSR register to select the same positive terminal voltage as CMP1.

Set the RVSL bit of the CMP2_PMSR register and select the negative voltage as the upper limit of the window;

4. Set the CWDE bit of the CMP2_MDR register to 1 to start the window comparison;
5. Set the CENA bit in the CMP1_MDR and CMP2_MDR registers to 1 to enable the comparator.
6. Waiting time for stabilization tCMP; (For specific values, please refer to the product **Datasheet**);
7. Set the FCKS bit of the CMP2_FIR register to set the digital filter;
8. Set the EDGS and CIEN bits of the CMP2_FIR register to set the interrupt and edge detection conditions;
9. Set CPOE and COPS bits of CMP2 _ OCRregister and VCOUT output.
10. Set the COEN bit in the CMP1_OCR and CMP2_OCR registers to 1 to enable the comparator output.

During window comparison, CMP2 completes operations such as monitoring, filtering, interruption and output of the comparison results.

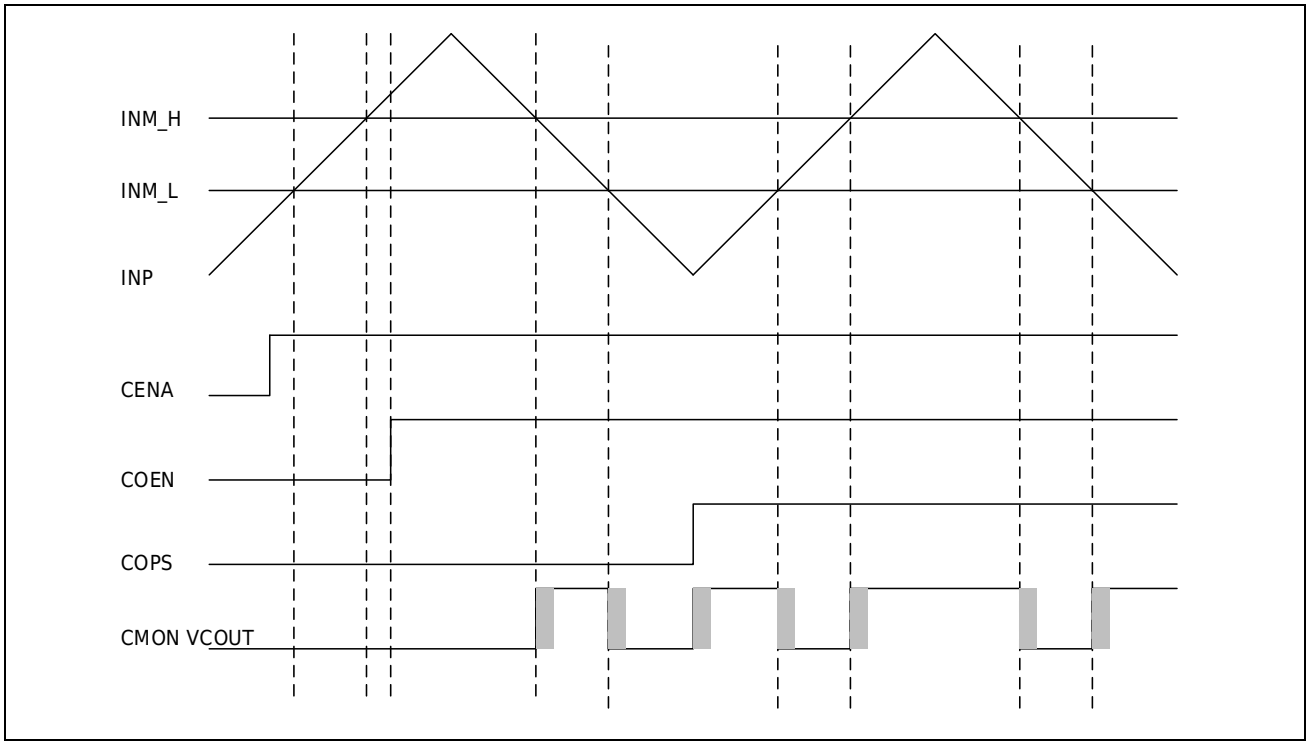


Figure 15-3 Diagram of Window Comparison

As shown in Figure 15-3, in the positive phase output, if the positive terminal voltage is within the window formed by the two negative terminal voltages after CMP is allowed to output, the CMON bit of the CMP2_MDR register is 1; if the positive terminal voltage is outside the window formed by the two negative terminal voltages, the CMON bit is 0. The result of CMON in the inverting output is

opposite to that in the non-inverting output. If a digital filter is used, the change edge of CMON will be relatively late by at least 6 CMPCLK cycles due to synchronization and filter sampling.

15.3.5 Input Scan

Input scanning allows multiple inputs to be compared with the same input in turn. When using, first select the positive or negative input to be scanned by setting the CSMD bit of the CMP_MDR register, and then set the SISL bit of the CMP_SCCR register to select the positive/negative input to be scanned. The SPRD bit in the CMP_SCCR register sets the scan period, which is the time interval between switching inputs during scanning. The SSTB bit sets the time to turn off the CMP output after switching the input. This is to wait for the CMP output to stabilize and avoid malfunction of the system.

Set the CSST bit in the CMP_MDR register to 1 and the scan starts. During scanning, the non-scanning ports remain unchanged, and the scanning ports switch cyclically according to the set scanning cycle. If a rising edge is detected in the comparison result, CSST is automatically cleared and the scan stops. At this time, the current input status of the positive and negative terminals can be determined by reading the CMP_SCMR register. When CSST is written to 1 again, the scanning action restarts.

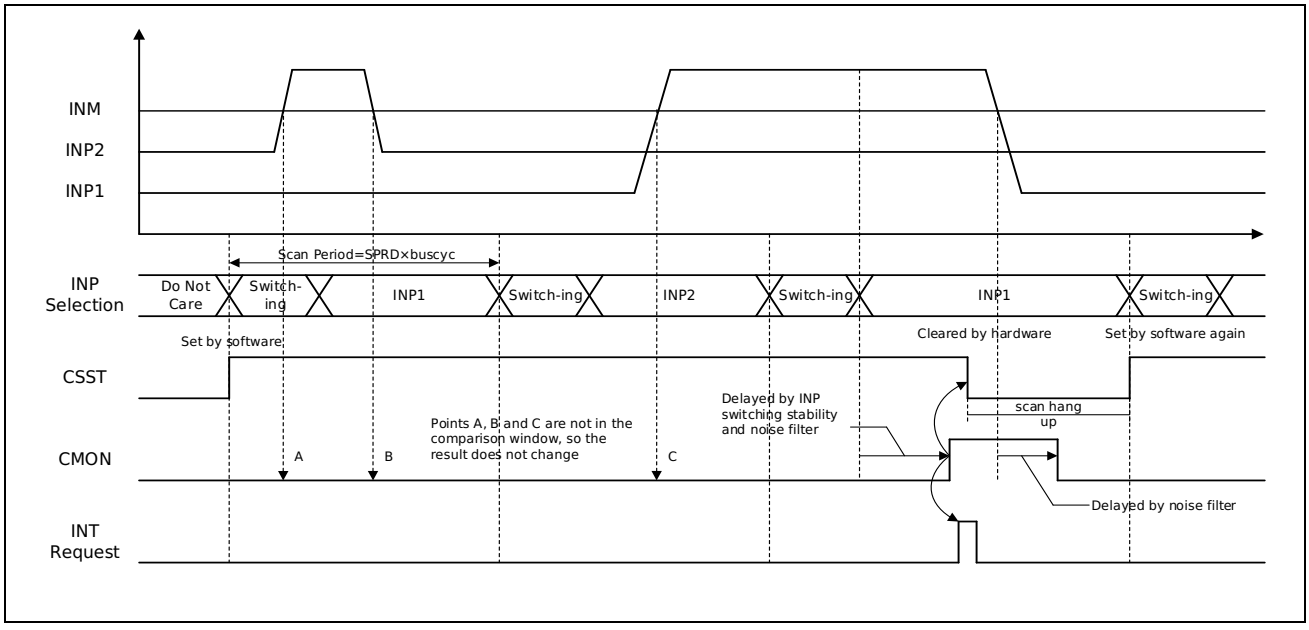


Figure 15-4 Input Scanning Schematic Diagram

Figure 15-4 is a schematic diagram of the action of scanning two positive inputs. In the first two scanning cycles, since the changes of INP1 and INP2 (the three change points A, B, and C in the figure) did not occur in their respective scanning cycles, the comparison results did not change. Next, when INP1 is scanned again, INP1 is higher than INM, causing the comparator to invert. At this time, the comparator output is turned off because it is waiting for the input switching to stabilize. Wait until the comparator output is stable, and then output the comparison result after noise

filtering. At the same time, the scanning action automatically stops and an interrupt occurs. Next, CMP will continue to work in the state of comparing INP1 with INM until the software restarts the scan.

Please refer to the following process for settings:

1. Set the CSMD of the CMP_MDR register to 0b10 or 0b11;
2. When CSMD is set to 0b10, set the RVSL bit of the CMP_PMSR register to select an INM input. When CSMD is set to 0b11, set the CVSL bit in the CMP_PMSR register to select an INP input;
3. Set the SISL bit of the CMP_SCCR register to select the scan target port;
4. Set the SSTB bit of the CMP_SCCR register to set the output stabilization time;
5. Set the SPRD bit of the CMP_SCCR register to set the scan period;
6. Set the EDGS bit of the CMP_FIR register to select the rising edge detection;
7. Set the CIEN bit of the CMP_FIR register to enable interrupts;
8. Set the CENA bit in the CMP_MDR register to enable the comparator;
9. Wait for the comparator to stabilize tCMP (please refer to the product **Datasheet** for specific values);
10. Set the COEN bit of the CMP_OCR register to enable the comparator output;
11. Set the CSST bit in the CMP_MDR register to start scanning.

Note:

In order to ensure correct operation, the set value must satisfy "scan cycle > output stabilization time + filter sampling cycle × 4 + CMPCLK cycle × 8". At the same time, it is necessary to set the detection of rising edge to automatically stop scanning or cause an interrupt. If you want to detect the falling edge, you can set the CMP inverted output to achieve. In addition, since the input scans of CMP1 and 2 need to be started separately, it is difficult to keep them in sync. It is recommended not to use input scanning with window comparisons.

15.3.6 Window Output

CMP can use the outputs of TMR4, TMR6, HRPWM, and TMRA as blank-window signals to control the window output of the comparison result CMPOA, which provides convenience for application scenarios such as timer PWM zero-crossing detection. Blank-Window has two modes: level and edge.

1) Level Mode

When the window output is enabled (the BWEN bit in the CMP_OCR register is 1) and the level mode is selected (the BWMD bit in the CMP_OCR register is 0), select one or more timer PWMs as Blank-Window through the CTWS bit in the CMP_BWSR1 register and select the effective level for each Blank-Window through the CTWP bit. In this way, the window output CMPOA will be fixed to the state set by the BWOL bit of the CMP_OCR register when Blank-Window valid.

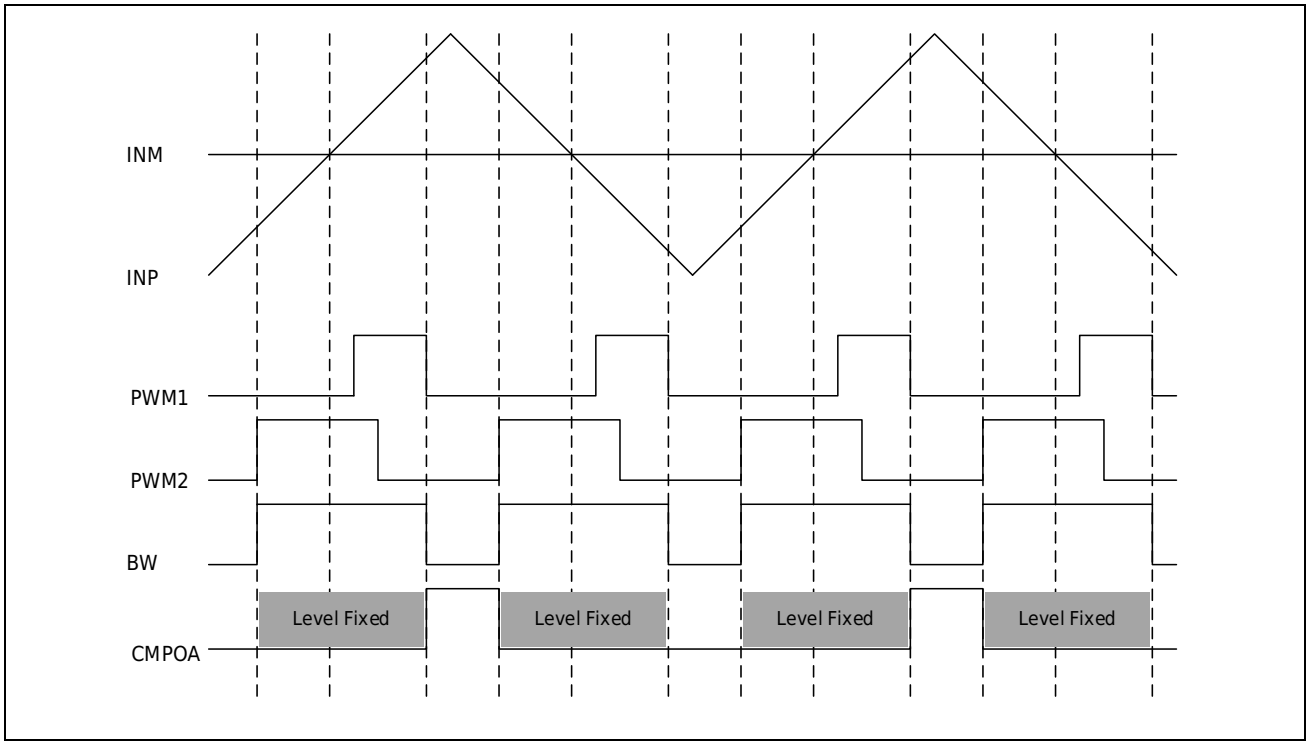


Figure 15-5 Window Output Schematic 1

As shown in Figure 15-5, two PWMs are selected as Blank-Window and both are valid at high level. When Blank-Window valid, the output state of CMPOA is selected as low level. Therefore, when the Blank-Window signal is at a high level, the CMPOA is fixed to a low level, and when the Blank-Window signal is at a low level, the CMPOA outputs the comparison result normally.

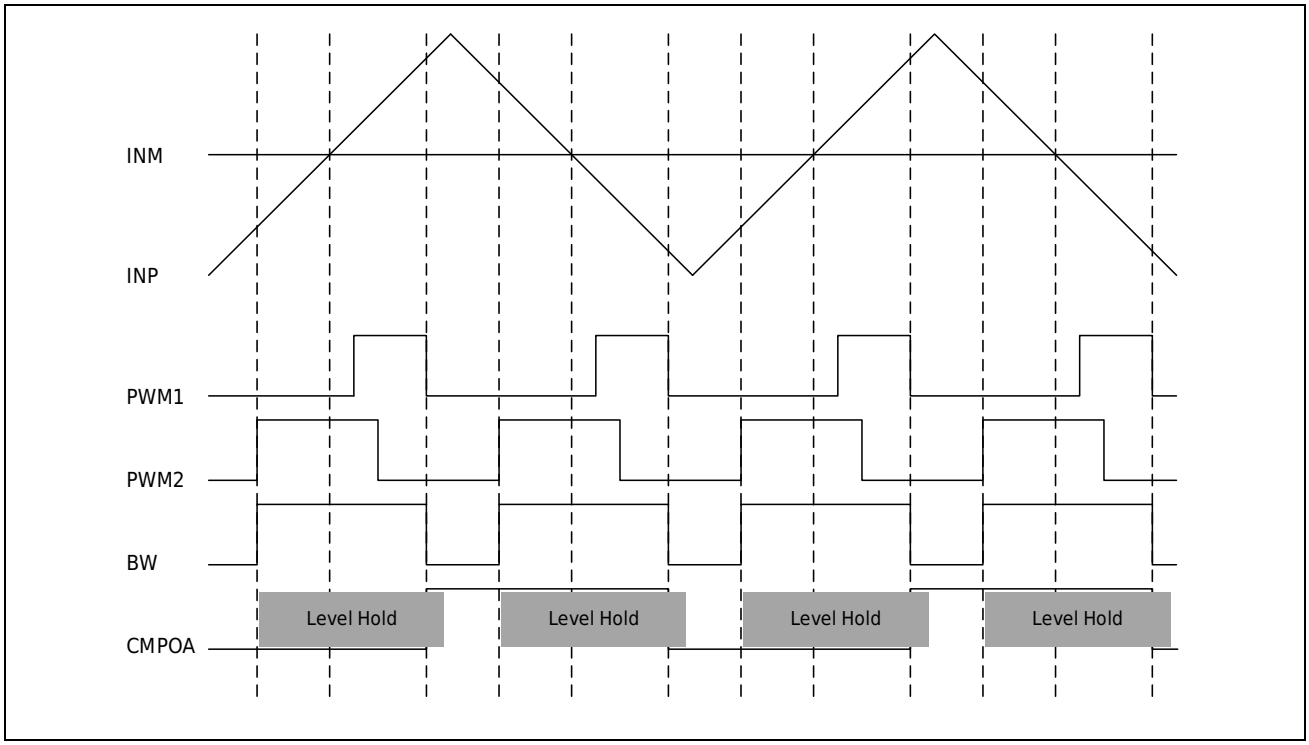


Figure 15-6 Window Output Schematic Diagram II

As shown in Figure 15-6, two PWMs are selected as Blank-Window and both are valid at high level. When Blank-Window valid, the output state of CMPOA is maintained. Therefore, when the Blank-Window signal is high, CMPOA maintains the previous state; when the Blank-Window signal is low, CMPOA outputs the comparison result normally, but the output will be 1 to 3 CMPCLK cycles later, which is caused by synchronization and sampling.

2) Edge Mode

When the window output is enabled (the BWEN bit in the CMP_MDR register is 1) and the edge mode is selected (the BWMD bit in the CMP_OCR register is 1), select one or more timer PWMs as Blank-Window through the CTWS bit in the CMP_BWSR1 register, and select the effective edge of the window signal through the TWEG bit in the CMP_BWSR2 register. In this way, the window output CMPOA will be fixed to the state set by the BWOL bit of the CMP_OCR register at the effective edge of the Blank-Window signal, and maintained for N CMPCLK cycles set by the MSKW bit of the CMP_BWSR2 register.

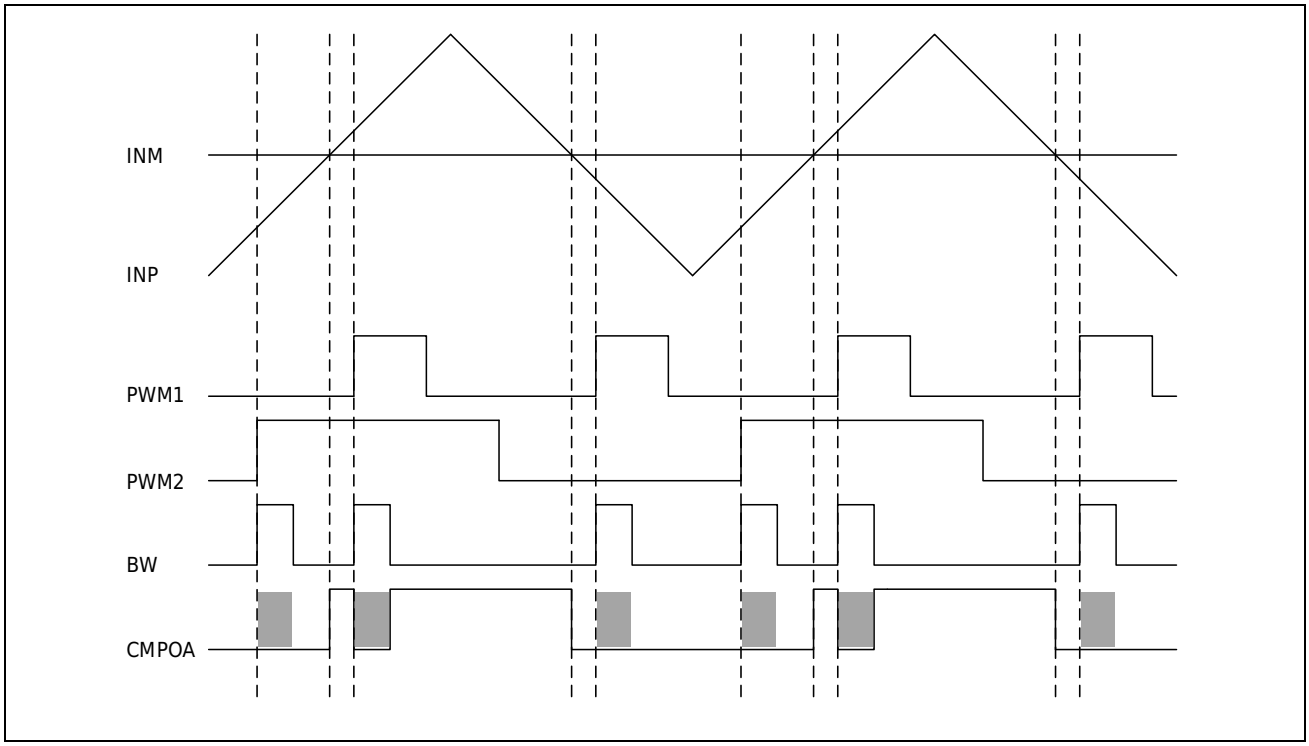


Figure 15-7 Window Output Schematic Diagram 3

As shown in Figure 15-7, two PWMs are selected as Blank-Window and the rising edge is selected as the valid edge. When Blank-Window valid, the output state of CMPOA is selected as low level. Therefore, a Blank-Window signal is generated at each rising edge of the PWM signal, and when the Blank-Window signal is at a high level, the CMPOA is fixed to a low level, and when the Blank-Window signal is at a low level, the CMPOA outputs the comparison result normally.

15.3.7 Hysteresis Function

To avoid false outputs caused by noise signals, each comparison channel has a configurable hysteresis function and the hysteresis voltage is selected by the HYST bit of CMPx_MDR. The hysteresis function disable when the HYST bits are set to 0b000.

15.3.8 Noise Filter

All comparison channels contain a noise filter each. The noise filter samples the comparator output CMPOA in each sampling clock cycle. If the same value is sampled three times in succession, the value is output to COUT in the next sampling clock cycle. Otherwise, COUT remains unchanged. The sampling clock can be selected by the FCKS bit in the CMP_FIR register. When the FCKS bit is set to 0b00, the noise filter is turned off.

15.3.9 External Pin output

The comparison result COUT output by the noise filter can be output to the external port VCOUT. The CPOE and COPS bits of the CMP_OCR register can be used to set whether VCOUT output is allowed and the output polarity (positive output or negative output). The port corresponding to the reset VCOUT is a universal port and input state, so it must also be set to VCOUT output via port register.

15.4Interrupt and Events

The results of all comparison channels can generate interrupt requests. When used, the CIEN bit of the CMP_FIR register must be set to 1 and the edge of the comparison result that can generate an interrupt must be set through the EDGS bit of the CMP_FIR register (a value other than 0b00). If noise filtering and inverting output are used, the edge detection is the signal after noise filtering and inverting output.

When INTC_WKEN.CMP*WKEN=1, CMPO can wake up the low-power Stop Mode. The system will be awakened and generate an interrupt when CMPO changes from low to high.

The conditions for generating 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, regardless of the CIEN bit.

15.5Register Description

Table 15-3 CMP Base Address

Name	Base Address	Description
CMP1	0x4003 8400	CMP1 base address
CMP2	0x4003 8500	CMP2 base address
CMP3	0x4003 B000	CMP3 base address

Table 15-4 CMP Register List

Register	Description	Offset Address	Bit Width	Reset Value
CMPx_MDR	Comparator Operating Mode Register	0x00	8	0x00
CMPx_FIR	Comparator Filter and Interrupt Registers	0x01	8	0x00
CMPx_OCR	Comparator Output Control Register	0x02	8	0x00
CMPx_OMR	Comparator Result Monitor Register	0x03	8	0x00
CMPx_PMSR	Comparator Positive and Negative Input Selection Register	0x04	32	0x0000 0000
CMPx_BWSR1	Comparator Output Blank Window Setting Register 1	0x10	32	0x0000 0000
CMPx_BWSR2	Comparator Output Blank Window Setting Register 2	0x14	16	0x0000
CMPx_SCCR	Comparator Input Scan Control Register	0x18	32	0x0000 0000
CMPx_SCMR	Comparator Input Scan Monitor Register	0x1C	32	0x0000 0000

In the table, x=1~3

All registers will maintain when software reset and watchdog reset occur.

15.5.1 Comparator Operating Mode Register (CMPx_MDR, x=1~3)

Reset Value: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
CSST	CSMD[1:0]		WDEN	HYST[2:0]			CENA

Bit	Marking	Bit Name	Function	Read and Write
b7	CSST	Input Scan Start	0: Stop input scanning 1: Start input scanning	RW
b6~b5	CSMD[1:0]	Input Scan Mode	0x: Turn off input scanning 10: Scan positive input INP 11: Scan the negative input INM	RW
b4	WDEN	Window comparison enable	WDEN is only present in CMP2. After setting to window comparison, the filtering, edge detection and output control of the comparison result are all completed by CMP2. 0: Universal comparison mode 1: Window comparison mode	RW
b3~b1	HYST[2:0]	Hysteresis voltage selection	The values listed below are typical values. Please refer to the product data sheet for the error range. 000: 0mv (no hysteresis) 001：10mv 010：20mv 011：30mv 100：40mv 101：50mv 110：60mv 111：70mv	RW
b0	CENA	Comparator Work Permit	0: Voltage comparator stopped 1: Voltage comparator operation Each time you change CENA from "0" to "1", please wait for tCMP (refer to the product data sheet) to ensure that CMP is stable before performing subsequent operations.	RW

15.5.2 Comparator Filter and Interrupt Register (CMPx_FIR, x=1~3)

Reset Value: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
CRF	CFF	EDGS[1:0]		CIEN	FCKS[2:0]		

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b7	CRF	Comparator Output Rising edge flag	0: The rising edge of the comparator output COUT is not detected. 1: The rising edge of the comparator output COUT is detected After CRF is set, it needs to be cleared by writing 1.	RW
b6	CFF	Comparator Output Falling edge flag	0: The falling edge of the comparator output COUT is not detected. 1: The falling edge of the comparator output COUT is detected After CFF is set, it needs to be cleared by writing 1.	RW
b5~b4	EDGS[1:0]	Comparator output edge Test selection	00: Do not detect the edges of the comparator output COUT 01: Detect the rising edge of the comparator output COUT 10: Detect the falling edge of the comparator output COUT 11: Detect the rising and falling edges of the comparator output COUT Changing EDGS [1: 0] may cause a interrupt or peripheral trigger event, so set the register if the interrupt disable or peripheral trigger is invalid. After the register is set, clear the corresponding interrupt flag.	RW
b3	CIEN	Comparator interrupt enable	0: Disable comparator interrupt 1: Allow the comparator to interrupt	RW
b2~b0	FCKS[2:0]	Comparator result filtering Sampling selection	000: No noise filter is used 001: Use noise filter, sample through PCLK0 010: Use noise filter, sample through PCLK0/2 011: Use noise filter, sample through PCLK0/4 100: Use noise filter, sample through PCLK0/8 101: Use noise filter, sample through PCLK0/16 110: Use noise filter, sample through PCLK0/32 111: Use a noise filter and sample through PCLK0/64 Please rewrite FCKS[2:0] when the comparator output is disabled (that is, the COEN bit is "0"). When FCKS [2: 0] switches from '0b00' to a different value, use the filtered output after 4 samples as a interrupt request or a peripheral trigger event. Changing the FCKS [2: 0] may cause a interrupt or peripheral trigger event, so set the register if the interrupt disable or peripheral trigger is invalid. After the register is set, clear the corresponding interrupt flag.	RW

15.5.3 Comparator Output Control Register (CMPx_OCR, x=1~3)

Reset Value: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
BWOL[1:0]		BWMD	BWEN	Res	CPOE	COPS	COEN

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b7~b6	BWOL[1:0]	Blank-Window Output Level	0 0: The comparator outputs low level during Blank-Window active (output is off) 1: The comparator outputs high level during Blank-Window active (output is off) 1 x: The comparator output remains during Blank-Window active (output off)	RW
b5	BWMD	Blank-Window Mode	0: Level mode 1: Edge mode	RW
b4	BWEN	Blank-Window Enable	0: Disable the Blank-window function 1: Enable the Blank-window function	RW
b3	Res	Reserved Bit	Read at “0”, write at “0”	RW
b2	CPOE	VCOUT output enables	0: Disable VCOUT/EMB/HRPWM output 1: Allow VCOUT/EMB/HRPWM output	RW
b1	COPS	Comparator output polarity selection	0: Comparator positive phase output 1: Comparator inverted output Override COPS when comparator output is disabled (i.e. COEN bit is "0"). Changing the COPS bit may cause a interrupt or peripheral trigger event, so set the register if the interrupt disable or peripheral trigger is invalid. After the register is set, clear the corresponding interrupt flag.	RW
b0	COEN	Comparator output permissible	0: Disable comparator output (comparator output fixed to low level) 1: Allow comparator output	RW

15.5.4 Comparator Result Monitoring Register (CMPx_OMR, x=1~3)

Reset Value: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
Res							CMON

Bit	Marking	Bit Name	Function	Read and Write
b7~b1	Res	Reserved Bit	Read at “0”, write at “0”	RW
b0	CMON	Comparison Result Monitoring	In normal comparison mode 0: The positive terminal voltage is lower than the negative terminal voltage 1: The positive terminal voltage is higher than the negative terminal voltage In window comparison mode (valid only for CMP2) 0: Positive terminal voltage is outside the window 1: Positive terminal voltage is within the window	R

15.5.5 Comparator Positive and Negative Input Selection Register (CMPx_PMSR, x=1~3)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24
Res							
b23	b22	b21	b20	b19	b18	b17	b16
Res						CVSL[1:0]	
b15	b14	b13	b12	b11	b10	b9	b8
Res							
b7	b6	b5	b4	b3	b2	b1	b0
Res			RVSL[4:0]				

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b18	Res	Reserved Bit	Read at “0”, write at “0”	RW
b17~b16	CVSL[1:0]	Positive input selection	Select the positive input voltage of the comparator. The set value invalid when the positive input is scanned. 00: No positive voltage input 01: Select INP1 as the positive terminal voltage (See Table 15-2 for selected input) 10: Select INP2 as the positive terminal voltage (See Table 15-2 for selected input) Other: prohibitions When the scan positive input valid (ie CMP_MDR.CSMD[1:0] = "0b10"), the register setting invalid. Override CVSL when comparator output is disabled (i.e. COEN bit is "0"). After rewriting CVSL each time, please wait for 300ns before enabling the comparator output (i.e., COEN bit is set to "1"). Changing the CVSL may cause a interrupt or peripheral trigger event, so set the register if the interrupt disable or peripheral trigger is invalid. After the register is set, clear the corresponding interrupt flag.	RW
b15~b5	Res	Reserved Bit	Read at “0”, write at “0”	RW
b4~b0	RVSL[4:0]	Negative input selection	Select the negative input voltage of the voltage comparator. The set value invalid when the negative input is scanned. 00000: No negative voltage input 00001: Select INM1 as the negative terminal voltage (Selected inputs are shown in Table 15-2) 00010: Select INM2 as the negative terminal voltage (Selected inputs are shown in Table 15-2) 00100: Select INM3 as the negative terminal voltage (Selected inputs are shown in Table 15-2) 01000: Select INM4 as the negative terminal voltage (Selected inputs are shown in Table 15-2) 10000: Select INM5 as the negative terminal voltage (Selected inputs are shown in Table 15-2) Other: prohibitions When the scan negative input valid (ie CMP_MDR.CSMD[1:0] = "0b11"), the register setting invalid. Override RVSL when comparator output is disabled (i.e. COEN bit is "0"). After rewriting RVSL each time, please wait for 300ns before enabling the comparator output (i.e., COEN bit is set to "1"). Changing the RVSL may cause a interrupt or peripheral trigger event, so set the register if the interrupt disable or peripheral trigger is invalid. After the register is set, clear the corresponding interrupt flag.	RW

15.5.6 Comparator Output Blank Window Setting Register 1 (CMPx_BWSR1, x=1~3)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24
CTWP15	CTWP14	CTWP13	CTWP12	CTWP11	CTWP10	CTWP9	CTWP8
b23	b22	b21	b20	b19	b18	b17	b16
CTWP7	CTWP6	CTWP5	CTWP4	CTWP3	CTWP2	CTWP1	CTWP0
b15	b14	b13	b12	b11	b10	b9	b8
CTWS15	CTWS14	CTWS14	CTWS12	CTWS11	CTWS10	CTWS9	CTWS8
b7	b6	b5	b4	b3	b2	b1	b0
CTWS7	CTWS6	CTWS5	CTWS4	CTWS3	CTWS2	CTWS1	CTWS0

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b16	CTWP15~0	Window signal polarity	When Blank-Window is in level mode (CMP_OCR.BWMD=0), it sets the effective level of the window signal output by the MASK comparator. (This bit setting valid only when the corresponding CTWS is 1) 0: When the window signal is high, the comparator output is turned off. 1: When the window signal is low, the comparator output is turned off.	RW
b15~b0	CTWS15~0	Window signal enable	When Blank-Window valid (CMP_OCR.BWEN=1), select the window signal for the MASK comparator output. (The timer PWM signal corresponding to each bit is shown in Table 15-5) 0: Do not select the timer PWM output as the window signal 1: Select timer PWM output as window signal	RW

Table 15-5 Timer Window PWM List

CTWS bit	CMP1	CMP2	CMP3
15	TMR4_OWL	TMR4_OWL	TMR4_OWL
14	TMR4_OWH	TMR4_OWH	TMR4_OWH
13	TMR4_OVL	TMR4_OVL	TMR4_OVL
12	TMR4_OVH	TMR4_OVH	TMR4_OVH
11	TMR4_OUL	TMR4_OUL	TMR4_OUL
10	TMR4_OUH	TMR4_OUH	TMR4_OUH
9	TMR6_1_PWMA	TMR6_2_PWMA	TMR6_1_PWMB
8	HRPWM_PWMA_5_PRE	HRPWM_PWMA_6_PRE	HRPWM_PWMB_5_PRE
7	HRPWM_PWMA_3_PRE	HRPWM_PWMA_4_PRE	HRPWM_PWMB_3_PRE
6	HRPWM_PWMA_1_PRE	HRPWM_PWMA_2_PRE	HRPWM_PWMB_1_PRE
5	TMRA_2_PWM3	TMRA_4_PWM3	TMRA_3_PWM3
4	TMRA_2_PWM2	TMRA_4_PWM2	TMRA_3_PWM2
3	TMRA_2_PWM1	TMRA_4_PWM1	TMRA_3_PWM1
2	TMRA_1_PWM3	TMRA_3_PWM3	TMRA_1_PWM3
1	TMRA_1_PWM2	TMRA_3_PWM2	TMRA_1_PWM2
0	TMRA_1_PWM1	TMRA_3_PWM1	TMRA_1_PWM1

15.5.7 Comparator Output Blank Window Setting Register 2 (CMPx_BWSR2, x=1~3)

Reset value: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8
Res						TWEG[1:0]	
b7	b6	b5	b4	b3	b2	b1	b0
MSKW[7:0]							

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b15~b10	Res	Reserved Bit	Read at “0”, write at “0”	RW
b9~b8	TWEG[1:0]	Window Signal Edge	When Blank-Window is in edge mode (CMP_OCR.BWMD=1), it sets the effective edge of the window signal output by the MASK comparator. 00: No valid edge 01: The comparator output is turned off at the rising edge of the window signal 10: Comparator output is turned off at the falling edge of the window signal 11: The comparator output is turned off at the changing edge of the window signal	
b7~b0	MSKW[7:0]	Blank-Window effective width	When Blank-Window is in edge mode (CMP_OCR.BWMD=1), it sets the width of the MASK comparator output. It can be set to any value between 0 and 255. When the window signal detects an edge set by the TWEG bit, the comparator output will be turned off and last for the number of CMPCLK cycles set by the MSKW bit.	RW

15.5.8 Comparator Input Scan Control Register (CMPx_SCCR, x=1~3)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24
Res			SSTB[5:0]				
b23	b22	b21	b20	b19	b18	b17	b16
SPRD[7:0]							
b15	b14	b13	b12	b11	b10	b9	b8
Res							
b7	b6	b5	b4	b3	b2	b1	b0
Res			SISL[4:0]				

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b30	Res	Reserved Bit	Read at “0”, write at “0”	RW
b29~b24	SSTB[5:0]	Output stabilization time	The output stabilization time of CMP after the input is switched during input scanning. CMP output stabilization time = CMPCLK period × SSTB setting value **Note:** 1. Please set the INP switching stabilization time to be longer than the "input channel switching stabilization time" stated in the electrical features according to the CMPCLK cycle. 2. If you need to rewrite SSTB, please stop scanning first.	RW
b23~b16	SPRD[7:0]	Input switching cycle	The time interval between switching inputs during input scanning is called the scanning period. Scan cycle = CMPCLK cycle × SPRD setting value **Note:** 1. In order to ensure correct operation, please confirm when setting SPRD Scan cycle> output stabilization time + filter sampling cycle × 4 + CMPCLK cycle × 8 2. If you need to rewrite SPRD, please stop scanning first.	RW
b15~b5	Res	Reserved Bit	Read at “0”, write at “0”	RW
b4~b0	SISL[4:0]	Input Selection	When the positive input is scanned, b1~b0 correspond to the positive input INP2~INP1 of CMP respectively; When the negative input is scanned, b4~b0 correspond to the positive input INM5~INM1 of CMP respectively; 0: The input is not scanned. 1: Input as scan object **Note:** If you need to rewrite SISL, please stop scanning first.	RW

15.5.9 Comparator Input Scan Monitor Register (CMPx_SCMR, x=1~3)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24
Res							
b23	b22	b21	b20	b19	b18	b17	b16
Res						CVST[1:0]	
b15	b14	b13	b12	b11	b10	b9	b8
Res							
b7	b6	b5	b4	b3	b2	b1	b0
Res			RVST[4:0]				

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b18	Res	Reserved Bit	Read at “0”, write at “0”	RW
b17~b16	CVST[1:0]	CMP positive input status	This register is a read-only register and is used to monitor the input status of CMP during input scanning. b17~b16 correspond to the positive input INP2~INP1 of CMP respectively. 0: No longer comparing INPn (n=1~2). 1: INPn (n=1~2) is being compared.	R
b15~b5	Res	Reserved Bit	Read at “0”, write at “0”	RW
b4~b0	RVST[4:0]	CMP negative input status	This register is a read-only register and is used to monitor the input status of CMP during input scanning. b4~b0 correspond to the negative input INM5~INM1 of CMP respectively. 0: No longer comparing INMn (n=1~5). 1: INMn (n=1~5) is being compared.	R

15.6Precautions

15.6.1 Module Stop Function

CMP has module stop function, by setting module stop register can turn off the digital part of the module. CMP1 and 2 share a module stop control bit PWC_FCG3.bit8, and the CMP3 module stop control bit is PWC_FCG3.bit9. CMP1~3 are initially in the stop state, and their respective registers can only be accessed when the module is set to work. For related register settings, please refer to the [Action Mode and Low Power Mode] section.

15.6.2 Power Consumption when the Module is Stopped

When the CMP enters the module stop state from the working state, the comparator will continue to work and the Power Consumption is equivalent to the working state. To further reduce Power Consumption, clear the CENA bit in the CMP_MDR register to 0.

15.6.3 Power Consumption When Stopping Low Power Mode

When the chip enters the stop low-power mode, if the CMP is in operation, it will continue to operate after entering the stop low-power mode, which is equivalent to the level before entering the stop low-power mode. To further reduce Power Consumption, clear the CENA bit in the CMP_MDR register to 0 before entering stop low-power mode.

16 Analog-to-Digital Converter (ADC)

16.1 Overview

12-bit ADC is an analog-to-digital converter with successive approximation. This MCU is equipped with 3 ADC units. Unit 1 supports 16 channels, unit 2 supports 12 channels, and unit 3 supports 10 channels. It can convert analog signals from external pins and inside the chip. The analog input channels can be arbitrarily combined into a sequence, and a sequence can be converted in a single scan or in a continuous scan. It supports multiple consecutive conversions on any specified channel and averages the conversion results. The ADC module is also equipped with an analog watchdog function to monitor the conversion result of any specified channel to detect whether it exceeds the range set by the user.

16.2 Main Features

- High performance
  - Configurable 12-bit, 10-bit and 8-bit resolution
  - The frequency ratio of the ADC digital interface clock PCLK4 and the conversion clock PCLK2 (also called ADCLK) can be set to 1:1, 2:1, 4:1, 8:1, 1:2, 1:4
    PCLK2 can be a PLL clock asynchronous with the system clock HCLK. In this case, the frequency ratio PCLK4:PCLK2=1:1
    PCLK2 frequency supports up to 60MHz
  - Sampling rate: 2.5MSPS (PCLK2 = 60MHz, 12 bits, sampling 11 cycles, conversion 13 cycles)
  - The sampling time of each channel is independently programmed
  - Channel-independent data register
  - Data registers can be configured with left/right alignment
  - Consecutive multiple conversion average function
  - Oversampling function
  - Simulation watchdog, monitoring conversion results
  - The ADC module can be set to stop when not in use
- Analog input channel
  - Total 22 external analog inputs (3 internal analog channels)
  - 1 internal analog input: internal reference voltage
- Conversion start condition
  - Software Setup Conversion Started
  - Start of Peripheral Device Synchronization Trigger Conversion
  - External Pin Triggering Start
- Conversion mode

- 2 scan sequences A and B, optionally specifying a single or multiple channels
- Sequence A single scan
- Sequence A continuous scanning
- Sequence A multiple data buffer mode
- Double sequence scanning, sequence A and B independently select trigger source, sequence B has higher priority than A
- Co-working mode (For devices with two or three ADCs)

■ Interrupt and Event Signal Output

- Sequence A Scan End Interrupt and Event ADC _ EOCA
- Sequence B Scan End Interrupt and Event ADC _ EOCB
- Analogwatchdog 0 compared interrupt and event ADC _ CMP0
- Simulation watchdog 1 comparing interrupt and event ADC _ CMP1
- All four of these event outputs can start the DMA

16.3Functional Description

16.3.1 ADC System Block Diagram

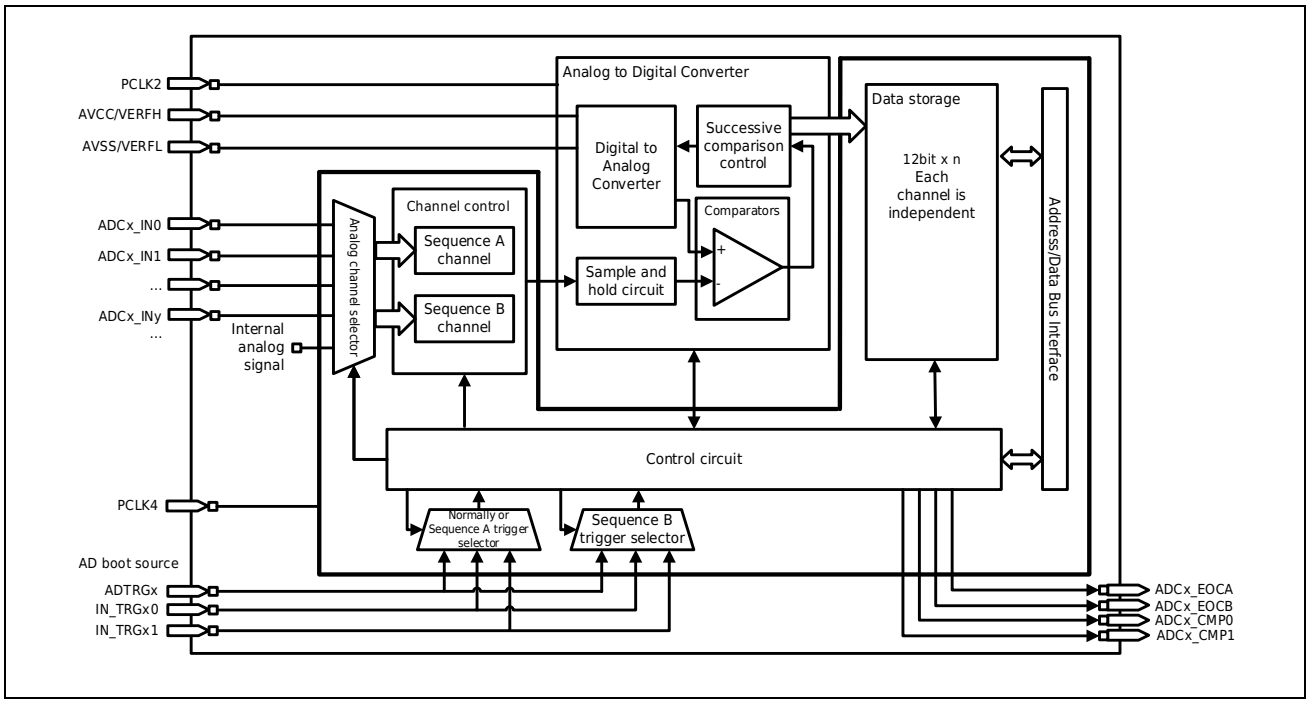


Figure 16-1 ADC Block Diagram

This chip is equipped with 3 ADC module units, and the configuration of each unit is different. For details, refer to the following table:

Table 16-1 Specifications of each ADC unit

Item	Unit 1 (ADC1)	Unit 2 (ADC2)	Unit 3 (ADC3)
Power supply/reference voltage	AVCC/VREFH		

Item		Unit 1 (ADC1)	Unit 2 (ADC2)	Unit 3 (ADC3)
		AVSS/VREFL		
Analog channel*1	CH0	ADC1_IN0	-	ADC3_IN0
	CH1	ADC1_IN1	-	ADC3_IN1
	CH2	ADC1_IN2	-	ADC3_IN2
	CH3	ADC1_IN3	-	ADC3_IN3
	CH4	ADC12_IN4	ADC12_IN4	-
	CH5	ADC12_IN5	ADC12_IN5	-
	CH6	ADC123_IN6	ADC123_IN6	ADC123_IN6
	CH7	ADC123_IN7	ADC123_IN7	ADC123_IN7
	CH8	ADC123_IN8	ADC123_IN8	ADC123_IN8
	CH9	ADC123_IN9	ADC123_IN9	ADC123_IN9
	CH10	ADC12_IN10	ADC12_IN10	ADC3_IN10
	CH11	ADC12_IN11	ADC12_IN11	ADC3_IN11/ Internal analog channel
	CH12	ADC12_IN12	ADC12_IN12	-
	CH13	ADC12_IN13	ADC12_IN13	-
	CH14	ADC12_IN14	ADC12_IN14	-
	CH15	ADC12_IN15/ Internal analog channel	ADC12_IN15/ Internal analog channel	-
Hardware trigger source	External pin	ADTRG1	ADTRG2	ADTRG3
	On-chip peripherals	IN_TRG10	IN_TRG20	IN_TRG30
		IN_TRG11	IN_TRG21	IN_TRG31

Note:

The virtual channels CH0~CH15 in ADC and the physical channels ADCx_INy (y=0~15) (actual analog input source) can be freely mapped by setting registers. This table shows the default mapping relationship after reset.

16.3.2 ADC Clock

The ADC module requires two clocks: digital interface clock PCLK4 and analog circuit clock PCLK2. PCLK4 and PCLK2 are in synchronization relationship, and the frequency ratio can be set to 1:1, 2:1, 4:1, 8:1, 1:2, 1:4.

PCLK2 can select a PLL clock source that is asynchronous with the system clock HCLK. In this case, PCLK4 is the same as PCLK2, and they are synchronous and have the same frequency.

Note:

Please set the frequency of PCLK2 within 1MHz~60MHz.

16.3.3 Channel Selection

The ADC module supports channel mapping, that is, the mapping between the virtual channels in the module and the actual physical channels. A virtual channel refers to a hypothetical channel in the ADC module. For example, if register ADC_CHSELRA is set to 0x1, it means that sequence A selects to convert CH0. This CH0 is the virtual channel, and register ADC_DR0 is the conversion result register of virtual channel CH0. The physical channel refers to the actual analog channel, that is, the analog input ADCx_INy of the external pin. The mapping between virtual channels and physical channels can be configured through the register ADC_CHMUXR. For details, refer to the Register Description. Unless otherwise specified in this chapter, channel n or CHn refers to a virtual channel.

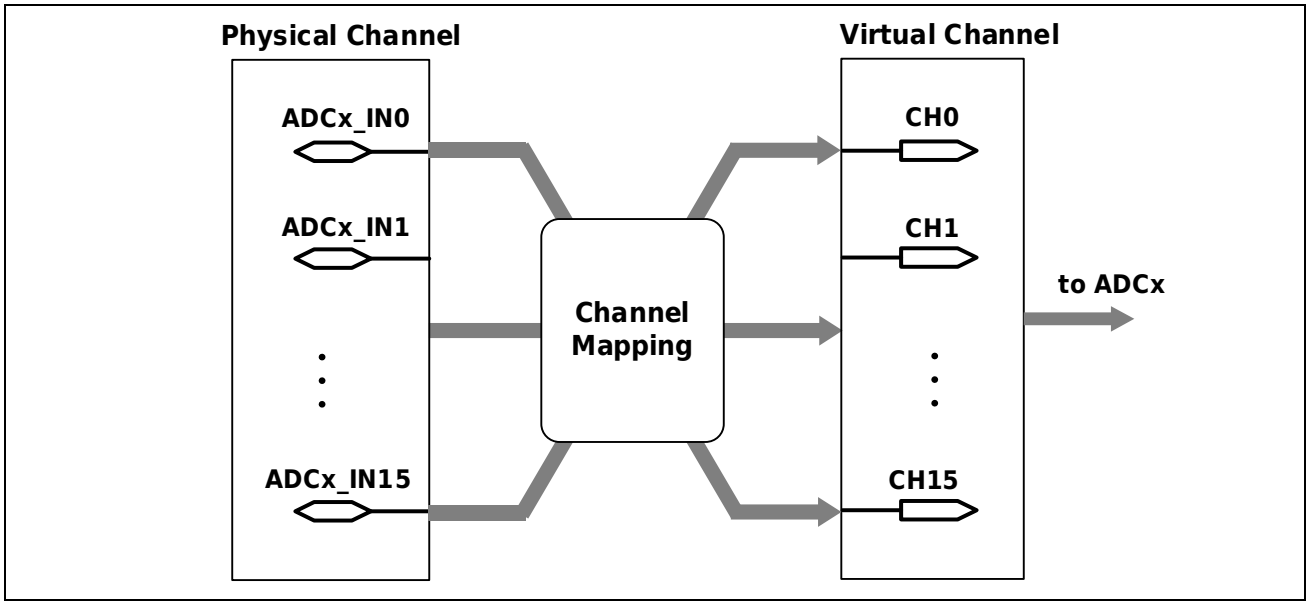


Figure 16-2 Channel Mapping Diagram

The ADC module has multiple channels and can be configured as two sequences: sequence A, sequence B for conversion. Sequences A and B are equipped with independent channel selection registers ADC_CHSELRA, ADC_CHSELRB. Register each represents a channel, such as bit0 bit write 1 to convert CH0 and write 0 to not convert CH0. Two sequences can be converted independently by selecting any one or more channels. For example: ADC_CHSELRA is set to 0x0055, ADCHSELRB is set to 0x0002, when the trigger condition of sequence A occurs, the four channels CH0, CH2, CH4 and CH6 will be converted in sequence. When the trigger condition for sequence B occurs, CH1 is converted.

Among them, the internal analog output shares one CH with the maximum external input channel ADCx_INy of each unit (y=15 when x=1, y=15 when x=2, and y=11 when x=3). Setting ADC_CHMUX changes the internal analog output and the mapping relationship between ADCx_INy and CHn at the same time. When you need to convert the internal analog channel or ADCx_INy, you need to first set the extended channel register ADC_EXCHSELR, and then write 1 to the

corresponding virtual channel bit in ADC_CHSELRA/ADC_CHSELB. When ADC_EXCHSELR is set to 0x0, it means converting external analog input, and when ADC_EXCHSELR is set to 0x1, it means converting internal analog channels.

After reset, the internal analog channels are in a closed state. Before using these channels, please refer to the [Power Control (PWC)] section to set the corresponding registers to allow internal analog voltage output.

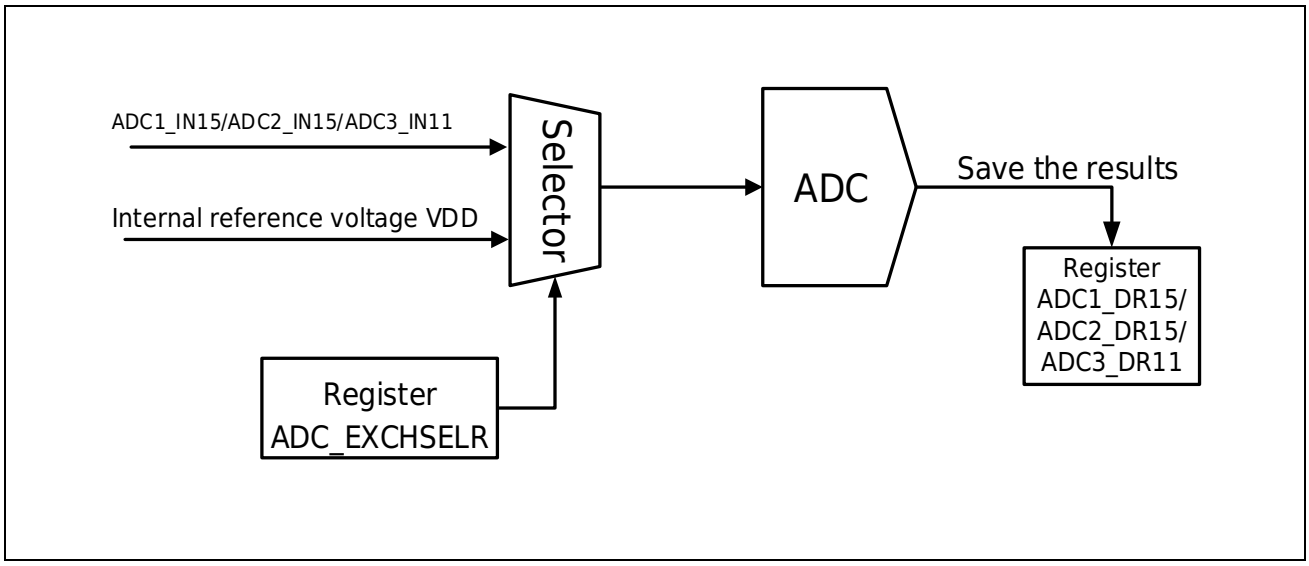


Figure 16-3 Internal Analog Channel Selection

In addition, most physical channels can be input to multiple ADC module units, such as ADC123_IN6 can be input to ADC_1, 2, 3 units, and ADC12_IN4 can be input to ADC_1, 2 units. Combined with the multi-ADC collaborative working mode and channel mapping function, high sampling rate conversion can be easily achieved for the specified channel.

Note:

Do not select the same channel in sequence A and B. For channels that do not exist, please do not set the corresponding registers and keep the status after reset.

16.3.4 Trigger Source Selection

Sequence A, sequence B independently selects the trigger source. Selectable trigger sources include external port ADTRGx, internal events IN_TRGx0, IN_TRGx1. Among them, the falling edge input of port ADTRGx is valid. IN_TRGx0, IN_TRGx1 are set by registers ADC_TRGSEL0,1, which can select rich event sources inside the chip. In addition, writing register ADC_STR can generate a sequence A software trigger signal. The sequence A software trigger can only be used when the ADC is in standby state. Software triggering has nothing to do with the setting of the trigger source selection register ADC_TRGSR.

16.3.5 Sequence A single scan mode

The ADC control register ADC_CR0.MS[2:0] is set to 0b000 to select the sequence A single scan mode.

In this mode, when the sequence A start condition selected by the register ADC_TRGSR occurs, or the ADC_STR.STRT bit writes 1 software trigger, the ADC starts, and all channels selected in the sequence A channel selection register ADC_CHSELRA are sequentially sampled and converted, and the conversion results are stored in the corresponding data register 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.EOCAF is set to 1 and the sequence A conversion end event ADC _ EOCA is generated, which can be used to start the DMA. If the ADC _ ICR.EOCAIEN is 1, the interrupt request is generated in the state of interrupt enable.

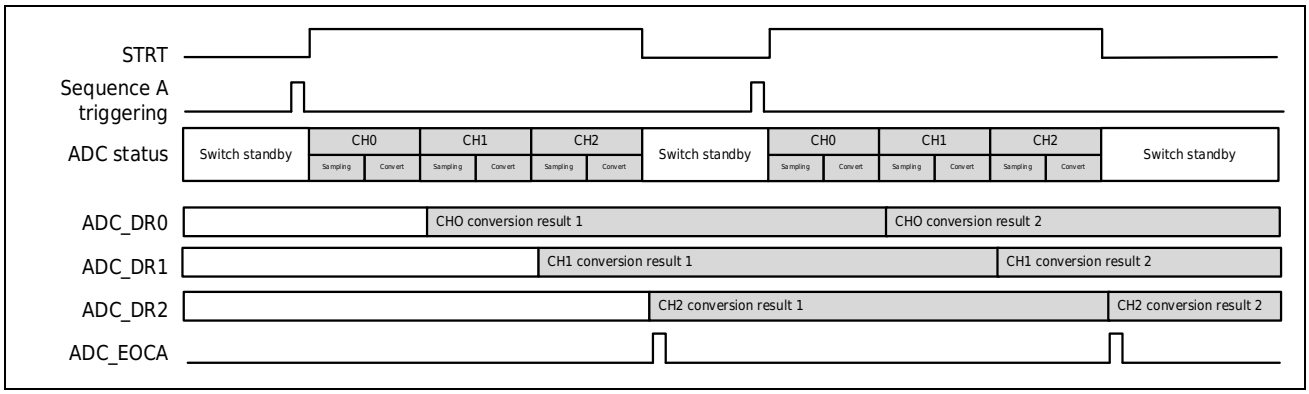


Figure 16-4 Sequence A Single Sweep Mode

Sequence A single scan mode software process:

1. Confirm that ADC_STR.STRT is 0, ADC is in conversion standby state;
2. ADC control register ADC_CR0.MS[2:0] is set to 0b000 to select sequence A single scan mode;
3. Set the sequence A channel select register ADC_CHSELRA;
4. Set the sampling time registerADC _ SSTR;
5. ADC _ STR.STRT Write 1 software trigger sequence A or set registerADC _ TRGSR select sequence A trigger condition;
6. Query sequence A conversion end flag bit EOCAF;
7. Read each channel data registerADC _ DR;
8. Write 0 clears the EOCAF flag bit to prepare for the next conversion.

The CPU query modes in steps 6 to 8 above can also be replaced by interrupt mode, which can be used to process the converted data. Or start DMA read data using the ADC _ EOCA event.

16.3.6 Sequence A continuous scanning mode

The ADC control register ADC_CR0.MS[2:0] is set to 0b001 to select the sequence A continuous scan mode.

Sequence A continuous scan mode is similar to sequence A single scan mode. The difference lies in that continuous scan mode does not enter conversion standby state after channel conversion, but starts to convert sequence A. 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.

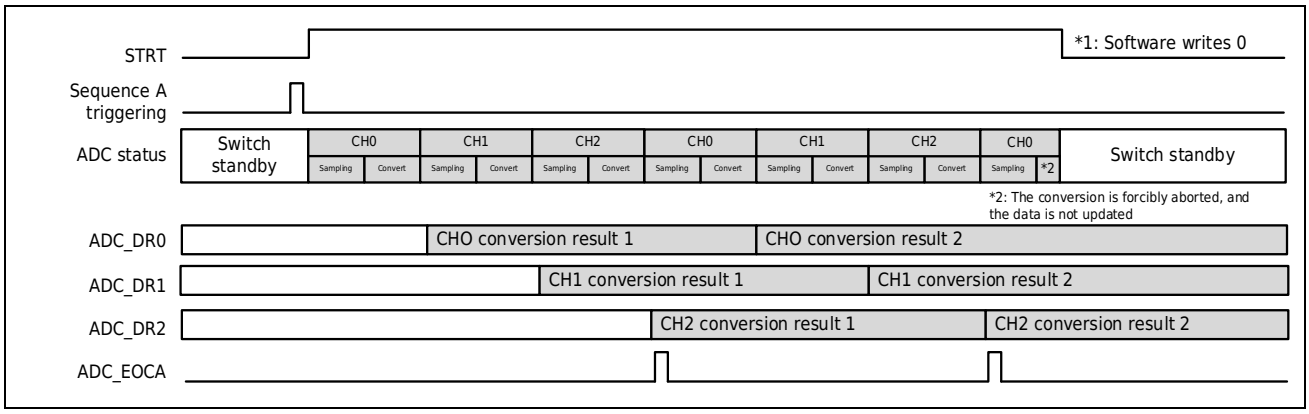


Figure 16-5 Continuous Scanning

Sequence A Continuous Scan Mode software flow:

1. Confirm that ADC_STR.STRT is 0, ADC is in conversion standby state;
2. ADC control register ADC_CR0.MS[2:0] is set to 0b001 to select sequence A continuous scan mode;
3. Set the sequence A channel select register ADC_CHSELRA;
4. Set the sampling time registerADC _ SSTR;
5. ADC _ STR.STRT Write 1 software trigger sequence A or set registerADC _ TRGSR select sequence A trigger condition;
6. Query sequence A conversion end flag bit EOCAF;
7. Read each channel data registerADC _ DR;
8. Write 0 clears the EOCAF flag bit to prepare for the next conversion.
9. Write 0 to the STRT bit when you do not need to continue the conversion, and read STRT to confirm that 0 to determine if the ADC is in conversion standby.

The above-mentioned query methods in steps 6 to 8 can also be replaced with interrupt methods, which can be used to process the converted data. Or start DMA read data using the ADC _ EOCA event.

Note:

Since it is a continuous conversion, the interval between scans is shorter, especially when only one channel conversion is selected. It is recommended to use the ADC _ EOCA event to start DMA read data in order to avoid data loss due to intimely processing in query mode.

16.3.7 Sequence A Data Buffer Mode

The ADC control register ADC_CR0.MS[2:0] is set to 0b100 to select the sequence A data buffer mode.

The operation of Sequence A data buffer mode is similar to that of Sequence A single scan mode. The difference is that after the conversion of one channel is completed, the data buffer mode does not directly start the conversion of the next channel, but enters the conversion standby state until the start condition occurs again.

When all channel conversion ends, the sequence A conversion end flag bit ADC _ ISR.EOCAF is set to 1 and the sequence A conversion end event ADC _ EOCA is generated, which can be used to start the DMA. If the ADC _ ICR.EOCAIEN is 1, the interrupt request is generated in the state of interrupt enable.

In this mode, you can use the register ADC_CHSELRA to select multiple virtual channels, and then use the channel mapping function to map these virtual channels to the same physical channel. In this way, when all channels are converted, you can read the conversion data from the ADC_DR register corresponding to each virtual channel in turn.

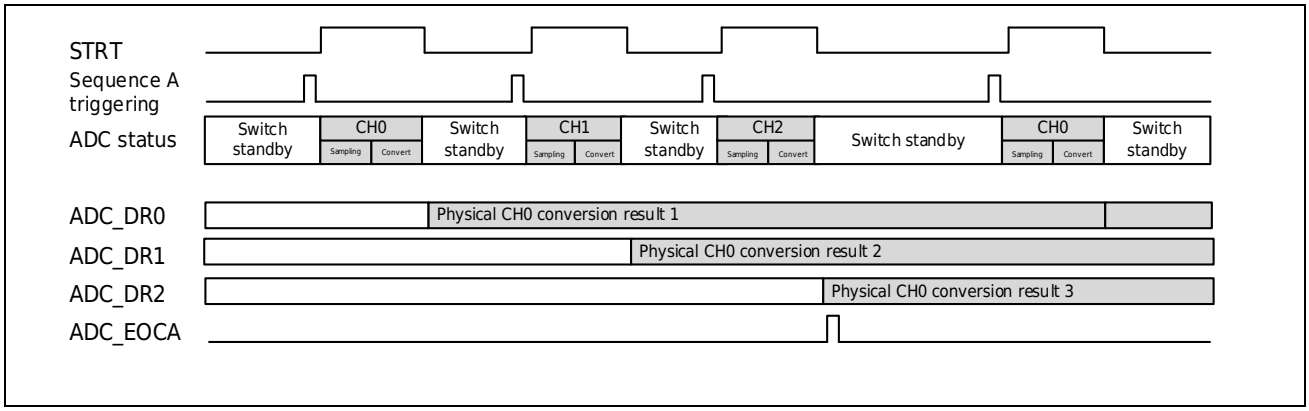


Figure 16-6 Data Buffering Mode

Software flow of sequence A data buffer mode:

1. Confirm that ADC_STR.STRT is 0, ADC is in conversion standby state;
2. ADC control register ADC_CR0.MS[2:0] is set to 0b100 to select sequence A data buffer mode;

3. Set the sequence A channel select register ADC_CHSELRA;
4. Set the Channel Mapping Control Register ADC_CHMUXR;
5. Set the sampling time registerADC _ SSTR;
6. ADC _ STR.STRT Write 1 software trigger sequence A or set registerADC _ TRGSR select sequence A trigger condition;
7. Query the end of single conversion ADC_STR.STRT=0;
8. Repeat steps 6 to 7 until all channels are converted;
9. Query sequence A conversion end flag bit EOCAF;
10. Read each channel data registerADC _ DR;
11. Write 0 clears the EOCAF flag bit to prepare for the next conversion.

The above-mentioned query methods in steps 9 to 11 can also be replaced with interrupt methods, which can be used to process the converted data. Or start DMA read data using the ADC _ EOCA event.

16.3.8 Double Sequence Scanning Mode

The ADC control register ADC_CR0.MS[2:0] is set to 0b010 or 0b011 to select the dual-sequence scan mode, that is, both sequence A and sequence B can start scanning by their respective selected trigger conditions.

When MS [2: 0] = 0b010, sequences A and B are equivalent to two independent single-scan sequences. MS [2: 0] = 0b011 sequence A is continuous scan mode, B is single scan mode. MS[2:0]=110b Sequence A is data buffer mode, B is single scan mode.

In sequence A, the trigger source is selected by ADC_TRGSR.TRGSELA[2:0], and the converted channel is selected by ADC_CHSELRA. In sequence B, the trigger source is selected by ADC_TRGSR.TRGSELB[2:0], and the converted channel is selected by ADC_CHSELRB.

The sequence A conversion end flag bit ADC _ ISR.EOCAF is set to 1 and the sequence A conversion end event ADC _ EOCA is generated. If the sequence A conversion end flag bit ADC _ ISR.EOCAIEN is set to 1, the sequence A conversion end interrupt request is generated. 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_ISCR.EOCBIEN is 1, the interrupt is enabled and the sequence B conversion end interrupt is generated at the same time. ask.

In double 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 Sequences A and B of Various Competitions

ADC Conversion	Trigger Signal Occurrence	Mode of Treatment	
		ADC_CR1.RSCHSEL=0	ADC_CR1.RSCHSEL=1
Sequence A conversion in process	Sequence A triggering	Invalid trigger signal	
	Sequence B triggering	1) The sequence A conversion is interruptd and the sequence B conversion is started. 2) After the conversion of sequence B is complete, sequence A continues to be converted from the channel interruptd.	1) The sequence A conversion is interruptd and the sequence B conversion is started. 2) After the conversion of sequence B is complete, sequence A is reconverted from the first channel.
Sequence B conversion in process	Sequence A triggering	Sequence B All channel conversion is complete and sequence A conversion is started.	
	Sequence B triggering	Invalid 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.	

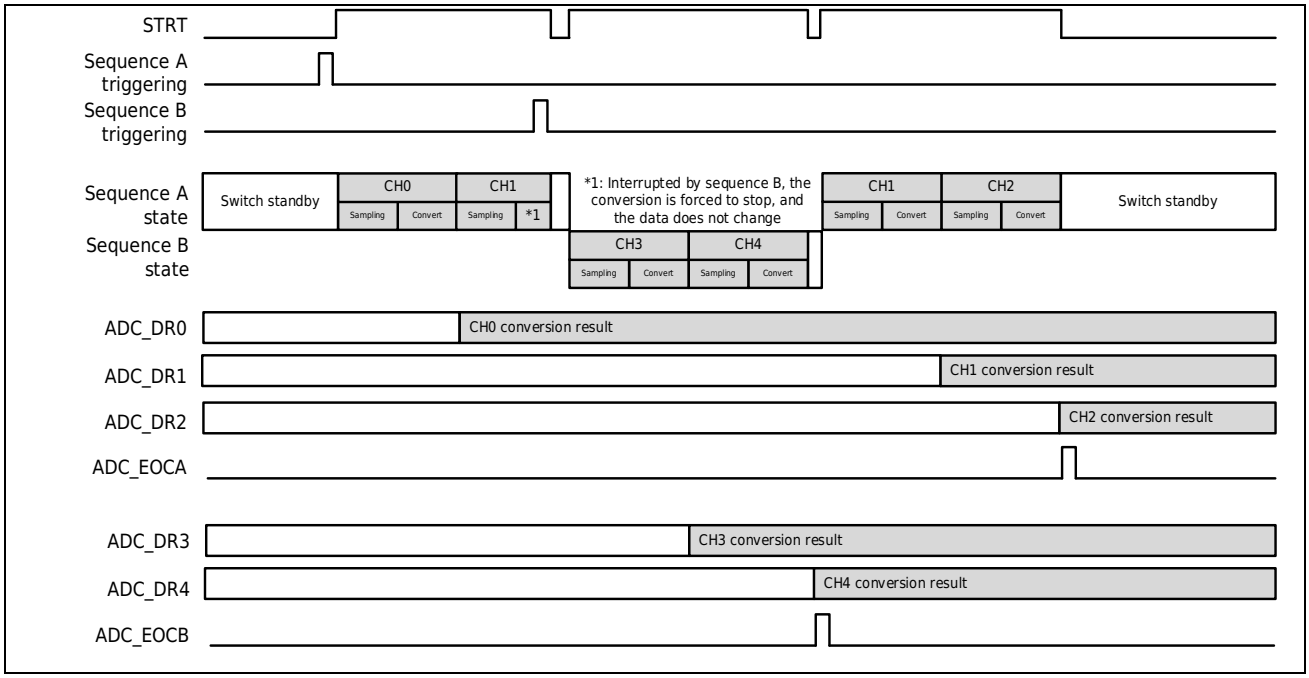


Figure 16-7 Double Sequence Scan Mode (Sequence A restarts from interrupted channel)

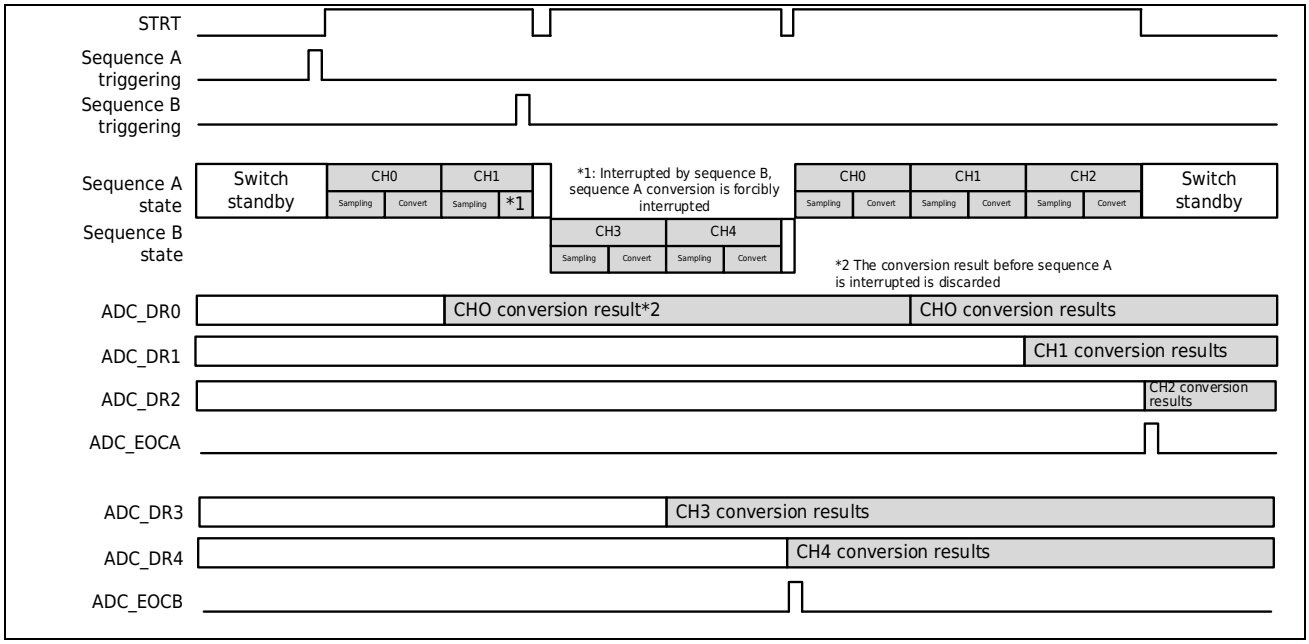


Figure 16-8 Double Sequence Scan Mode (Sequence A Restarts from the First Channel)

The software process of double sequence scanning mode:

1. Confirm that ADC_STR.STRT is 0, ADC is in conversion standby state;
2. ADC control register ADC_CR0.MS[2:0] is set to 0b010 or 0b011 to select dual sequence scanning mode;
3. Set register ADC_CR1.RSCHSEL selects the start mode after sequence A is broken.
4. Set the sequence A channel select register ADC_CHSELRA;
5. Set the sequence B channel select register ADC_CHSELRB.
6. Set the sampling time registerADC _ SSTR;
7. Set the register ADC _ TRGSR selection sequence A and B trigger conditions.
8. By querying EOCAF, EOCBF, or ADC _ EOCA, or ADC _ EOCBinterrupt, or starting DMA to process the converted data after the sequence A or B conversion ends.

Note:

Do not select the same channel in sequence A and B. Do not select the same trigger source for sequences A and B.

16.3.9 Analog Watchdog Function

The analog watchdog function is to compare the conversion results at the end of A/D conversion of the channel. The ADC supports two comparison windows: Comparison window 0 and comparison window 1. Take comparison window 0 as an example, as shown in the following figure, if the conversion result is in the protected area, the watchdog comparison interrupt and event ADC _ CMP0 are generated.

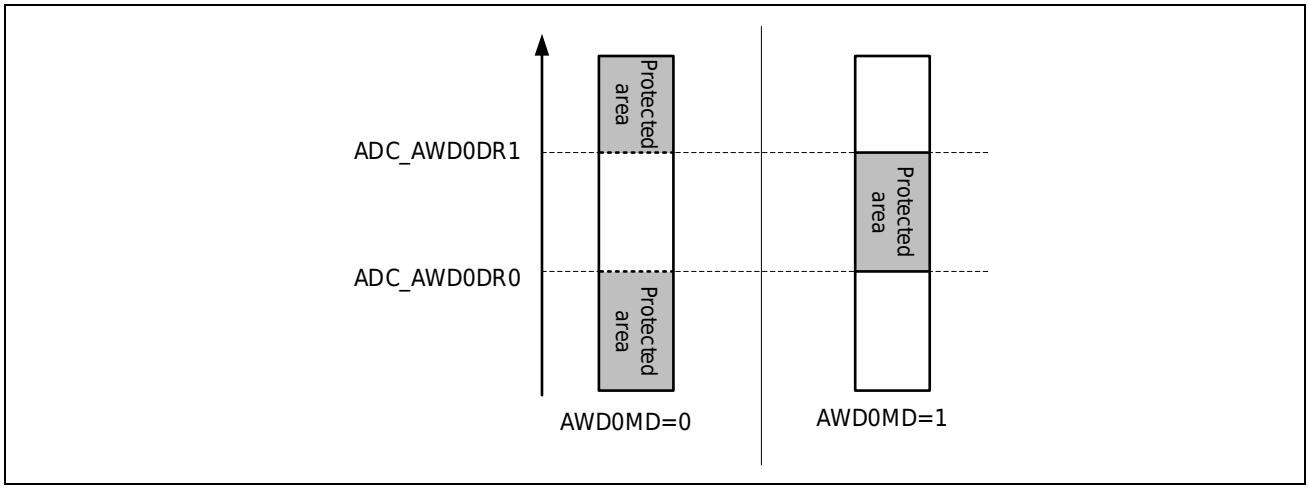


Figure 16-9 Analog Watchdog Protection Area (Compare Mode)

The software flow using the analogwatchdog function is illustrated by comparison window 0:

1. Set threshold registers AD_AWD0DR0, ADC_AWD0DR1;
2. Set the comparison channel registerAD _ AWD0 CHSR and select the channel to be compared
3. Set ADC_AWDCR.AWD0MD to select compare mode;
4. Set the ADC _ AWDCR.AWD0IENinterrupt permission bit;
5. Setting ADC_AWDCR.AWDEN allows analog watchdog function;
6. According to the previous text, set scan mode, start AD for conversion;
7. In ADC _ CMP0interrupt, or after ADC conversion, query the comparison state registerADC _ AWDSR.AWD0F and process the comparison results accordingly.

The comparison window 1 is used the same way as the comparison window 0.

The two comparison windows can be used in combination. When the window combination function is valid, at the end of the conversion of the channel selected by window 1, the comparison interrupt ADC_CMP1 outputs no longer the comparison result of window 1 alone, but the combined result generated by performing logical OR, logical AND or logical XOR on the comparison results of windows 0 and 1 according to the settings. The software flow of using the window combination comparison function is similar to that of using a single window. After setting windows 0 and 1, it is necessary to set the ADC_AWDCR.AWDCM[1:0] register to select the combination mode.

16.3.10 Sampling Time and Conversion Time of Analog Input

In single scan mode, ADC conversion can be started by software setting, internal trigger IN_TRGx0, 1 and external pin trigger ADTRGx. 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 not$t_D$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 cycles can be adjusted according to the input impedance setting register ADC_SSTR. t_{CMP} represents the successive comparison time, 12-bit precision is 13 PCLK2, 10-bit precision is 11 PCLK2, and 8-bit precision is 9 PCLK2.

The time of one scan conversion $t_{SCAN}=t_D+\sum t_{CONV}+t_{ED}$. Among them, $\sum t_{CONV}$ represents the sum of the conversion time of all scanning channels. Since the sampling timet_{SPL}can be set independently, the conversion time $t_{CONV}$of each channel can be different.

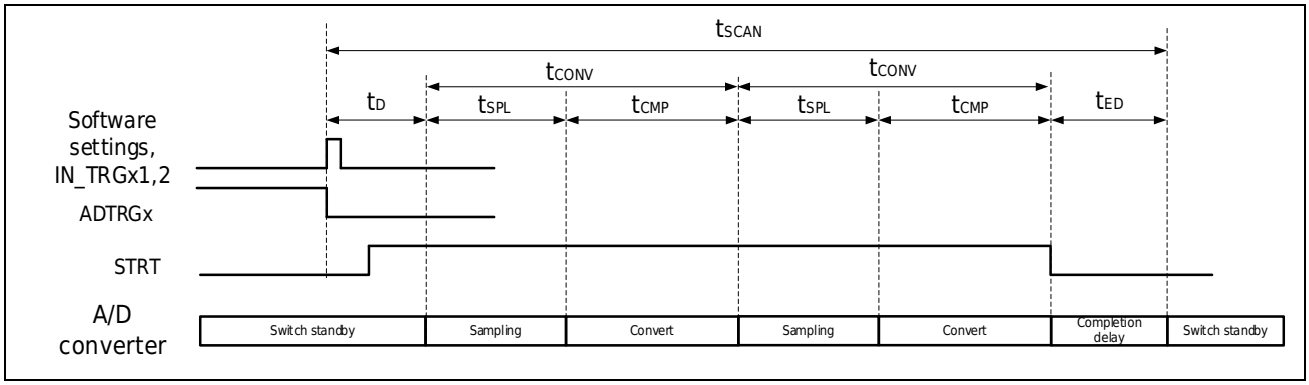


Figure 16-10 ADC Conversion Time

Table 16-3 AD Conversion Time

Marking		Description		Conditions			
				Synchronous Peripheral Trigger	**Asynchronous Peripheral Trigger *Note**	**External Pin Trigger**	**Software Triggering**
t_D		Scan start time	ADC is idle, start conversion	4 PCLK2	3 PCLK4 + 4 PCLK2 + 1 PCLK4_SYNC	2 PCLK4 + 4 PCLK2	3 PCLK2
			Interrupted in sequence A conversion, starting sequence B conversion	5 PCLK2	4 PCLK4 + 5 PCLK2 + 1 PCLK4_SYNC	2 PCLK4 + 5 PCLK2	-
t_{CONV}	t_{SPL}	Sampling time		ADSSTRx.SST[7:0] x PCLK2			
	t_{CMP}	Successive conversion time	12-bit resolution	13 PCLK2			
			10-bit resolution	11 PCLK2			
			8-bit resolution	9 PCLK2			
t_{ED}		Scan completion time		3 PCLK2			
t_{TD}		Minimum continuous trigger interval		$\sum t_{CONV}$ + 6 PCLK2			

Note:

Asynchronous peripheral triggering refers to the peripheral triggering situation when the ADC module selects the PLL clock operation that is asynchronous to the system clock. At this time, the

peripheral module clock and the ADC module clock are in an asynchronous relationship. PCLK4_SYNC represents the original synchronous clock of the ADC module (that is, the clock set by CMU_SCFGR). At this time, PCLK4 and PCLK2 are the same, both of which are asynchronous PLL clocks.

16.3.11 ADC Data Register Automatic Clear Function

When ADC_CR0.CLREN is 1, the ADC conversion data register ADC_DR will be automatically cleared to 0x0000 after being read by the CPU or DMA.

Using this feature, you can detect whether the data registerADC _ DR is updated. Examples are provided below.

- When ADC_CR0.CLREN is 0 and the auto-clear function disable, the analog quantity to be measured (0x0222) is not converted for some reason or the result is not updated to the data register ADC_DR, and the ADC_DR register continues to hold the previous conversion value (0x0111). The unupdated (0x0111) will be read during the ADC conversion completion interrupt processing. In order to detect whether the ADC conversion value valid, it is necessary to store the previous conversion value in RAM and make a judgment by comparing the conversion result.
- If ADC_CR0.CLREN is 1 and the auto-clear function is enabled, the ADC_DR register will be automatically cleared to 0x0000 after the previous conversion result (0x0111) is read by the CPU or DMA. After the ADC conversion, if the conversion result is not correctly transferred to the ADC_DR register, the ADC_DR register will remain at 0x0000. At this time, if 0x0000 is read during the interrupt processing, it will be easy to determine whether the ADC conversion data is stored correctly.

16.3.12 Conversion Data Average Calculation Function

The ADC conversion average calculation function is a function that performs 2, 4, 8, 16, 32, 64, 128 or 256 consecutive conversions on the same channel and saves the conversion results to the data register after averaging. Using the average calculation function can remove certain noise components to make the conversion result more accurate.

The register ADC_CR0.AVCNT[2:0] sets the number of continuous conversions, and the register ADC_AVCHSELR selects any one or more channels that need to be averaged.

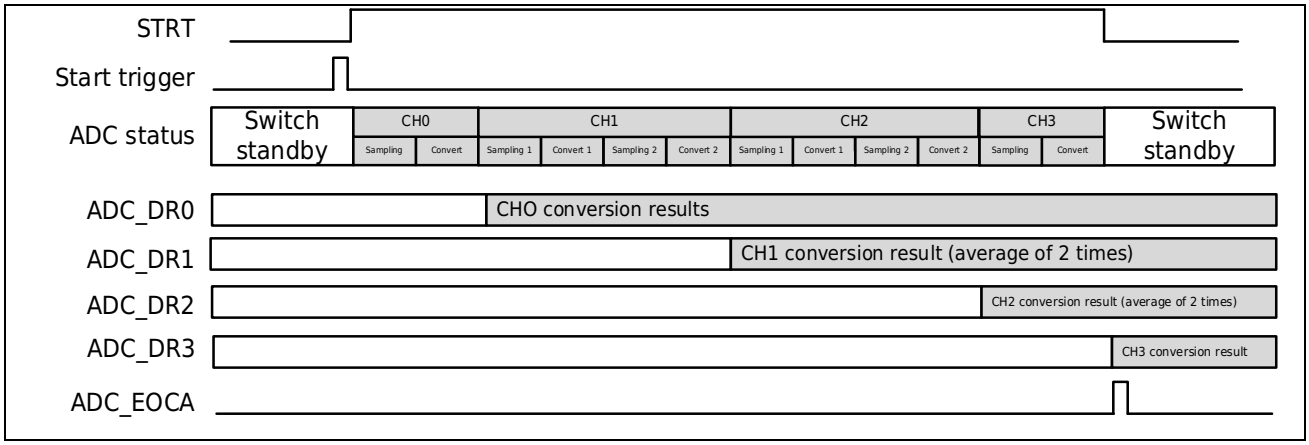


Figure 16-11 Conversion Action when Averaging Function is Valid

Figure 16-11 in the single scan mode of sequence A , choose to convert the 4 channels CH0~3 , among which CH1 and 2 are set to the 2 -time average mode. During the scanning process, CH1 and 2 will perform two consecutive conversions, and save the averaged results to the data registers ADC_DR1 and 2 of the corresponding channel.

16.3.13 Oversampling

By setting the register ADC_CR2.OVSMOD to 1 and setting the register ADC_AVCHSELR to select any one or more channels that need to be oversampled, the AD enters the oversampling mode. In this mode, a 16-bit precision conversion result is obtained through continuous multiple sampling conversions and result shifting.

The final conversion result is calculated using the following formula.

The sampling rate N is set by register ADC_CR0.AVCNT[2:0] and can be set from 2 to 256 samples.

The divisor series M is set by register ADC_CR2.OVSS[3:0] and can be set to shift right by up to 8 bits.

16.3.14 Multi-ADC Cooperative Working Mode

On chips equipped with two or three ADC modules, the ADC collaborative working mode can be used.

In the ADC collaborative working mode, ADC1 acts as the master control unit and synchronizes the conversion of ADC2 and ADC3 through the trigger signal of ADC1. That is, the setting of the sequence A trigger source selection register ADC_TRGSR.TRGSELA[2:0] of ADC2 and ADC3 is invalid. All ADC modules are triggered by the trigger source selected by the Sequence A trigger source selection register of ADC1. In this mode, writing 1 to the ADC_STR.STRT register will not start the conversion, that is, the software start is invalid.

When using the cooperative work mode, please prohibit the sequence B action, so as not to disturb the synchronization.

The two ADC modules ADC1 and ADC2 can be set to work together, or the three ADC modules ADC1, ADC2, and ADC3 can be set to work together. Depending on the specific specifications of the product, ADC3 may not be equipped.

The ADC can be configured into the following four cooperative working modes:

- One-Shot Parallel Trigger Mode
- One-shot delayed trigger mode
- Loop Parallel Trigger Mode
- Cyclic delay trigger mode

One-Shot Parallel Trigger Mode

The Sequence A trigger condition of ADC1 triggers all ADC modules in cooperative mode at the same time, and triggers only once.

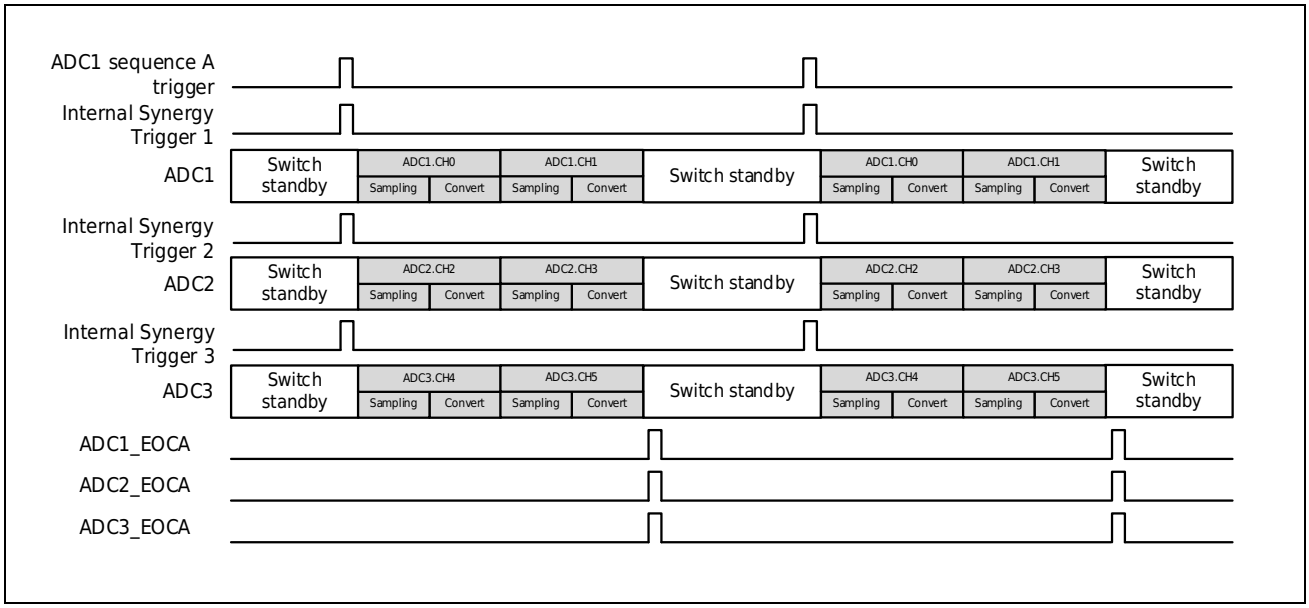


Figure 16-12 Single Parallel Trigger Mode (Three ADCs)

Note:

It is forbidden for multiple ADCs to convert the same analog input at the same time. An analog channel can only sample one ADC module at a time, otherwise the accuracy is not guaranteed. The same applies below.

The software setting process of this mode is as follows:

1. Write 0 to the collaborative work permission register ADC_SYNCCR.SYNCEN to confirm that the collaborative work is invalid;
2. Set up the ADC1 module;

a) Confirm that ADC1_STR.STRT is 0, and ADC1 is in conversion standby state

b) Set the control register ADC1_CR0.MS[2:0] to 0b000: sequence A single scan mode, or 0b001: sequence A continuous scan mode

c) Set Sequence A Channel Select Register ADC1_CHSELRA

d) Set the sampling time register ADC1_SSTR

e) Set the sequence A trigger source selection register ADC1_TRGSR

3. Set up the ADC2 module;

a) Confirm that ADC2_STR.STRT is 0, and ADC2 is in conversion standby state

b) Set the control register ADC2_CR0.MS[2:0], the channel selection register ADC2_CHSELRA, and the channel sampling time register ADC2_SSTR.

Note:

In order to ensure the synchronous operation of ADC2 and ADC1, the above registers should be set to the same value as the registers of ADC1 as far as possible. The specific channels do not need to be the same, as long as the number of channels and the sampling time of the corresponding channels are consistent.

4. Set the ADC3 module (when three ADCs work together);

a) Confirm that ADC3_STR.STRT is 0, and ADC3 is in conversion standby state

b) Set the control register ADC3_CR0.MS[2:0], the channel selection register ADC3_CHSELRA, and the channel sampling time register ADC3_SSTR.

Note:

Same as ADC2, in order to ensure the synchronous operation of ADC3 and ADC1, the above registers should be set to the same value as the registers of ADC1 as much as possible.

5. Set the synergy mode control register ADC_SYNCCR.SYNCMD[2:0], write 0b010: ADC1 and ADC2 work together. Or write 0b011: ADC1, ADC2, ADC3 three ADCs work together;

6. Write 1 to the collaborative work permission register ADC_SYNCCR.SYNCEN, and the collaborative work is valid;

7. Wait for ADC1 sequence A trigger source input, and process the results after ADC1, 2, and 3 complete conversion.

One-shot delayed trigger mode

After the sequence A trigger condition of ADC1 triggers ADC1, it triggers ADC2 to start conversion after a set delay, and then triggers ADC3 to start conversion after a set delay. Each ADC module is only triggered once.

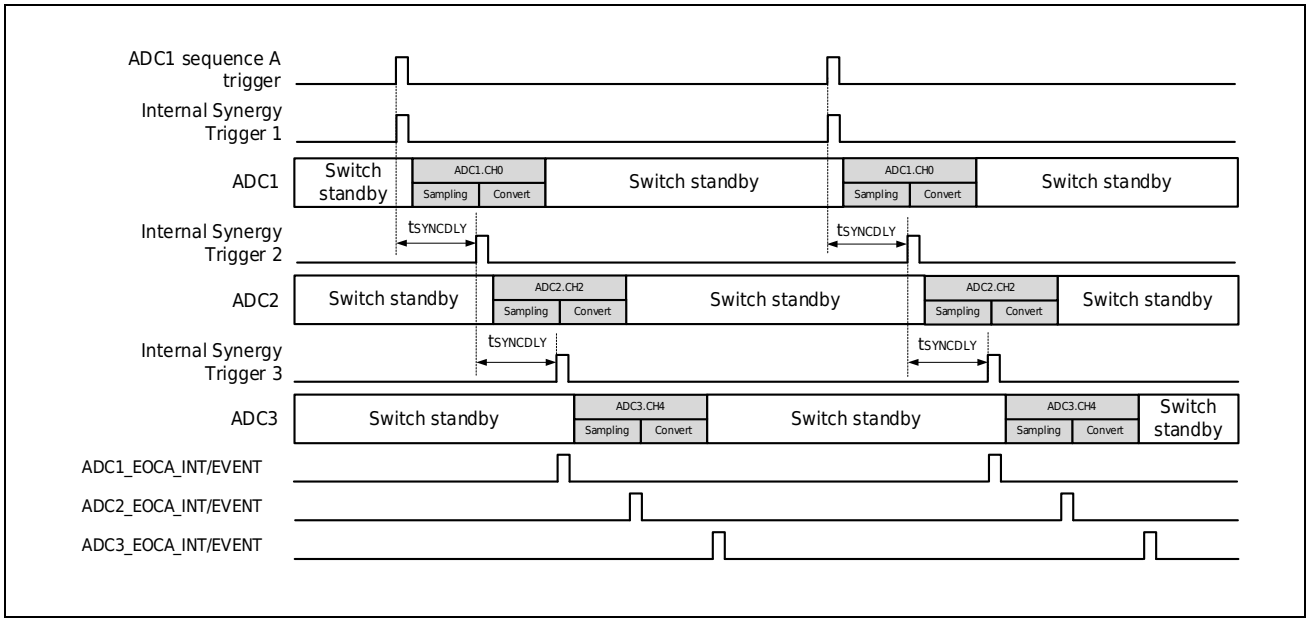


Figure 16-13 Single delay trigger mode (three ADCs)

Note:

1. *After the ADC1 sequence A trigger is input for the first time, before the ADC3 coordinated trigger occurs, inputting the ADC1 sequence A trigger again will be ignored.*
2. *If each ADC unit converts the same analog channel, the sampling time needs to be staggered, that is, the delay time $t_{SYNCDLY}$ and the channel opening sampling time t_{SPL} need to satisfy: $t_{SYNCDLY} > t_{SPL}$.*

The software setting process of this mode is as follows:

1. Write 0 to the collaborative work permission register ADC_SYNCCR.SYNCEN to confirm that the collaborative work is invalid;
2. Set up ADC1, 2, 3 modules (refer to single parallel mode);
3. Set the cooperative mode control register ADC_SYNCCR.SYNCDLY[7:0] to set the startup delay of the two ADCs;
4. Set the cooperative mode control register ADC_SYNCCR.SYNCMD[2:0], write 0b000: ADC1 and ADC2 work together. Or write 0b001: ADC1, ADC2, ADC3 three ADCs work together;
5. Write 1 to the collaborative work permission register ADC_SYNCCR.SYNCEN, and the collaborative work is valid;
6. Wait for ADC1 sequence A trigger source input, and process the results after ADC1, 2, and 3 complete conversion.

Loop Parallel Trigger Mode

The Sequence A trigger condition of ADC1 triggers all ADC modules in cooperative mode simultaneously, and then triggers all ADC modules simultaneously again after each specified delay. Until the user initiates the software to stop the ADC1 module, or prohibits the cooperative work mode.

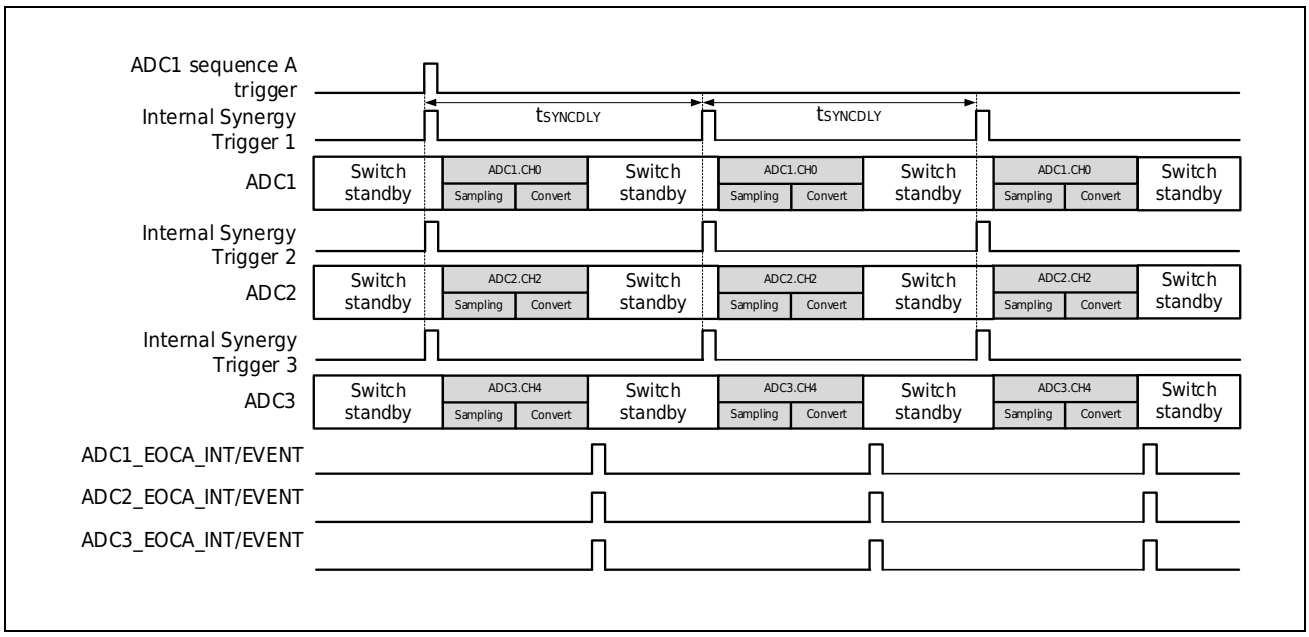


Figure 16-14 Cyclic Parallel Trigger Mode (Three ADCs)

Note:

The delay time $t_{SYNCDLY}$ and the time $t_{SCAN}$of a scan conversion need to satisfy: $t_{SYNCDLY}$>t_{SCAN}.

The software setting process of this mode is as follows:

1. Write 0 to the collaborative work permission register ADC_SYNCCR.SYNCEN to confirm that the collaborative work is invalid;
2. Set up ADC1, 2, 3 modules, refer to single parallel mode. ADC_CR0.MS[2:0] is set to 0b000: sequence A single scan mode;
3. Set the collaborative mode control register ADC_SYNCCR.SYNCDLY[7:0] to set the delay of each parallel trigger;
4. Set the cooperative mode control register ADC_SYNCCR.SYNCMD[2:0], write 0b110: ADC1 and ADC2 work together. Or write 0b111: ADC1, ADC2, ADC3 three ADCs work together;
5. Write 1 to the collaborative work permission register ADC_SYNCCR.SYNCEN, and the collaborative work is valid;
6. Wait for ADC1 sequence A trigger source input, and process the results after ADC1, 2, and 3 complete conversion.

Cyclic Delay Trigger Mode

After ADC1 is triggered by the sequence A trigger condition, ADC2, ADC3, ADC1, ADC2, etc. are triggered in a cycle after each set delay until the user actively stops the ADC1 module or prohibits the collaborative working mode.

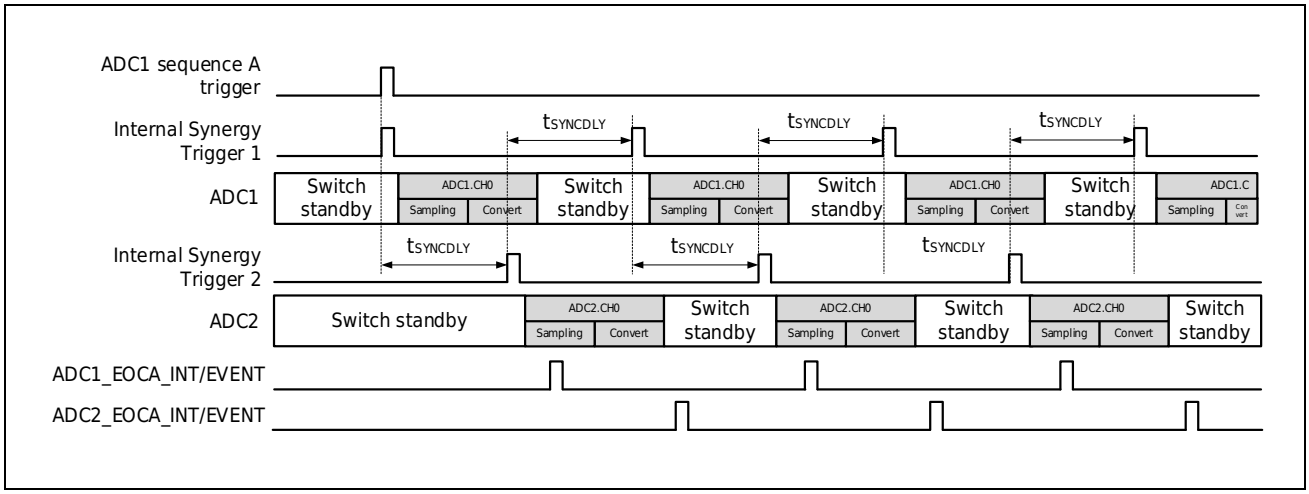


Figure 16-15 Cyclic Delay Trigger Mode (Two ADCs)

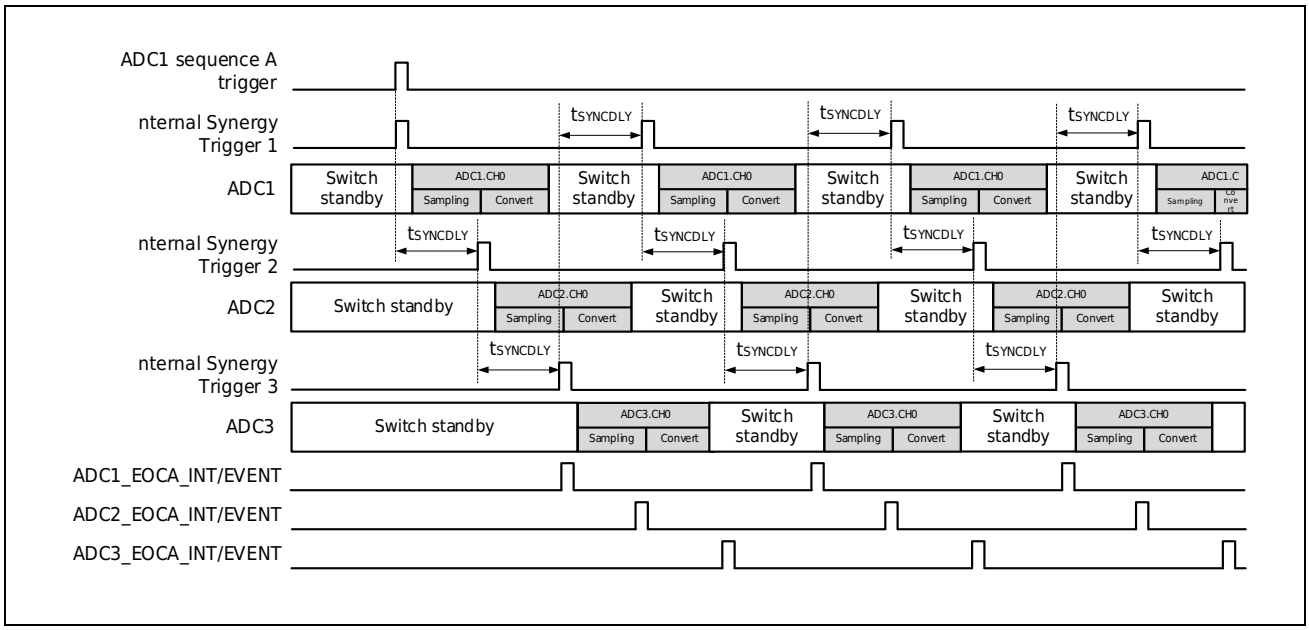


Figure 16-16 Cyclic Delay Trigger Mode (Three ADCs)

Note:

When the two ADCs work together, The delay time $t_{SYNCDLY}$and the time $t_{SCAN}$of a scan conversion need to satisfy: $t_{SYNCDLY}>t_{SCAN}/2$. When the three ADCs work together, it needs to satisfy: $t_{SYNCDLY}>t_{SCAN}/3$. At the same time, if ADC1, ADC2, and ADC3 convert the same analog channel, the sampling time needs to be staggered, that is, $t_{SYNCDLY}>t_{SPL}$.

The software setting process of this mode is as follows:

1. Write 0 to the collaborative work permission register ADC_SYNCCR.SYNCEN to confirm that the collaborative work is invalid;
2. Set ADC1, 2, 3 modules, refer to the circular parallel trigger mode;
3. Set the collaborative mode control register ADC_SYNCCR.SYNCDLY[7:0] to set the delay of each trigger;
4. Set the cooperative mode control register ADC_SYNCCR.SYNCMD[2:0], write 0b100: ADC1 and ADC2 work together. Or write 0b101: ADC1, ADC2, ADC3 three ADCs work together;
5. Write 1 to the collaborative work permission register ADC_SYNCCR.SYNCEN, and the collaborative work is valid;
6. Wait for ADC1 sequence A trigger source input, and process the results after ADC1, 2, and 3 complete conversion.

16.4Interrupt and Events

The ADC module can produce the following four types of event outputs. When each event occurs, if the corresponding interrupt permission register is set to valid, the interrupt application is also output.

1) Sequence A Scan End ADC _ EOCA, corresponding to interrupt enable registerADC _ ICR.EOCAIEN
2) Sequence B Scan End ADC _ EOCB, corresponding to interrupt enable registerADC _ ICR.EOCBIEN
3) Analog watchdog 0 ADC_CMP0, corresponding to the interrupt enable register ADC_AWDCR.AWD0IEN
4) Analog watchdog 1 ADC_CMP1, corresponding to the interrupt enable register ADC_AWDCR.AWD1IEN

These four event outputs can start other on-chip peripheral modules, including DMA transfer. DMA transfer can be used to continuously read ADC conversion results without software intervention. It is completely implemented by hardware and reduces the CPU load. For DMA settings, please refer to the [DMA Controller (DMA)] section. The event signal output has nothing to do with the control of the interrupt enable bit, as long as the condition occurs, it will be output.

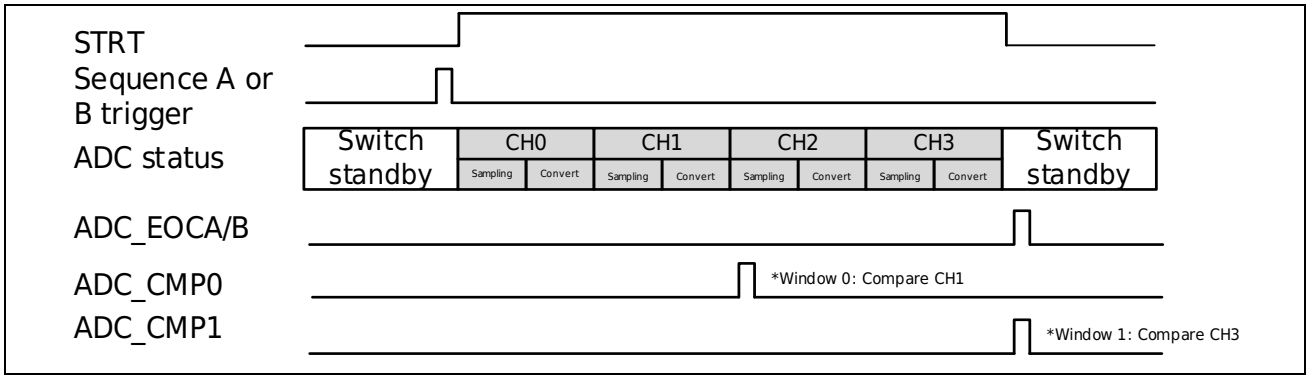


Figure 16-17 ADC interrupt and event output timing

16.5Register Description

Table 16-4 ADC Base Address

Name	Base Address	Description
ADC1	0x4004 0000	ADC1 base address
ADC2	0x4004 0400	ADC2 base address
ADC3	0x4004 0800	ADC3 base address

Table 16-5 ADC Register List

Register	Description	Offset Address	Bit Width	Reset Value
ADC_STR	ADC start register	0x00	8	0x00
ADC_CR0	ADC Control Register0	0x02	16	0x0000
ADC_CR1	ADC Control Register1	0x04	16	0x0000
ADC_CR2	ADC Control Register2	0x06	16	0x0000
ADC_TRGSR	ADC Conversion Start Trigger Register	0x0a	16	0x0000 0000
ADC_CHSELRA	ADC Channel Selection Register A	0x0c	32	0x0000 0000
ADC_CHSELRB	ADC Channel Selection Register B	0x10	32	0x0000 0000
ADC_AVCHSELR	ADC Mean Channel Selection Register	0x14	32	0x0000 0000
ADC_EXCHSELR	ADC Extended Channel Selection Register	0x18	8	0x00
ADC_SSTRx (1)	ADC Sampling Period Register	0x20+x	8	0x0B
ADC_CHMUXR0	ADC Channel Mapping Control Register 0	0x38	16	0x3210
ADC_CHMUXR1	ADC Channel Mapping Control Register 1	0x3A	16	0x7654
ADC_CHMUXR2	ADC Channel Mapping Control Register 2	0x3C	16	0xBA98
ADC_CHMUXR3	ADC Channel Mapping Control Register 3	0x3E	16	0xFEDC
ADC_ISR	ADC Interruption Status Register	0x44	8	0x00
ADC_ICR	ADC interrupt enable register	0x45	8	0x03
ADC_ISCLRR	ADC Interrupt Status Reset Register	0x46	8	0x00
ADC_SYNCCR	ADC Cooperative Mode Control Register	0x4C	16	0x0C00
ADC_DRx (1)	ADC Data Register	0x50+2*x	16	0x0000
ADC_AWDCR	Analog watchdog control register	0xA0	16	0x0000
ADC_AWDSR	Analogwatchdog status register	0xA2	8	0x00
ADC_AWDSCLRR	Analog watchdog status reset register	0xA3	8	0x00
ADC_AWD0DR0	Analog Watchdog Window 0 Threshold Register 0	0xA4	16	0x0000
ADC_AWD0DR1	Analog Watchdog Window 0 Threshold Register 1	0xA6	16	0xFFFF
ADC_AWD0CHSR	Analog watchdog window 0 comparison channel selection register	0xA8	8	0x00
ADC_AWD1DR0	Analog Watchdog Window 1 Threshold Register 0	0xAC	16	0x0000
ADC_AWD1DR1	Analog Watchdog Window 1 Threshold Register 1	0xAE	16	0xFFFF

Register	Description	Offset Address	Bit Width	Reset Value
ADC_AWD1CHSR	Analog watchdog window 1 comparison channel selection register	0xB0	8	0x00

Note: The x in ADC_SSTRx and ADC_DRx represents the channel, x=0~15 in ADC1, x=0~15 in ADC2, and x=0~11 in ADC3.

16.5.1 ADC Start Register (ADC_STR)

Reset Value: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
Res							STRT

Bit/Bitfield	Marking	Bit Name	Function	RW
b7~b1	Res	Reserved Bit	Read at 0, write at 0	RW
b0	STRT	Start of AD conversion	0: Stop conversion 1: Start conversion Set the "1" condition: (1) Software setup (2) The selected trigger condition occurs. (3) ADC conversion Clear"0"condition: (1) Software "0" (2) "0" will be cleared automatically after conversion. **Note:** Write 1 generates a software trigger when STRT is 0 (ADC is idle), starting sequence A Write 1 is invalid when STRT is 1 (in ADC action). Write 0 when STRT is 1 to force the stop of AD conversion. 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 is invalid when STRT is 0.	RW

16.5.2 ADC Control Register 0 (ADC_CR0))

Reset value: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8
Res					AVCNT[2:0]		
b7	b6	b5	b4	b3	b2	b1	b0
DFMT	CLREN	ACCSEL[1:0]		Res	MS[2:0]		

Bit/Bitfield	Marking	Bit Name	Function	RW
b15~b11	Res	Reserved Bit	Read at 0, write at 0	RW
b10~b8	AVCNT[2:0]	Times selection	000: Average of 2 consecutive conversions 001: Average of 4 consecutive conversions 010: Average of 8 consecutive conversions 011: Average of 16 consecutive conversions 100: Average of 32 consecutive conversions 101: Average of 64 consecutive conversions 110: Average of 128 consecutive conversions 111: Average of 256 consecutive conversions	RW
b7	DFMT	Data format	0: Right alignment of converted data 1: Left alignment of converted data **Note:** In oversampling mode, this bit setting invalid.	RW
b6	CLREN	Data register automatic purge	0: Automatic removal of prohibitions 1: Automatic clearance **Note:** After the CLREN bit is set, registerADC _ DRx is automatically cleared after CPU, DMA, etc. The automatic clearing function is mainly used to detect whether the data register is updated.	RW
b5~b4	ACCSEL[1:0]	Resolution selection	00: 12-bit resolution 01: 10-bit resolution 10: 8-bit resolution 11: Setting prohibited **Note:** In oversampling mode, please set this bit to 00.	RW
b3	Res	Reserved Bit	Read at 0, write at 0	RW
b2~b0	MS[2:0]	Pattern selection	000: Sequence A single scan mode, sequence B invalid 001: Sequence A continuous scan mode, sequence B invalid 010: Sequence A single scan mode, Sequence B single scan mode 011: Sequence A continuous scan mode, sequence B single scan mode 100: Sequence A data buffer mode, sequence B invalid 110: Sequence A data buffer mode, sequence B single scan mode Other setting values are prohibited	RW

Note:

Set this register when ADC _ STR. STRT is "0".

16.5.3 ADC control register 1 (ADC_CR1)

Reset value: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8
Res							
b7	b6	b5	b4	b3	b2	b1	b0
Res					RSCHSEL	Res	

Bit/Bitfield	Marking	Bit Name	Function	RW
b15~b3	Res	Reserved Bit	Read at 0, write at 0	RW
b2	RSCHSEL	Sequence A Restart Channel Selection	0: After sequence Binterrupt, sequence A continues to scan from the interrupt channel when it restarts 1: Rescan from the first channel when sequence A restarts after sequence Binterrupt	RW
b1~b0	Res	Reserved Bit	Read at 0, write at 0	RW

Note:

Set this register when ADC _ STR. STRT is "0".

16.5.4 ADC Control Register 2 (ADC_CR2)

Reset value: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8
Res			OVSMOD	OVSS[3:0]			
b7	b6	b5	b4	b3	b2	b1	b0
Res							

Bit/Bitfield	Marking	Bit Name	Function	RW
b15~b13	Res	Reserved Bit	Read at 0, write at 0	RW
b12	OVSMOD	Oversampling Mode	0: Regular Mode 1: Oversampling mode	RW
b11~b8	OVSS[3:0]	Shift Selection	0000: No shift 0001: Shift right 1 bit 0010: Shift right 2 bit 0011: Shift right 3 bit 0100: Shift right 4 bit 0101: Shift right 5 bit 0110: Shift right 6 bit 0111: Shift right 7 bit 1000: shift right 8 bits **Note:** When OVSMOD=0 (Regular Mode), this bit setting invalid.	RW
b7~b0	Res	Reserved Bit	Read at 0, write at 0	RW

Note:

1. *Set this register when ADC _ STR. STRT is "0".*
2. *Please use 16bit.*

16.5.5 ADC Conversion Start Trigger Register (ADC_TRGSR)

Reset value: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8
TRGENB	Res					TRGSELB[1:0]	
b7	b6	b5	b4	b3	b2	b1	b0
TRGENA	Res					TRGSELA[1:0]	

Bit/Bitfield	Marking	Bit Name	Function	RW
b15	TRGENB	Sequence B trigger enable	0: Sequence B in-chip or external pin trigger disable 1: Sequence B in-chip or external pin trigger enable **Note:** Select external pin trigger to be valid. If ADTRGx is changed from "High" to "Low" and a falling edge is detected, the scan conversion begins. Keep "Low" 1.5 * PCLK4 cycles above.	RW
b14~b10	Res	Reserved Bit	Read at 0, write at 0	RW
b9~b8	TRGSELB[1:0]	Sequence B trigger condition selection	In sequence B active mode (ADC _ CR0.MS [1] = 1) as trigger condition of sequence B 00：ADTRGx 01：IN_TRGx0 10：IN_TRGx1 11：IN_TRGx0 + IN_TRGx1 **Note:** Only valid in Sequence B valid mode. Other mode settings are invalid. The interval between triggers must be greater than or equal to the scan period t_{SCAN}, and if less, the trigger is invalid.	RW
b7	TRGENA	Sequence A trigger enable	0: Sequence A in-chip or external pin trigger disable 1: Sequence A in-chip or external pin trigger enable **Note:** Select external pin trigger to be valid. If ADTRGx is changed from "High" to "Low" and a falling edge is detected, the scan conversion begins. Keep "Low" 1.5 * PCLK4 cycles above.	RW
b6~b2	Res	Reserved Bit	Read at 0, write at 0	RW
b1~b0	TRGSELA[1:0]	Sequence A trigger condition selection	Trigger condition for sequence A. 00: ADTRGx (x = 1~3, represents the ADC unit number) 01：IN_TRGx0 10：IN_TRGx1 11：IN_TRGx0 + IN_TRGx1 **Note:** When the ADC is idle, writing 1 to ADC_STR.STRT is a software trigger, ignoring the settings of TRGENA and TRGSELA[1:0], and directly starting ADC conversion. The interval between triggers must be greater than or equal to the scan period t_{SCAN}, and if less, the trigger is invalid.	RW

Note:

Set this register when ADC _ STR. STRT is "0".

16.5.6 ADC Channel Select Register A (ADC_CHSELRA)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24
CHSELA[31:24]							
b23	b22	b21	b20	b19	b18	b17	b16
CHSELA[23:16]							
b15	b14	b13	b12	b11	b10	b9	b8
CHSELA[15:8]							
b7	b6	b5	b4	b3	b2	b1	b0
CHSELA[7:0]							

Bit/Bitfield	Marking	Bit Name	Function	RW
b31~b0	CHSELA[31:0]	Switching channel selection	Channel selection for sequence A, each representing a channel, CHSELA [x] representing a channel CHx, can be any combination. 0: No channel selected 1: Select Corresponding Channels The corresponding bits of non-existent channels are reserved bits, which are 0 when read and 0 when written. **Note:** Please do not select the same channel in sequence A and sequence B.	RW

Note:

Set this register when ADC _ STR. STRT is "0".

16.5.7 ADC Channel Select Register B (ADC_CHSELRB)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24
CHSELB[31:24]							
b23	b22	b21	b20	b19	b18	b17	b16
CHSELB[23:16]							
b15	b14	b13	b12	b11	b10	b9	b8
CHSELB[15:8]							
b7	b6	b5	b4	b3	b2	b1	b0
CHSELB[7:0]							

Bit/Bitfield	Marking	Bit Name	Function	RW
b31~b0	CHSELB[31:0]	Switching channel selection	Channel selection for sequence B, each representing a channel, CHSELB [x] representing a channel CHx, can be any combination. Only valid in double sequence scan mode. 0: No channel selected 1: Select Corresponding Channels The corresponding bits of non-existent channels are reserved bits, which are 0 when read and 0 when written. **Note:** Please do not select the same channel in sequence A and sequence B.	RW

Note:

Set this register when ADC _ STR. STRT is "0".

16.5.8 ADC Average Channel Select Register (ADC_AVCHSELR)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24
AVCHSEL[31:24]							
b23	b22	b21	b20	b19	b18	b17	b16
AVCHSEL[23:16]							
b15	b14	b13	b12	b11	b10	b9	b8
AVCHSEL[15:8]							
b7	b6	b5	b4	b3	b2	b1	b0
AVCHSEL[7:0]							

Bit/Bitfield	Marking	Bit Name	Function	RW
b31~b0	AVCHSEL[31:0]	Average channel selection	Each bit represents a channel, and AVCHSEL[x] represents the channel CHx, and any combination can be selected. 0: No channel selected 1: Select Corresponding Channels The corresponding bits of non-existent channels are reserved bits, which are 0 when read and 0 when written. **Note:** When AVCHSEL and the corresponding channels of ADC_CHSELRA or ADC_CHSELB are selected at the same time, the channel will continuously perform the set number of ADC conversions during scanning, and update the conversion results into the data register after averaging. If the corresponding channel AVCHSEL is not set, the channel will perform a normal conversion.	RW

Note:

Set this register when ADC _ STR. STRT is "0".

16.5.9 ADC Extended Channel Select Register (ADC_EXCHSELR)

Reset Value: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
Res							EXCHSEL

Bit/Bitfield	Marking	Bit Name	Function	RW
b7~b1	Res	Reserved Bit	Read at “0”, write at “0”	RW
b0	EXCHSEL	Extended channel selection	ADCx_IN15 and on-chip analog channel selection 0: Select external pin ADC1_IN15 or ADC2_IN7 or ADC3_IN11 1: Select the internal analog channel. For the setting of the internal analog source, please refer to the description of [Power Control (PWC)]	RW

Note:

Set this register when ADC _ STR. STRT is "0".

16.5.10 ADC Sampling Status Register (ADC_SSTRx)

Reset Value: 0x0B

b7	b6	b5	b4	b3	b2	b1	b0
SST[7:0]							

Bit/Bitfield	Marking	Bit Name	Function	RW
b7~b0	SST[7:0]	Number of sampling cycles	The number of sampling cycles can be set from 5 to 255 cycles. Channels CH0~15 are set by ADC_SSTRx, x=0~15 respectively. **Note:** When the PCLK2 frequency is 50MHz, one sampling period is 20ns, and the initial conversion state has 11 sampling periods. Register can be set to adjust sampling time when external input impedance R_{AIN} is too large or PCLK2 frequency is low. The sampling time should not be less than 5 cycles. SST ≧ $(R_{AIN}+R_{ADC})*C_{ADC}*\ln(2^{N+2})*f_{ADC}+1$ 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 PCLK2 frequency (Hz). Please refer to the corresponding **Data Sheet** for detailed description of electrical features.	RW

Note:

1. *Set this register when ADC _ STR. STRT is "0".*
2. *The sampling time of the internal reference voltage should not be less than 1μs.*
3. *The "x" in ADC_SSTRx represents the channel, x=0~15 in ADC1, x=0~15 in ADC2, and x=0~11 in ADC3.*

16.5.11 ADC Channel Mapping Control Register (ADC_CHMUXR)

ADC_CHMUXR0 reset value: 0x3210

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
CH03MUX[3:0]				CH02MUX[3:0]				CH01MUX[3:0]				CH00MUX[3:0]			

ADC_CHMUXR1 reset value: 0x7654

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
CH07MUX[3:0]				CH06MUX[3:0]				CH05MUX[3:0]				CH04MUX[3:0]			

ADC_CHMUXR2 reset value: 0xBA98

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
CH11MUX[3:0]				CH10MUX[3:0]				CH09MUX[3:0]				CH08MUX[3:0]			

ADC_CHMUXR3 reset value: 0xFEDC

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
CH15MUX[3:0]				CH14MUX[3:0]				CH13MUX[3:0]				CH12MUX[3:0]			

<table>
<tr><th>Bit/Bitfield</th><th>Marking</th><th>Bit Name</th><th>Function</th><th>RW</th></tr>
<tr><td>b15~b12
b11~b8
b7~b4
b3~b0</td><td>CHxMUX[3:0]</td><td>Channel x mapping selection (x=0~15)</td><td>The corresponding bit of the channel that does not exist is 0 when reading, and writes 0 when writing
The mapping relationship for different ADC units CHx is as follows:
<table>
<tr><th>Set value</th><th>ADC1 mapping object</th><th>ADC2 mapping object</th><th>ADC3 mapping object</th></tr>
<tr><td>0x0</td><td>ADC1_IN0</td><td>-</td><td>ADC3_IN0</td></tr>
<tr><td>0x1</td><td>ADC1_IN1</td><td>-</td><td>ADC3_IN1</td></tr>
<tr><td>0x2</td><td>ADC1_IN2</td><td>-</td><td>ADC3_IN2</td></tr>
<tr><td>0x3</td><td>ADC1_IN3</td><td>-</td><td>ADC3_IN3</td></tr>
<tr><td>0x4</td><td>ADC12_IN4</td><td>ADC12_IN4</td><td>-</td></tr>
<tr><td>0x5</td><td>ADC12_IN5</td><td>ADC12_IN5</td><td>-</td></tr>
<tr><td>0x6</td><td>ADC123_IN6</td><td>ADC123_IN6</td><td>ADC123_IN6</td></tr>
<tr><td>0x7</td><td>ADC123_IN7</td><td>ADC123_IN7</td><td>ADC123_IN7</td></tr>
<tr><td>0x8</td><td>ADC123_IN8</td><td>ADC123_IN8</td><td>ADC123_IN8</td></tr>
<tr><td>0x9</td><td>ADC123_IN9</td><td>ADC123_IN9</td><td>ADC123_IN9</td></tr>
<tr><td>0xA</td><td>ADC12_IN10</td><td>ADC12_IN10</td><td>ADC3_IN10</td></tr>
<tr><td>0xB</td><td>ADC12_IN11</td><td>ADC12_IN11</td><td>ADC3_IN11/ Internal analog channel</td></tr>
<tr><td>0xC</td><td>ADC12_IN12</td><td>ADC12_IN12</td><td>-</td></tr>
<tr><td>0xD</td><td>ADC12_IN13</td><td>ADC12_IN13</td><td>-</td></tr>
<tr><td>0xE</td><td>ADC12_IN14</td><td>ADC12_IN14</td><td>-</td></tr>
<tr><td>0xF</td><td>ADC12_IN15/ Internal analog channel</td><td>ADC12_IN15/ Internal analog channel</td><td>-</td></tr>
</table>
Note:
Please do not set it to a non-existent analog input.</td><td>RW</td></tr>
</table>

Note:

1. *Set this register when ADC _ STR. STRT is "0".*
2. *Unit 3 is only configured with ADC_CHMUXRn (n=0~2).*

16.5.12 ADC Interrupt Status Register (ADC_ISR)

Reset Value: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
Res			SASTPDF	Res		EOCBF	EOCAF

Bit/Bitfield	Marking	Bit Name	Function	RW
b7~b5	Res	Reserved Bit	Read at 0, write at 0	RW
b4	SASTPDF	Sequence A is interrupted	In dual sequence scanning mode, the bit is set to 1 when sequence A is interrupted by sequence B with higher priority during scanning. This register bit is read-only	R
b3~b2	Res	Reserved Bit	Read at 0, write at 0	RW
b1	EOCBF	Sequence B conversion completion flag	Sequence B Scan all channels selected 1 This register bit is read-only	R
b0	EOCAF	Sequence A conversion completion flag	Sequence A Selected Channels Scan All Post-1 This register bit is read-only	R

16.5.13 ADC Interrupt Enable Register (ADC_ICR)

Reset Value: 0x03

b7	b6	b5	b4	b3	b2	b1	b0
Res						EOCBIEN	EOCAIEN

Bit/Bitfield	Marking	Bit Name	Function	RW
b7~b2	Res	Reserved Bit	Read at 0, write at 0	RW
b1	EOCBIEN	Sequence B conversion complete interrupt enable	0: Sequence B conversion complete interrupt disable 1: Sequence B conversion complete interrupt enable	RW
b0	EOCAIEN	Sequence A conversion complete interrupt enable	0: Sequence A conversion complete interrupt disable 1: Sequence A conversion complete interrupt enable	RW

16.5.14 ADC Interrupt Status Reset Register (ADC_ISCLRR)

Reset Value: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
Res			CLRSASTPDF	Res		CLREOCBF	CLREOCAF

Bit/Bitfield	Marking	Bit Name	Function	RW
b7~b5	Res	Reserved Bit	Read at 0, write at 0	RW
b4	CLRSASTPDF	Sequence A interrupted flag reset	Writing 0 has no effect, writing 1 resets the SASTPDF status bit. Read always 0	W
b3~b2	Res	Reserved Bit	Read at 0, write at 0	RW
b1	CLREOCBF	Sequence B conversion complete flag reset	Write 0 has no effect. Write 1 resets the EOCBF status bit. Read always 0	W
b0	CLREOCAF	Sequence A conversion complete flag reset	Write 0 has no effect. Write 1 resets the EOCAF status bit. Read is always 0.	W

16.5.15 ADC Synergy Mode Control Register (ADC_SYNCCR)

Reset Value: 0x0c00

b15	b14	b13	b12	b11	b10	b9	b8
SYNCDLY[7:0]							
b7	b6	b5	b4	b3	b2	b1	b0
Res	SYNCMD[2]	SYNCMD[1]	SYNCMD[0]	Res			SYNCEN

Bit/Bitfield	Marking	Bit Name	Function	RW
b15~b8	SYNCDLY[7:0]	Synchronization delay time	In delayed trigger mode, the start-up delay time of the two ADCs is $t_{SYNCDLY}$. 0x1 means $t_{SYNCDLY}$ = 1*PCLK2, 0xff means $t_{SYNCDLY}$ = 255*PCLK2 **Note:** This register is set when SYNCEN is "0". Please do not write 0x00. According to the sampling time and conversion time of each ADC, set a reasonable delay time to avoid the error increase caused by multiple ADCs being in the sampling state at the same time, and avoid triggering again before the conversion of the ADC is completed, resulting in synchronization failure. The recommended settings are as follows: One-shot delay mode: $t_{SYNCDLY} > t_{SPL}$ Two ADC cycle delay trigger mode: $t_{SYNCDLY} > t_{SPL}$, and $t_{SYNCDLY} > t_{SCAN}/2$ Three ADC cycle delay trigger mode: $t_{SYNCDLY} > t_{SPL}$, and $t_{SYNCDLY} > t_{SCAN}/3$ Single parallel trigger mode: the setting of this register is invalid. Cyclic parallel trigger mode: $t_{SYNCDLY} > t_{SCAN}$	RW
b7	Res	Reserved Bit	Read at 0, write at 0	RW
b6~b4	SYNCMD[2:0]	Synchronization mode selection	SYNCMD[2] 0: single trigger 1: Loop trigger SYNCMD[1] 0: delayed trigger mode 1: Parallel trigger mode SYNCMD[0] 0: ADC1 and ADC2 work synchronously, ADC3 works independently 1: ADC1, ADC2 and ADC3 work synchronously **Note:** This register is set when SYNCEN is "0". When using a single trigger, please set the ADC to be synchronized to Sequence A single scan, or Sequence A continuous scan mode. When using loop trigger mode, set the ADC to Sequence A single-sweep mode.	RW
b3~b1	Res	Reserved Bit	Read at 0, write at 0	RW
b0	SYNCEN	Synchronous Mode License	0: Synchronous mode is invalid 1: Synchronous mode is valid **Note:** Synchronous mode only supports sequence A. Before SYNCEN writes 1, please turn off sequence B of several ADCs involved in synchronization (ADC_CR0.MS[1]=0), and select the same number of channels for sequence A, and set the same channel sampling time ADC_SSTRx. In order to avoid the inconsistent scanning time t_{SCAN} of each ADC, causing subsequent synchronization failures. When the software writes 0 to ADC1_STR.STRT to force the conversion to stop, SYNCEN is automatically cleared to 0.	RW

Note:

This register is only available in the master ADC (i.e. ADC1), and is not available in other ADC units.

16.5.16 ADC Data Register (ADC_DRx)

The ADC_DR register is a read-only register used to store the ADC conversion data of each channel. The reset value is 0x0000.

The conversion result data is stored differently depending on the acquisition mode, data alignment, and conversion resolution.

Oversampling Mode

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
AD[15:0]															

Right alignment of data - 12 bit resolution

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
0	0	0	0	AD[11:0]											

Right alignment of data - 10 bit resolution

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
0	0	0	0	0	0	AD[9:0]									

Right alignment of data - 8-bit resolution

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
0	0	0	0	0	0	0	0	AD[7:0]							

Data Left Alignment - 12-bit Resolution

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
AD[11:0]												0	0	0	0

Data Left Alignment - 10-bit Resolution

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
AD[9:0]										0	0	0	0	0	0

Data Left Alignment - 8-bit Resolution

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
AD[7:0]								0	0	0	0	0	0	0	0

Note:

The x in ADC_DRx represents the channel, x=0~15 in ADC1, x=0~15 in ADC2, and x=0~11 in ADC3.

16.5.17 Analog Watchdog Control Register (ADC_AWDCR)

Reset value: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8
Res						AWDCM[1:0]	

b7	b6	b5	b4	b3	b2	b1	b0
Res	AWD1MD	AWD1IEN	AWD1EN	Res	AWD0MD	AWD0IEN	AWD0EN

Bit/Bitfield	Marking	Bit Name	Function	RW
b15~b10	Res	Reserved Bit	Read at 0, write at 0	RW
b9~b8	AWDCM[1:0]	Watchdog Window Combination Selection	00: Invalid window combination, ADC _ CMP1 output window 1 independent comparison result 01: The combination of windows is valid, the logic of the ADC _ CMP1 output window 0 comparing to window 1 or 10: The combination of windows is valid, and the logic of the comparison between ADC _ CMP1 output window 0 and window 1 is 11: The combination of windows is valid, and the logical OR of the output window 0 of the ADC _ CMP1 is compared with window 1. **Note:** Both Window 0 and Window 1 are required to use the Window Combination feature, i.e. AWD0EN Write 1 and AWD1EN Write 1. If Window 0 is not the same as Window 1, make sure Window 1 is selected after Window 0 is selected during scan conversion. The ADC _ CMP1interrupt or event is output at the end of the channel conversion selected for Window 1.	RW
b7	Res	Reserved Bit	Read at 0, write at 0	RW
b6	AWD1MD	Comparison mode of watchdog window 1	Window 1 protected area selection 0: Conversion result < AWD1DR0, or conversion result > AWD1DR1 1: Conversion result = AWD1DR0, and conversion result = AWD1DR1.	RW
b5	AWD1IEN	Watchdog window 1interrupt enable	0: Watchdog Window 1 comparison interruptADC _ CMP1 invalid 1: Watchdog Windows1 Compare the validity of interruptADC _ CMP1	RW
b4	AWD1EN	Watchdog Window 1 Compare Enabled	0: Watchdog Window 1 comparison function invalid 1: Watchdog Window 1 invaild	RW
b3	Res	Reserved Bit	Read at 0, write at 0	RW
b2	AWD0MD	Comparison mode of watchdog window 0	Window 0 protected area selection 0: Conversion result < AWD0DR0, or conversion result > AWD0DR1 1: Conversion result = AWD0DR0, and conversion result = AWD0DR1.	RW
b1	AWD0IEN	Watchdog Window 0 interrupt enable	0: Watchdog Window 0 Compare interruptADC _ CMP0 Invalid 1: Watchdog Windows0 Compared with interruptADC _ CMP0	RW
b0	AWD0EN	Watchdog Window 0 Compare Enabled	0: Watchdog Window 0 comparison function invalid 1: Watchdog Window 0 Compare Function Effective	RW

16.5.18 Analogwatchdog Status Register (AD _ AWDSR)

Reset Value: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
Res			AWDCMF	Res		AWD1F	AWD0F

Bit/Bitfield	Marking	Bit Name	Function	RW
b7~b5	Res	Reserved Bit	Read at 0, write at 0	RW
b4	AWDCMF	Watchdog window combination comparison status bit	When the watchdog window combination comparison function is effective (that is, AWDCM [1: 0] = 0b01/0b10/0b11), when window 0 and window 1 compare results meet the combination condition, set 1. Invalid write to this register bit.	R
b3~b2	Res	Reserved Bit	Read at 0, write at 0	RW
b1	AWD1F	Watchdog window 1 comparative status position	At the end of the selected channel conversion of Window 1, set the conversion result to 1 when the comparison condition is met. Invalid write to this register bit.	R
b0	AWD0F	Watchdog window 0 comparison status bit	At the end of the channel conversion selected for Window 0, set the conversion result to 1 when the comparison condition is met. Invalid write to this register bit.	R

16.5.19 Analogwatchdog Status Reset Register (ADC _ AWDSCLRR)

Reset Value: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
Res			CLRAWDCMF	Res		CLRAWD1F	CLRAWD0F

Bit/Bitfield	Marking	Bit Name	Function	RW
b7~b5	Res	Reserved Bit	Read at 0, write at 0	RW
b4	CLRAWDCMF	Watchdog Window Combination Comparison State Position Reset	Write 0 has no effect. Write 1 resets the AWDCMF status bit. Read always 0	RW
b3~b2	Res	Reserved Bit	Read at 0, write at 0	RW
b1	CLRAWD1F	Watchdog Window 1 Comparative State Reset	Write 0 has no effect. Write 1 resets the AWD1F status bit. Read always 0	RW
b0	CLRAWD0F	Watchdog Window 0 Comparative State Reset	Write 0 has no effect. Write 1 resets the AWD1F status bit. Read always 0	RW

16.5.20 Analog Watchdog Threshold Registers (ADC_AWD0DR0, ADC_AWD0DR1, ADC_AWD1DR0, ADC_AWD1DR1)

Reset value: ADC_AWD0DR0=0x0000, ADC_AWD0DR1=0xFFFF

ADC_AWD1DR0=0x0000, ADC_AWD1DR1=0xFFFF

b15	b14	b13	b12	b11	b10	b9	b8
AWDDR[15:8]							
b7	b6	b5	b4	b3	b2	b1	b0
AWDDR[7:0]							

Bit/Bitfield	Marking	Bit Name	Function	RW
b15~b0	AWDDR[15:0]	Comparative data	Comparative data	RW

AWD0DR0 sets the Window 0 low threshold and AWD0DR1 sets the Window 0 high threshold.

AWD1DR0 sets the Window 1 low threshold and AWD1DR1 sets the Window 1 high threshold.

AWD0DR0, AWD0DR1, AWD1DR0, AWD1DR1 may differ in resolution (12-bit, 10-bit or 8-bit) according to alignment (right or left alignment of data).

- Right alignment of data - 12 bit resolution low 12 bits [11: 0] available
- Right alignment of data - 10 bit resolution low 10 bits [9: 0] available
- Right Alignment of Data - 8-bit Resolution Low 8-bit [7: 0] Available
- Data left alignment - 12 bit resolution high 12 bits [15: 4] available
- Data left alignment - 10 bit resolution high 10 bits [15: 6] available
- Data left alignment - 8 bit resolution high 8 bits available [15: 8]

16.5.21 Analog Watchdog Comparison Channel Selection Register (ADC _ AWD0 CHSR, ADC _ AWD1 CHSR)

Reset Value: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
Res			AWDCH[4]	AWDCH[3]	AWDCH[2]	AWDCH[1]	AWDCH[0]

Bit/Bitfield	Marking	Bit Name	Function	RW
b7~b5	Res	Reserved Bit	Read at 0, write at 0	RW
b4~b0	AWDCH[4:0]	Comparison channel selection of watchdogs	ADC _ AWD0 CHSR selection window 0 and ADC _ AWD1 CHSR selection window 1. 0x00: CH0 0x01: CH1 And so on **Note:** Please do not set it to a non-existent channel.	RW

16.6Precautions

16.6.1 Considerations when Reading Data Register

Please access the ADC data register ADC_DR in half-word units. Do not access the data register in bytes.

16.6.2 Scan Finish Interrupt Handling Considerations

If the CPU does not read the first converted data in time, the second converted data will overwrite the first converted data.

16.6.3 Precautions for Module Stop and Low Power Consumption Setting

By setting the register PWC_FCG, the ADC module can be stopped to reduce Power Consumption. The initial state of ADC is stop state. When the ADC module needs to work, please first set the corresponding bit of the PWC_FCG register to cancel the stop, and wait for 1μs before starting the ADC conversion.

Before setting the module to stop, please make sure that the ADC is in conversion stop, that is, the ADC_STR.STRT bit is 0.

Before setting the system to enter the Stop Mode (STOP), please set the ADC to the module Stop Mode first.

Please refer to the [Action Mode and Low Power Mode] section for details.

16.6.4 ADC Conversion Analog Channel Input Pin Setting

When the chip pin is set as ADC analog channel input, please disable the digital function of the corresponding pin first (PCRxy.DDIS). Refer to [General-Purpose I/Os (GPIO)] chapter.

16.6.5 Noise Control

In order to prevent abnormal voltages such as surges from damaging analog input pins, it is recommended to use the protection circuit shown in the electrical characteristics chapter of the Data Manual.

17 Digital-to-Analog Converter (DAC)

17.1 Overview

This MCU is equipped with two 12-bit digital-to-analog converter units DAC. The DAC1 unit contains two DAC conversion channels, which can be converted independently or synchronously. The DAC2 unit contains one DAC conversion channel. The analog voltage output range can be set in two levels. Each conversion channel is equipped with an output amplifier, which can directly drive external loads without external op amps.

17.2 Main Features

- Three DAC conversion channels
- 12-bit conversion data can be configured as left-aligned or right-aligned format
- The two conversion channels of the same DAC can be converted synchronously
- Output can be used for the negative terminal voltage of the voltage comparator (CMP)
- The output is equipped with an amplifier function and can directly drive external loads
- ADC conversion priority mode can reduce interference during ADC conversion

17.3 Functional Description

17.3.1 DAC Functional Block Diagram

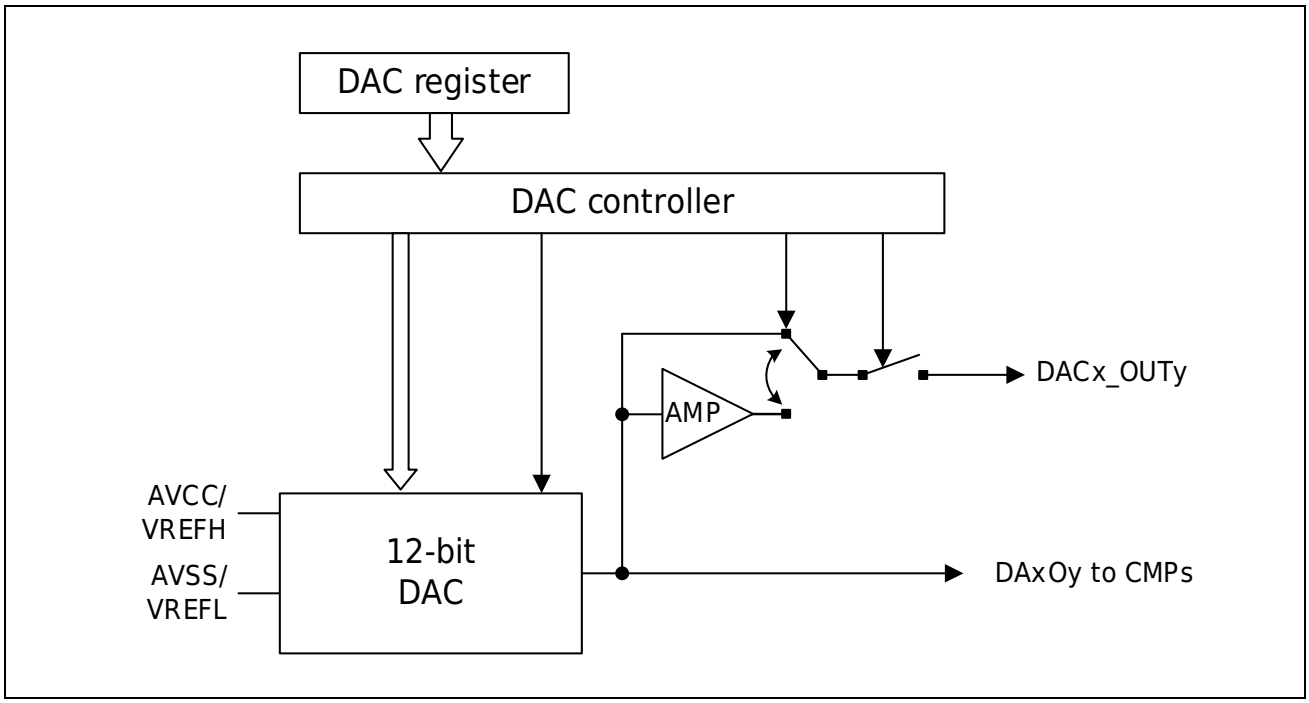


Figure 17-1 DAC Conversion Channel Block Diagram

Table 17-1 DAC Pinout

Pin Name	Input/Output	Function
AVCC/ VREFH	Input	Analog power supply/reference voltage
AVSS/ VREFL	Input	Analog ground/reference ground
DACx_OUTy	Output	DAC converts analog output (x=1~2, y=1~2)

17.3.2 DAC Conversion

Each conversion channel can perform DAC conversion independently. When DACx_DACR.DAyE (x=1~2, y=1, 2) is set to 1, the DAC conversion of the corresponding channel starts and the conversion result is output from the DACx_OUTy port. If only the conversion result is used as the negative input voltage of the voltage comparator CMP, the DACx_OUTy output can be turned off through the DACx_DAOCR register.

The following is an example of using DAC_ch1 to perform single-channel conversion. The action sequence is shown in Figure 17-2.

1) Set the DAC conversion data (DADR1) and data format (DACR.DPSEL).

2) Set DAC1_DACR.DA1E to 1 and DAC conversion starts. After the conversion time tDCONV, the conversion result is output from port DAC1_OUT1 and remains until the value of DAC1_DADR1 is rewritten or DAC1_DACR.DA1E is set to 0. The output analog voltage value can be calculated by the following formula:

   When DAC1_DACR.DA1E is 1, rewriting the value of DAC1_DADR1 will trigger a new DAC conversion. Similarly, after the conversion time tDCONV, a new conversion result is output from port DAC1_OUT1.

3) Set the DAC1_DACR.DA1E bit to 0 to turn off DAC1_ch1 and make DAC1_OUT1 output high impedance.

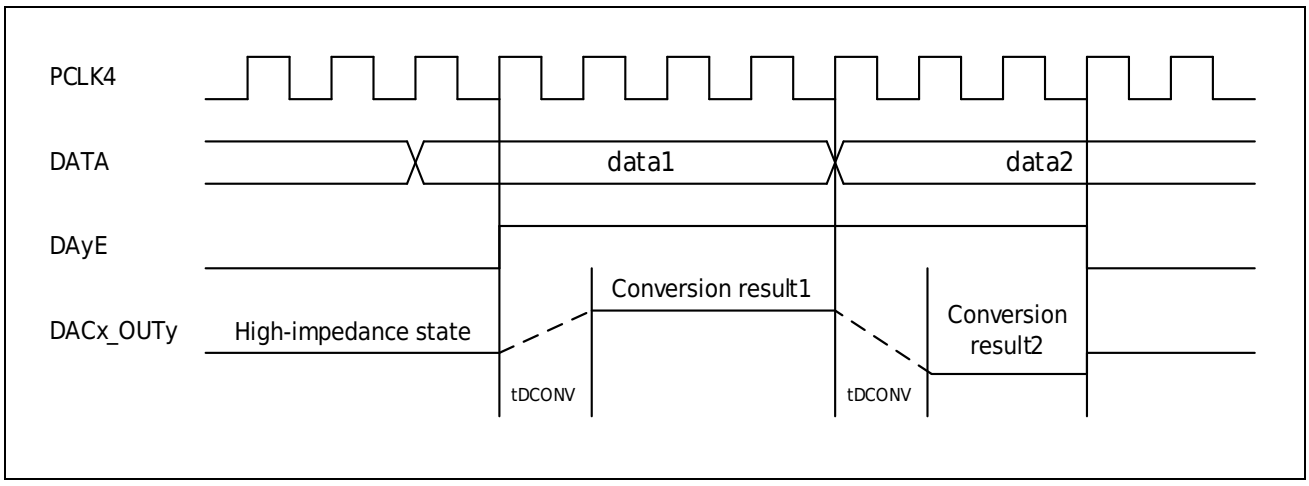


Figure 17-2 DAC Conversion Diagram

17.3.3 Synchronous conversion

Synchronous conversion of two conversion channels of the same DAC unit can be achieved by synchronously updating the conversion data.

Below is an example of using DAC1 to perform two-channel simultaneous conversions.

1. Set the DAC conversion data (DAC1_DADR1, DAC1_DADR2) and data format (DAC1_DACR.DPSEL).
2. Set DAC1_DACR.DAE to 1, and the DAC conversion of the two channels starts synchronously. After the conversion time t_{DCONV}, the conversion results are output from ports DAC1_OUT1 and DAC1_OUT2 respectively and are kept until the value of DADR is rewritten or DAC1_DACR.DAE is set to 0.
3. When DAC1_DACR.DAE is 1, rewriting the values of DAC1_DADR1 and DAC1_DADR2 at the same time will synchronously trigger the two channels to start a new DAC conversion. Likewise, after the conversion time tDCONV, new conversion results are output from DAC1_OUT1 and DAC1_OUT2.
4. Set DAC1_DACR.DAE to 0, the two channels are turned off synchronously, and DAC1_OUT1 and DAC1_OUT output high impedance.

17.3.4 ADC conversion priority mode

When the DAC starts a DAC conversion, there may be a brief surge in current on the analog power supply, which can interfere with the ongoing ADC conversion. The ADC conversion priority mode can effectively avoid this interference by changing the update timing of the DAC conversion data.

Set DACx_DAADPCR.ADPEN to 1 and DAC enters ADC conversion priority mode. At this time, if the value of DACx_DADR is rewritten during ADC conversion, the DAC conversion will not start immediately, but will start after the ADC conversion is completed. That is to say, the longest waiting time from rewriting DADR to actually starting DAC conversion is the time of one ADC conversion. Therefore, the DADR value does not match the analog output value during this period. However, if the ADC is in the stopped state (ADC_STR.STRT is 0) when DACx_DADR is rewritten, the DAC conversion will start after 2 ADCLK cycles. DACx_DAADPCR.ADCSLn (n=0~2) is used to select the ADC channel with priority for ADC conversion.

The following takes DAC1_ch1 as an example to explain the steps for setting the ADC conversion priority mode. The action sequence is shown in Figure 17-3.

1. Confirm that ADC is in stop state and set DAC1_DACR.ADPSL;
2. Confirm that ADC is in stop state and set DAC1_DACR.DA1E to 1;
3. Write the converted data to DAC1_DADR1;

4. When the conversion data A is written to DAC1_DADR1, ADC_ADCSR.ADST is 0, the ADC is in the stopped state, and the DAC conversion starts after 2 ADCLK cycles;

5. When conversion data B is written to DAC1_DADR1, ADC_ADCSR.ADST is 1, and the ADC is performing ADC conversion. The DAC conversion will not start until the ADC conversion is completed. Before the DAC conversion starts, the conversion data C has been written into DAC1_DADR1, so what is finally converted is the conversion data C, not the conversion data B.

To avoid conversion data loss, please check the DAC conversion status register (DACx_DAADPCR.DAySF) first and confirm that the current data conversion is completed before rewriting the DADR value.

Note:

x=1~2, y=1~2

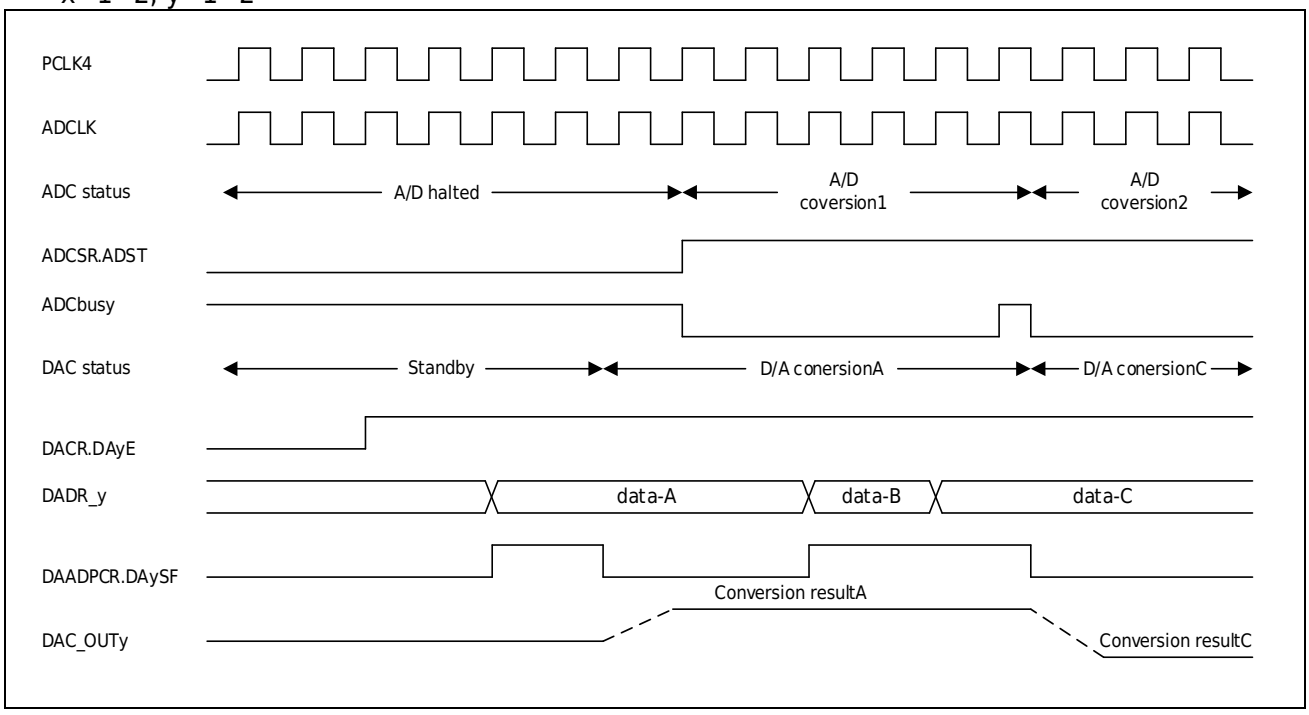


Figure 17-3 ADC conversion priority mode operation diagram

17.3.5 Data Transfer Mode

By setting DACR2.LDMDy = 1, the trigger event output by HRPWM can be set to trigger the data transfer from DADR to ACTIVE register (real-time data converted by DAC).

When the channel is enabled:

- If there is no trigger event of HRPWM output, data transmission will not be triggered even if the data of DADR is rewritten, and D/A conversion will not occur;
- When a trigger event of HRPWM output occurs, the data of DADR is transferred to the ACTIVE register and D/A conversion is performed.

The ACTIVE value of the DAC data conversion can be read out through the register DADACTRy.

17.4Register Description

Table 17-2 DAC Base Address

Name	Base Address	Description
DAC1	0x4004 1000	DAC1 base address
DAC2	0x4004 1400	DAC2 base address

Table 17-3 DAC Register List

Register	Description	Offset Address	Bit Width	Reset Value
DADR1	DAC data register 1	0x00	16	0x0000
DADR2	DAC data register 2	0x02	16	0x0000
DACR	DAC control register	0x04	16	0x0000
DAADPCR	DAC ADC conversion priority control register	0x06	16	0x0000
DAOCR	DAC Analog Output Control Register	0x1C	16	0x0000
DACR2	DAC control register 2	0x20	16	0x0000
DADACTR1	DAC Data ACTIVE Value Register 1	0x2c	16	0x0000
DADACTR2	DAC Data ACTIVE Value Register 2	0x2e	16	0x0000

17.4.1 DAC Data Register (DADRy, y=1~2)

When DACR.DPSEL=0 (data right-aligned)

Reset value: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res				DADR[11:0]											

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b15~b12	Res	Reserved Bit	Read 0 when reading, write 0 when writing	RW
b11~b0	DADR[11:0]	transformation Data	Transformation data setting value	RW

When DACR.DPSEL=1 (data left-aligned)

Reset value: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
DADR[11:0]												Res			

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b15~b4	DADR[11:0]	transformation Data	Transformation data setting value	RW
b3~b0	Res	Reserved Bit	Read 0 when reading, write 0 when writing	RW

DACx_DADR is used to store DAC conversion data, one for each channel. As long as the DAC conversion is enabled, the value in DACx_DADR is converted to an analog voltage and output from the analog port. 32-bit operation on the DACx_DADR of the same unit can achieve dual-channel synchronous conversion.

17.4.2 DAC Control Register (DACR)

Reset value: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res					DAAMP2	DAAMP1	DPSEL	Res					DA2E	DA1E	DAE

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b15~b11	Res	Reserved Bit	Read at “0”, write at “0”	RW
b10	DAAMP2	Channel 2 output amplifier enable	0: Disable channel 2 output amplifier 1: Enable channel 2 output amplifier	RW
b9	DAAMP1	Channel 1 output amplifier enable	0: Disable channel 1 output amplifier 1: Enable channel 1 output amplifier	RW
b8	DPSEL	Data register format selection	0: Right-aligned format 1: Left-aligned format	RW
b7~b3	Res	Reserved Bit	Read at “0”, write at “0”	RW
b2	DA2E	Channel 2 enable	This bit setting valid when DAE=0 0: Channel 2 disabled 1: Channel 2 enabled	RW
b1	DA1E	Channel 1 enable	This bit setting valid when DAE=0 0: Channel 1 disabled 1: Channel 1 enabled	RW
b0	DAE	Channel always enabled	0: Channel 1, 2 disabled 1: Channel 1, 2 enabled	RW

Note:

1. *DACR is used to control DAC conversion and DAC output. Please see Table 17-4 for details.*
2. *When the DA1E bit is 0 and the DAE bit is also 0, channel 1 stops DAC conversion. At this time, no matter what the value of DAAMP1 is, the analog output of channel 1 disable and the port is in a high-impedance state. When the DA2E bit is 0 and the DAE bit is also 0, channel 2 stops DAC conversion. At this time, no matter what the value of DAAMP2 is, the analog output of channel 2 disable and the port is in a high-impedance state.*
3. *If ADC conversion priority mode is selected, set the DA2E, DA1E, and DAE bits when the ADC is stopped (ADC_STR.STRT = 0). At the same time, in order to prevent the ADC from being accidentally started, please set the ADC trigger selection to software trigger.*

17.4.3 DAC Analog Output Control Register (DAOCR)

Reset value: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
DAO DIS2	DAO DIS1	Res													

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b15	DAODIS2	DACx_OUT2 Port Output Control	1: Disable DACx_OUT2 port from outputting analog voltage 0: Allow DACx_OUT2 port to output analog voltage	RW
b14	DAODIS1	DACx_OUT1 port output control	1: Disable DACx_OUT1 port from outputting analog voltage 0: Allow DACx_OUT1 port to output analog voltage	RW
b13~b0	Res	Reserved Bit	Read at “0”, write at “0”	RW

Table 17-4 DAC Conversion and Analog Output Control

DAE	DAyE	DAAMP	DAODIS	DAC Conversion	Output Amplifier	DAC_OUT Output	DAO Output
0	0	0/1	0/1	Stop	Stop	High Impedance	High Impedance
	1	0	0	Start up	Stop	Normal Output	Normal Output
			1	Start	Stop	High Impedance	Normal Output
		1	0	Start	Start	Amplified Output	Normal Output
			1	Start	Start	High Impedance	Normal Output
1	0	0	0	Start	Stop	Normal Output	Normal Output
			1	Start	Stop	High Impedance	Normal Output
		1	0	Start	Start	Amplified Output	Normal Output
			1	Start	Start	High Impedance	Normal Output
	1	0	0	Start	Stop	Normal Output	Normal Output
			1	Start	Stop	High Impedance	Normal Output
		1	0	Start	Start	Amplified Output	Normal Output
			1	Start	Start	High Impedance	Normal Output

17.4.4 DAC ADC Conversion Priority Control Register (DAADPCR)

Reset value: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
ADPEN	Res					DA2SF	DA1SF	Res					ADCSL3	ADCSL2	ADCSL1

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b15	ADPEN	ADC conversion priority mode selection	0: ADC conversion priority mode invalid 1: ADC conversion priority mode valid	RW
b14~b10	Res	Reserved Bit	Read at “0”, write at “0”	RW
b9	DA2SF	Channel 2 data update status	This flag changes only in ADC conversion priority mode 0: Channel 2 data update end 1: Channel 2 data is updating	RW
b8	DA1SF	Channel 1 data update status	This flag changes only in ADC conversion priority mode 0: Channel 1 data update end 1: Channel 1 data is updating	RW
b7~b3	Res	Reserved Bit	Read at “0”, write at “0”	RW
b2	ADCSL3	ADC_3 priority selection	0: ADC_3 is not selected as the ADC conversion priority object 1: Select ADC_3 as the ADC conversion priority object	RW
b1	ADCSL2	ADC_2 priority selection	0: ADC_2 is not selected as the ADC conversion priority object 1: Select ADC_2 as the ADC conversion priority object	RW
b0	ADCSL1	ADC_1 priority selection	0: ADC_1 is not selected as the ADC conversion priority object 1: Select ADC_1 as the ADC conversion priority object	RW

Note:

1. *DAADPCR is used to control the ADC conversion priority mode.*
2. *When ADPEN is set to 1, the ADC conversion priority mode is enabled. Please select the ADC unit as the priority object while setting ADPEN.*
3. *DA1SF is the DA conversion status flag, which reflects whether the data in the current channel data register has been converted.*

17.4.5 DAC Control Register 2 (DACR2)

Reset value: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
LDMD2	Res							LDMD1	Res						

Bit	Marking	Bit Name	Function	Read and Write
b15	LDMD2	Channel 2 data transfer mode selection	0: Real-time data transmission 1: The trigger event of HRPWM output triggers data transmission	RW
b14~b8	Res	Reserved Bit	Read at “0”, write at “0”	RW
b7	LDMD1	Channel 1 data transfer mode selection	0: Real-time data transmission 1: The trigger event of HRPWM output triggers data transmission	RW
b6~b0	Res	Reserved Bit	Read at “0”, write at “0”	RW

17.4.6 DAC Data ACTIVE Value Register (DADACTRy, y=1~2)

Reset value: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res				DATAACT[11:0]											

Bit	Marking	Bit Name	Function	Read and Write
b15~b12	Res	Reserved Bit	Read at “0”, write at “0”	RW
b11~b0	DATAACT	Channel y transformation data ACTIVE value	Transform the ACTIVE value of the data	R

17.5Precautions

17.5.1 Setting Module Stop Function

The module stop control register can be used to set the DAC to be enabled or stopped. After the system is reset, the DAC is stopped and the registers can be accessed only after the module is released from the stopped state. For details, see [Action Mode and Low Power Mode].

17.5.2 DAC Operation when the Module is Stopped

If the DAC is in the module stop state during DAC conversion, the analog output will be held and the current flowing through the analog power supply is the same as when the DAC is converting. If you need to further reduce the Power Consumption of the module in the stopped state, set the DAyE and DAE bits of DACx_DACR to all 0.

17.5.3 DAC Operation in Stop Mode

If the system enters Stop Mode during DAC conversion, the analog output will be held and the current flowing through the analog power supply is the same as during DAC conversion. If you need to further reduce the Power Consumption in Stop Mode, set the DAyE and DAE bits of DACx_DACR to all 0s.

17.5.4 DAC Operation in Power-Down Mode

If the system enters Power Down Mode while the DAC is converting, the analog output will be set to a high-impedance state.

17.5.5 Notes on Using Output Amplifiers

Please use the following initialization procedure when using the output amplifier.

1. Write all 0s to DACx_DADRy;
2. Set DACx_DACR.DAAMPy to 1;
3. Set DACx_DACR.DAE or DAyE to 1;
4. Wait 3μs before writing the conversion data to DACx_DADRy.

Turning off the DAC can stop the amplifier from working. To use the amplifier again, repeat steps 1 to 4.

18 Advanced Control Timer (Timer6)

18.1 Overview

Advanced Control Timer 6 (Timer6) is a high-performance timer with a 16-bit count width, which can provide rich and flexible combinations and various interrupts, events, and PWM outputs in various complex application scenarios. The timer supports two waveform modes, sawtooth wave and triangle wave, and can generate various PWM waveforms (single-side aligned independent PWM, bilateral symmetrical independent PWM, bilateral symmetrical complementary PWM, bilateral asymmetrical PWM, etc.); software synchronization and hardware synchronization (synchronous start, stop, clear, refresh, etc.) can be achieved between units; each reference value register supports cache function (single-level cache and double-level cache); supports pulse width measurement and period measurement; supports 2-phase orthogonal encoding counting and 3-phase orthogonal encoding counting; supports EMB control. This series of products is equipped with 4 units of Timer6.

18.2 Main Features

The basic functions and characteristics of Timer6 are shown in Table 18-1.

Table 18-1 Basic Functions and Features of Timer6

Waveform mode	Sawtooth wave (count up, count down)
	Triangle wave (up and down counting)
Basic functions	Capture input
	Software synchronization
	Hardware synchronization
	Cache function
	Pulse width measurement
	Periodic measurement
	Orthogonal coding count
	Universal PWM output
	EMB control
Interrupt output	Count comparison match interrupt
	Count cycle match interrupt
	Dead time error interrupt
Event output	Count compare match events
	Count period match event

18.3Functional Description

18.3.1 Timer6 Block Diagram

The basic block diagram of Timer6 is shown in Figure 18-1. As shown in the block diagram, "< t >" represents the unit number, that is, "< t >" is 1~4. This section refers to "< t >" when referring to "< t >" later.

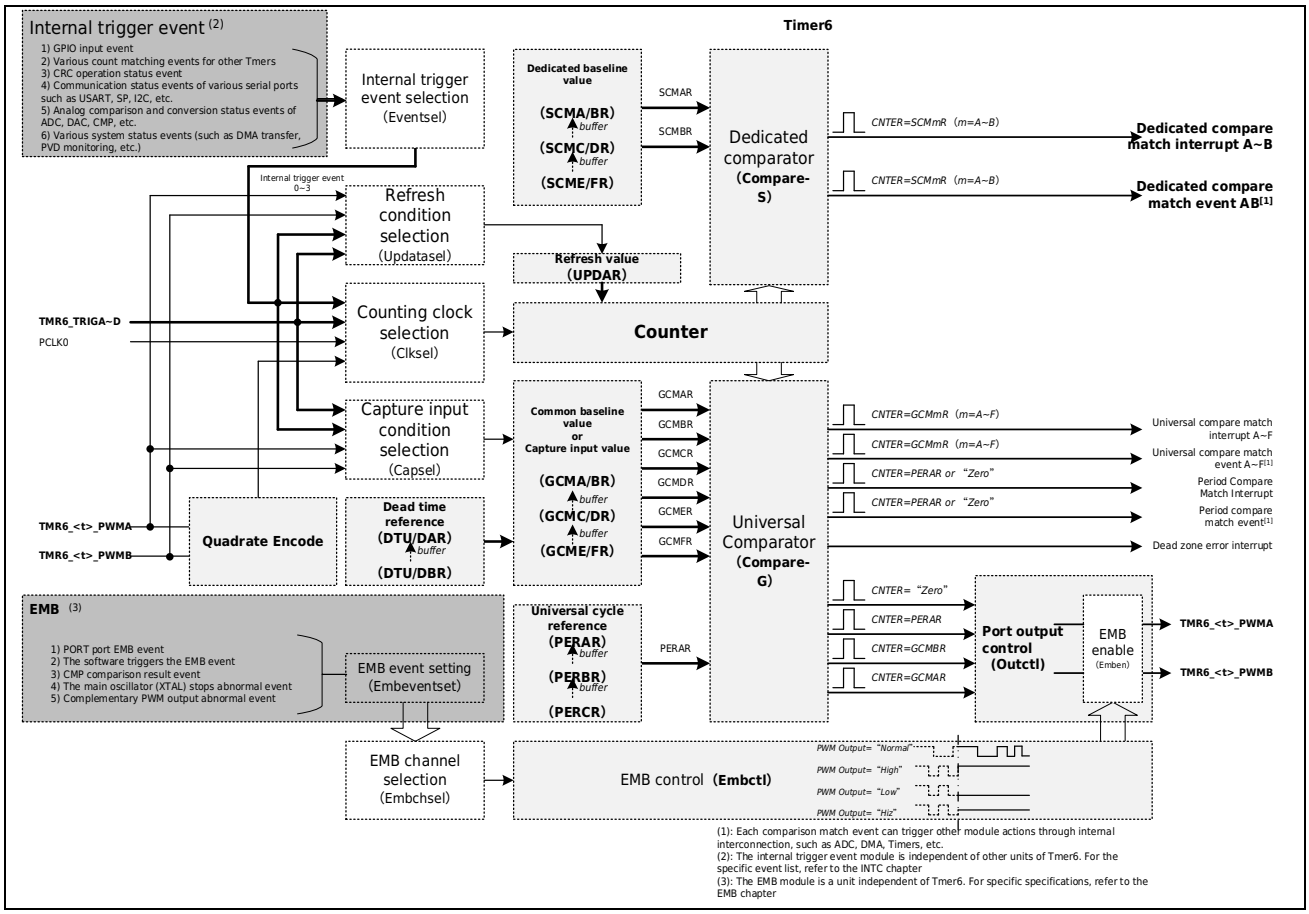


Figure 18-1 Timer6 Basic Block Diagram

Table 18-2 shows the list of input and output ports of Timer6.

Table 18-2 Timer6 Port List

Port name	Direction	Function
TMR6_<t>_PWMA	Input output	1) Quadrature encoding count clock input port or capture input port or comparison output port 2) Hardware start, stop, clear, refresh condition input port
TMR6_<t>_PWMB		
TMR6_TRIGA	Input	1) Hardware count clock input port or capture input port 2) Hardware start, stop, clear, refresh condition input port
TMR6_TRIGB		
TMR6_TRIGC		
TMR6_TRIGD		

18.3.2 Waveform Mode

Timer6 has two basic counting wave modes: Serrated wave mode and triangular wave mode. The basic waveforms of the two waveform modes are shown in Figure 18-2 and Figure 18-3.

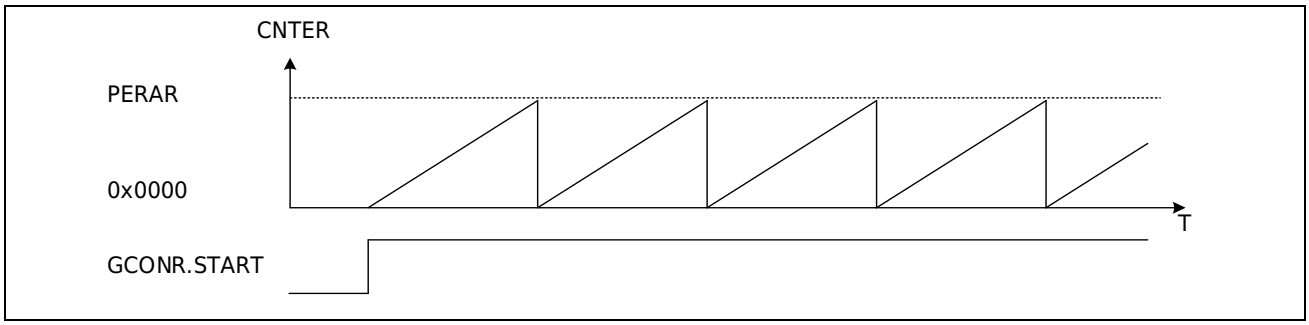


Figure 18-2 Sawtooth Waveform (Counting Up)

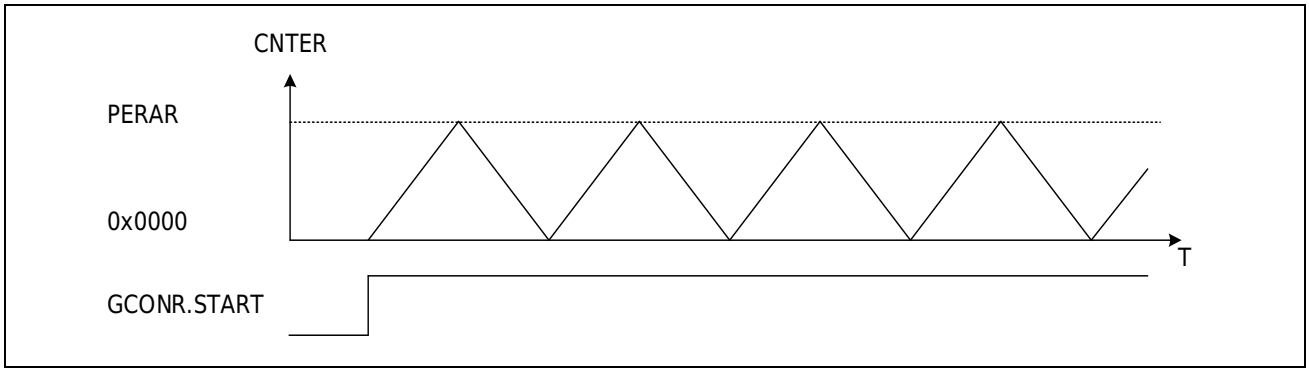


Figure 18-3 Triangle Waveform

18.3.3 Timer Selection

The counting clock of Timer6 can have the following options:

a) 1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024 frequency division of PCLK1 (GCONR.CKDIV[3:0] setting)

b) Internal trigger event input 0~3 (set by HCUPR.HCUP[11:8] or HCDOR.HCDO[11:8])

c) TMR6_TRIGA~D port input (HCUPR.HCUP[23:16] or HCDOR.HCDO[23:16] setting)

d) The quadrature encoding relationship input of TMR6_<t>_PWMA and TMR6_<t>_PWMB (set by HCUPR.HCUP[7:0] or HCDOR.HCDO[7:0])

The counting timer is a software counting mode, and the counting timer is b, c, and d hardware counting mode. As can be refer ton from the above description, the b, c and d clocks are independent of each other, can be set to be valid or invalid respectively, and a clock is automatically invalid when b, c and d clocks are selected.

18.3.4 Counting Direction

The timer counting direction of Timer6 can be changed by software. The method of changing the counting direction is slightly different in different waveform modes.

18.3.4.1 Sawtooth Counting Direction

In sawtooth wave mode, the counting direction can be set while the timer is counting or stopped.

When counting up, set GCONR.DIR=0 (count down), then the timer will change to down count mode after counting to overflow; when counting down, set GCONR.DIR=1 (count up)), the timer will change to up counting mode after counting to underflow.

The GCONR.DIR bit is set when counting is stopped, and the setting of GCONR.DIR is reflected in the counting until it overflows or underflows after the counting starts.

18.3.4.2 Triangular Wave Counting Direction

In the triangular wave mode, the set counting direction is invalid, and the counting direction will be changed automatically when counting to the counting peak point or counting valley point.

18.3.5 Comparison Output

Timer6 of each unit has 2 comparison output ports (TMR6_<t>_PWMA, TMR6_<t>_PWMB), which can output a specified level when the count value matches the comparison reference value. The GCMAR and GCMBR registers correspond to the count comparison reference values of TMR6_<t>_PWMA, TMR6_<t>_PWMB respectively. When the timer count value is equal to GCMA<B>R, the TMR6_<t>_PWMA or TMR6_<t>_PWMB port outputs the specified level.

The counting start level, counting stop level, counting overflow level, counting compare match level, etc. of the TMR6_<t>_PWMA port can be set by the PCNAR.STACA, PCNAR.STPCA, PCNAR.OVFCA, PCNAR.UDFCA, PCNAR.CMA<B>U<D>A, and PCNAR.FORCA bits of the port control register (PCNAR).

The counting start level, counting stop level, counting overflow level, counting compare match level, etc. of the TMR6_<t>_PWMB port can be set by the (PCNBR) PCNBR.STACB, PCNBR.STPCB, PCNBR.OVFCB, PCNBR.UDFCB, PCNBR.CMA<B>U<D>B, PCNBR.FORCB bit settings. Figure 18-4 is an example of the operation of the comparison output.

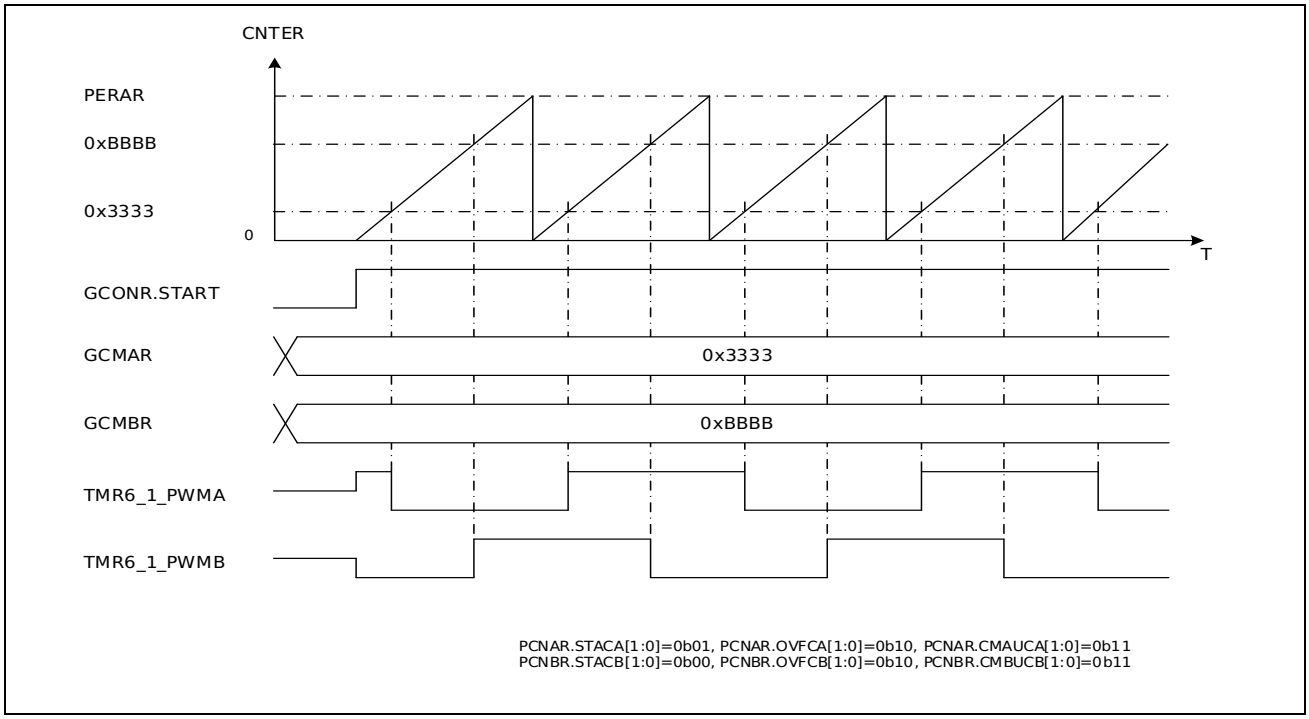


Figure 18-4 Compare Output Action

18.3.6 Capture Input

Each unit has a capture input function, with 2 sets of capture input registers (GCMAR, GCMBR), used to save the captured count value. Setting the CAPMDA<B> bit in the port control register (PCNA<B>R) to 1 enables the capture input function. When the corresponding capture input condition is set and the condition is valid, the current count value is saved to the corresponding capture register (GCMAR, GCMBR).

The conditions for each capture input of each unit can be internal trigger event input 0~3, port input of TMR6_TRIGA~D, port input of TMR6_<t>_PWMA or TMR6_<t>_PWMB, etc. The specific condition selection can be set through the hardware capture event selection register (HCPAR, HCPBR). Figure 18-5 is an example of an action for capturing input.

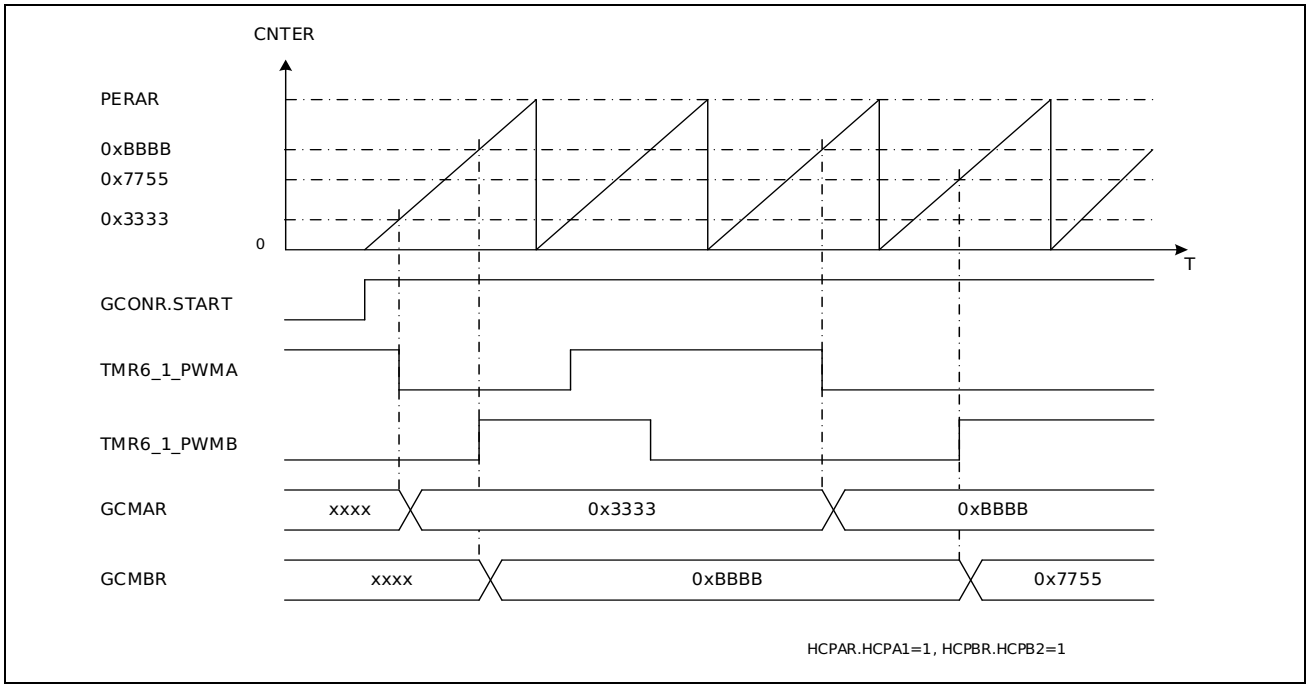


Figure 18-5 Capturing Input Actions

18.3.7 Counter Refresh

Timer6 has a universal refresh value register (UPDAR) that can refresh the count value of the universal count value register (CNTER) in real time when counting is stopped or in progress.

The refresh condition of the count value is set by the corresponding bit of the hardware refresh event selection register (HUPDR) or the software synchronous refresh control register (SUPDR). When the set refresh event valid, the value of the universal count value register (CNTER) is updated to the value specified in the universal refresh value register (UPDAR). Figure 18-6 is an example of hardware refresh operation in sawtooth wave mode counting.

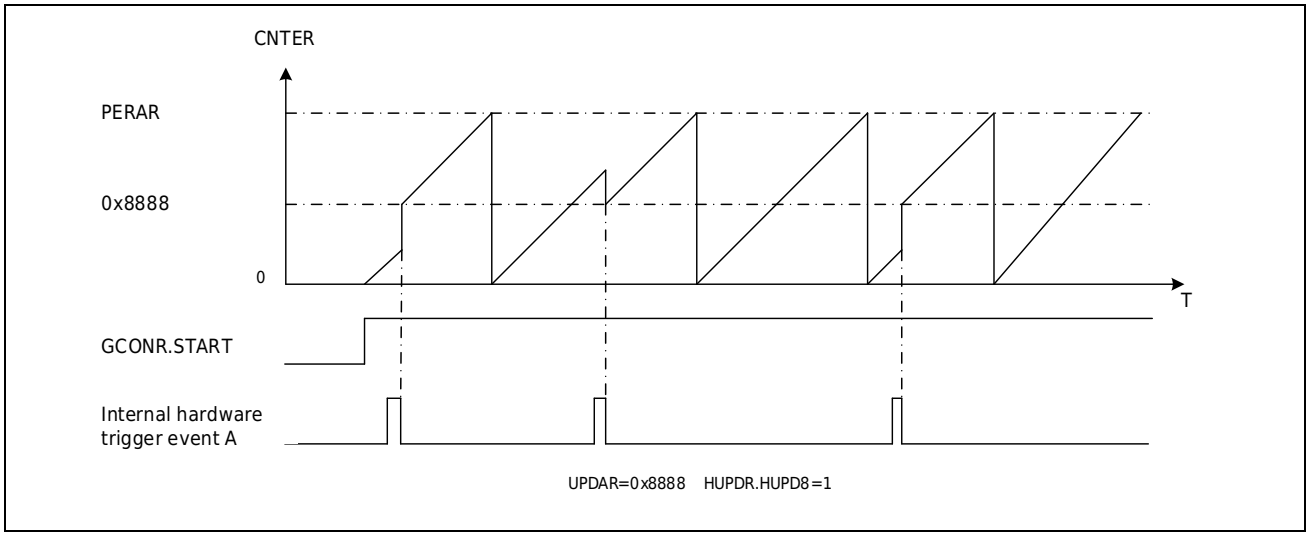


Figure 18-6 Hardware Refresh Action

18.3.8 Software Synchronization

18.3.8.1 Software Synchronization Start

Each unit can achieve synchronous start of the target unit counter (CNTER) by setting the relevant bits of the software synchronous start control register (SSTAR).

In addition, the product is also equipped with a count synchronization enable register (TMR_SYNENR) to synchronously start all the timers (including Timer0, Timer4, Timer6, TimerA) carried by the product. for application method and Register Description, please refer to the [Universal Timer (Timer0)] section. When Timer6 sets the count clock division in the Universal control register (TMR6_GCONR.CKDIV[2:0]), it does not support synchronization with other timers.

18.3.8.2 Software Co-stop

Each unit can achieve the synchronous stop of the target unit counter (CNTER) by setting the relevant bits of the software synchronous stop control register (SSTPR).

18.3.8.3 Software Synchronous Clear

Each unit can achieve synchronous clearing of the target unit counter (CNTER) by setting the relevant bits of the software synchronous clear control register (SCLRR).

18.3.8.4 Software Synchronization Refresh

Each unit can achieve synchronous refresh of the target unit counter (CNTER) by setting the relevant bits of the software synchronous refresh control register (SUPDR).

As shown in Figure 18-7, if SSTAR.SSTA1=SSTAR.SSTA2=1 is set, the software synchronization start of units 1 and 2 can be achieved, and if SSTPR.SSTP1=SSTPR.SSTP2=1 is set, the software synchronization stop of units 1 and 2 can be achieved.

Software synchronous action related registers (SSTAR, SSTPR, SCLRR, SUPDR) are a group of registers that are independent of the unit and shared between each unit. Each bit of this group of registers is only valid when writing 1, and writing 0 is invalid. When the SSTAR register is read, the timer status of each unit (counting stopped or counting) is read, and when the SSTPR, SCLRR, or SUPDR is read, zero is read.

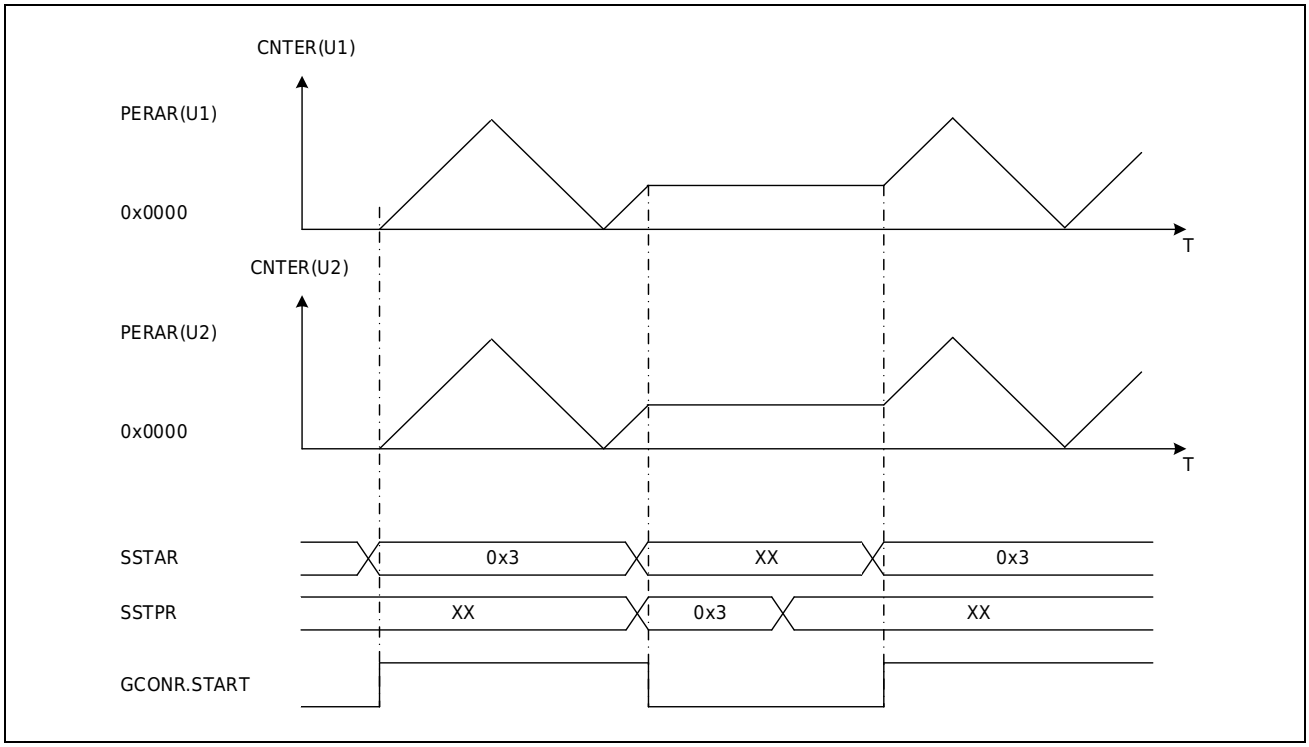


Figure 18-7 Software Synchronization Action

18.3.9 Hardware Synchronization

In addition to two independent Universal-purpose input ports (TMR6_<t>_PWMA, TMR6_<t>_PWMB), each unit also has four Universal-purpose external trigger input ports (TMR6_TRIGA, TMR6_TRIGB, TMR6_TRIGC, TMR6_TRIGD) and four internal trigger event input conditions, which can realize hardware synchronization actions between units.

The event source of the internal trigger event can be selected by setting the corresponding number in the Hardware Trigger Event Selection Register (TMR6_TRGSELx, x=0~3). For details of the register content, please refer to the [Timer6 Hardware Trigger Event Selection Register (TMR6_TRGSELx) (x=0~3)] section. For specific event correspondence, please refer to the [Interrupt controller (INTC)] section. When using the internal hardware trigger function, the function clock control register (PWC_FCG0) peripheral circuit trigger function is required to enable position 1.

18.3.9.1 Hardware Sync Start

Each unit can choose to start the timer by hardware, and the units with the same hardware start condition can realize synchronous start when the start condition is valid. The specific hardware start condition is determined by the setting of the hardware start event selection register (HSTAR).

18.3.9.2 Hardware Co-stop

Each unit can choose to stop the timer by hardware, and the units with the same hardware stop condition can realize synchronous stop when the stop condition is valid. The specific hardware stop condition is determined by the setting of the hardware stop event selection register (HSTPR).

18.3.9.3 Hardware Synchronous Clear

Each unit can choose to clear the timer by hardware, and select the units with the same hardware clearing condition to realize synchronous clearing when the clearing condition is valid. The specific hardware clear condition is determined by the setting of the hardware clear event select register (HCLRR).

18.3.9.4 Hardware Synchronous Refresh

Each unit can choose to refresh the timer in hardware. Units that select the same hardware refresh conditions can achieve synchronous refresh when the refresh conditions are valid. The specific hardware refresh conditions are determined by the setting of the hardware refresh event selection register (HUPDR).

18.3.9.5 Hardware Synchronous Capture Input

Each unit can choose to use hardware to realize the capture input function, and select the unit with the same capture input function condition to realize synchronous capture input when the capture input function condition is valid. The specific hardware capture input function conditions are determined by the settings of the hardware capture event selection registers (HCPAR, HCPBR).

18.3.9.6 Hardware Sync Count

Each unit can choose to use the hardware input as CLOCK to count, and the units with the same hardware counting conditions can realize synchronous counting when the hardware counting clock is valid. The specific hardware counting conditions are determined by the settings of the hardware increment event selection register (HCUPR) and the hardware decrement event selection register (HCDOR).

When the hardware synchronous counting function is selected, only the external input clock source is selected, which does not affect the start, stop, and reset actions of the timer. The start, stop, and reset of the timer also need to be set separately.

Figure 18-8 shows an example of hardware synchronization operation of units 1 and 2.

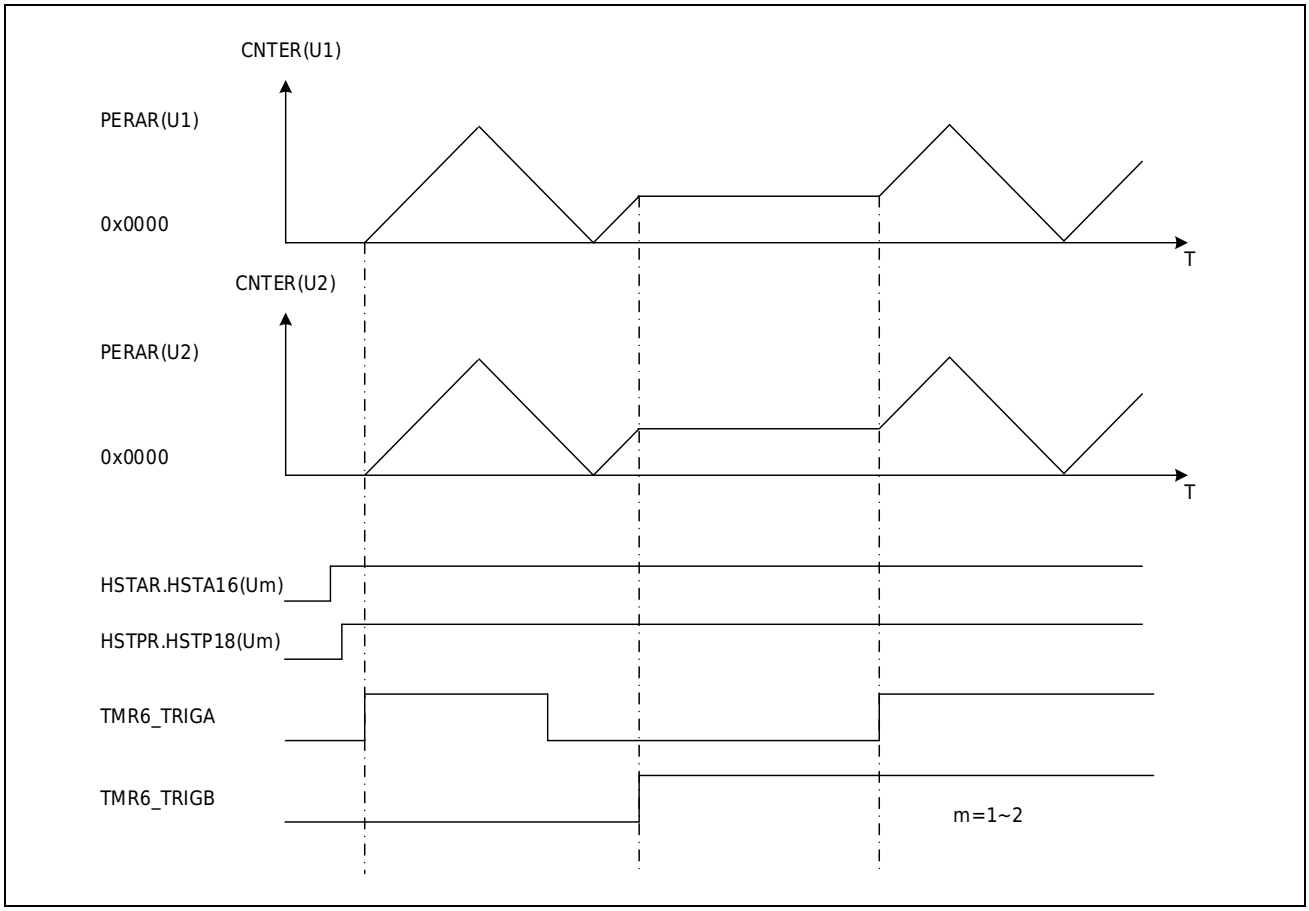


Figure 18-8 Hardware Synchronous Action

18.3.10 Pulse Width Measurement

When using the hardware trigger related functions of the TMR6_TRIGA~D port and TMR6_<t>_PWMA~B port (refer to the [Hardware Synchronization] section), each unit can realize independent pulse width measurement functions.

For example, by setting the hardware start condition of the counter to the rising edge of TMR6_TRIGA, and the hardware clear condition, stop condition, and capture input condition of the GCMAR register to the falling edge of TMR6_TRIGA, continuous pulse width measurement can be achieved. The corresponding actions are shown in Figure 18-9.

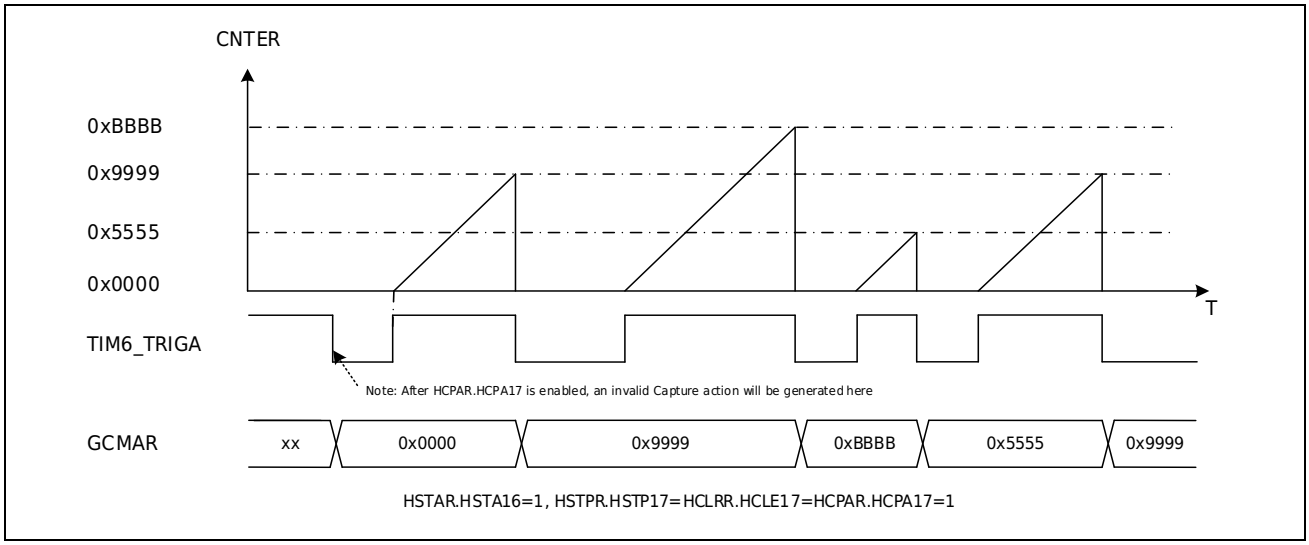


Figure 18-9 Pulse Width Measurement

18.3.11 Periodic Measurement

When using the hardware trigger related functions of the TMR6_TRIGA~D port and TMR6_<t>_PWMA~B port (refer to the [Hardware Synchronization] section), each unit can implement independent period measurement functions.

For example, by setting the counter's hardware start condition, hardware clear condition, and capture input condition of the GCMBR register to the rising edge of TMR6_TRIGB, continuous period measurement can be achieved. The corresponding actions are shown in Figure 18-10.

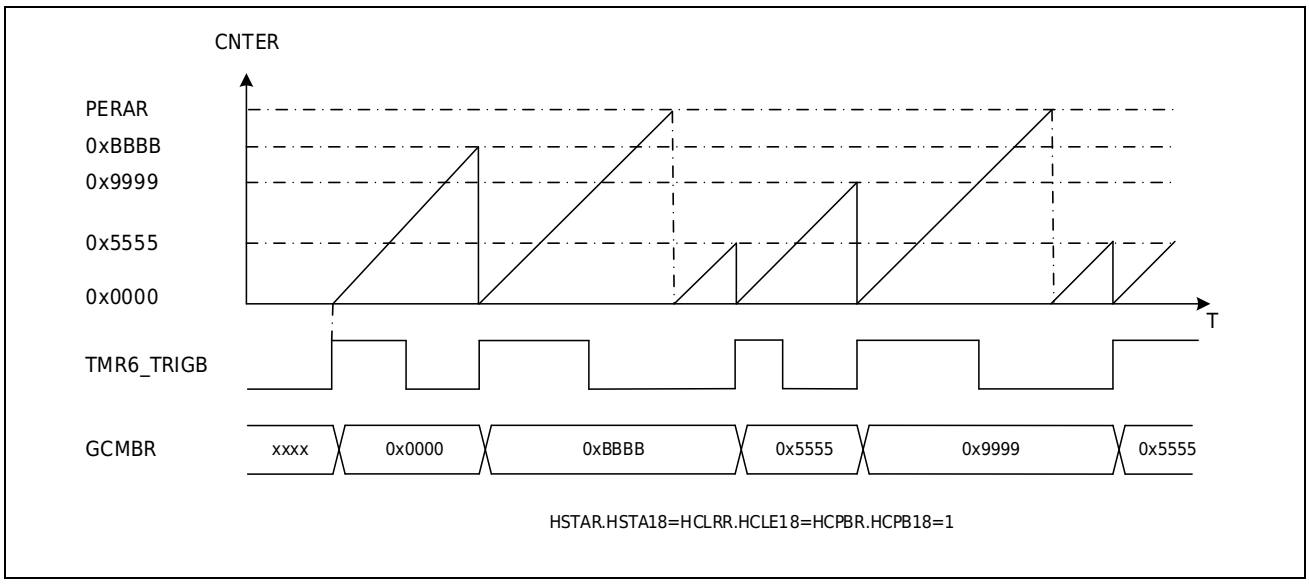


Figure 18-10 Cycle Measurement

18.3.12 Cache Function

The Universal period reference value, Universal comparison reference value, special comparison reference value, dead time reference value, etc. of Timer6 all have a cache function, which can realize period changes, duty cycle changes, dead time changes, etc. in hardware during counting.

The period reference value, Universal comparison reference value and special comparison reference value have single buffer and double buffer functions, and the dead time reference value has single buffer function.

18.3.12.1 Single Cache Action

Single buffer action means that by setting the buffer control register (BCONR.BENA<B><P><SPA><SPB>=1, BCONR.BSEA<B><P><SPA><SPB>=0) and the dead zone control register (DCONR.DTBENU<D>=1), the following events are selected to occur at the buffer transfer time point:

a) The value of the universal period reference buffer register (PERBR) is automatically transferred to the universal period reference register (PERAR)
b) The value of the universal comparison reference value buffer register (GCMCR, GCMDR) is automatically transferred to the universal comparison reference value register (GCMAR, GCMBR) (when comparing output)
c) The value of the universal comparison reference value register (GCMAR, GCMBR) is automatically transferred to the universal comparison reference value buffer register (GCMCR, GCMDR) (when capturing input)
d) The value of the dedicated comparison reference value buffer register (SCMCR, SCMDR) is automatically transferred to the dedicated comparison reference value register (SCMAR, SCMBR)
e) The value of the dead time reference buffer register (DTUBR, DTDBR) is automatically transferred to the dead time reference register (DTUAR, DTDAR)

As shown in Figure 18-11, this is the timing diagram of the single cache mode of the Universal Comparison Reference Value Register when unit 1 compares and outputs (PCNAR.CAPMDA = 0). It can be seen from the figure that changing the value of the universal comparison reference value register (GCMAR) during counting can adjust the output duty cycle, and changing the value of the universal period reference value register (PERAR) can adjust the output cycle.

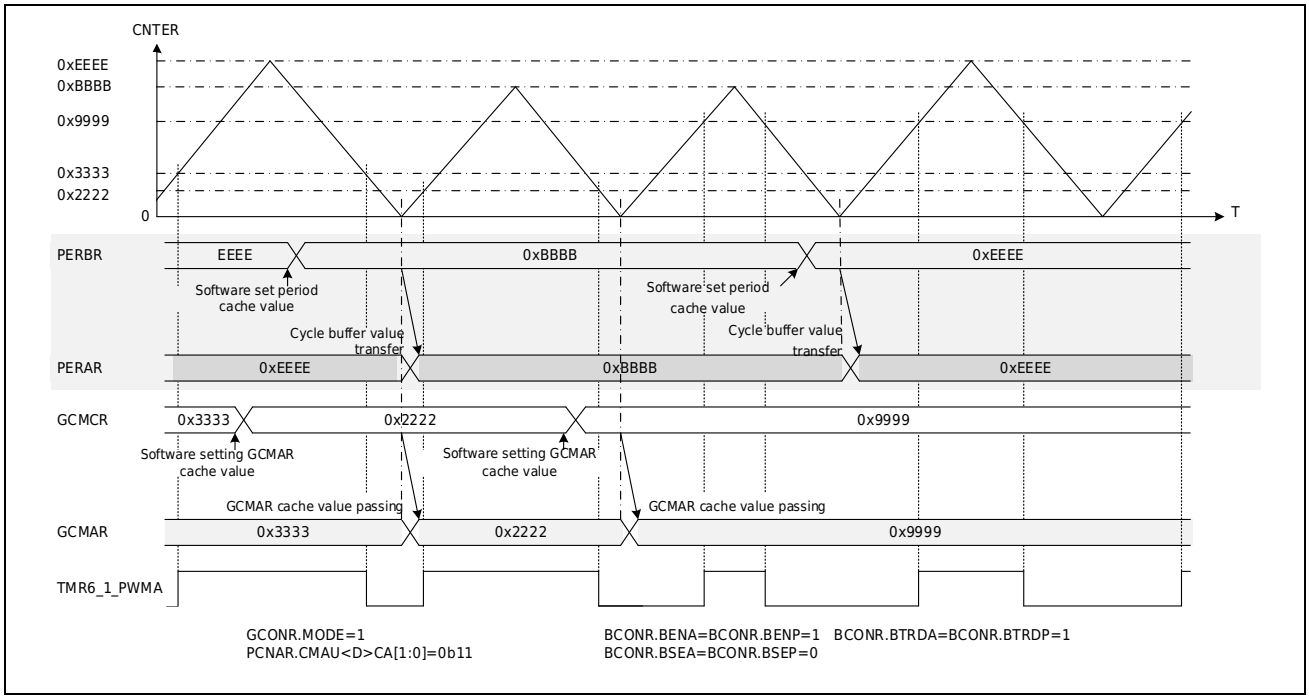


Figure 18-11 Single-buffer Mode Comparison Output Timing

18.3.12.2 Double Buffering Action

The double buffering action means that by setting the buffer control register (BCONR.BENA<B><P><SPA><SPB>=1, BCONR.BSEA<B><P><SPA><SPB>=1), the following events are selected to occur at the buffer transfer time point:

a) The value of the universal period reference value buffer register (PERBR) is automatically transferred to the universal period reference value register (PERAR), and the value of the universal period reference value double buffer register (PERCR) is automatically transferred to the universal period reference value buffer register (PERBR)

b) The value of the universal comparison reference value buffer register (GCMCR, GCMDR) is automatically transferred to the universal comparison reference value register (GCMAR, GCMBR), and the value of the universal comparison reference value double buffer register (GCMER, GCMFR) is automatically transferred to the universal comparison reference value buffer register (GCMCR, GCMDR) (when comparing output)

c) The value of the universal comparison reference value buffer register (GCMCR, GCMDR) is automatically transferred to the universal comparison reference value double buffer register (GCMER, GCMFR), and the value of the universal comparison reference value register (GCMAR, GCMBR) is automatically transferred to the universal comparison reference value buffer In registers (GCMCR, GCMDR) (when capturing input)

d) The value of the dedicated comparison reference value cache register (SCMCR, SCMDR) is automatically transferred to the dedicated comparison reference value register (SCMAR, SCMBR), and the value of the dedicated comparison reference value double buffer register (SCMER, SCMFR) is automatically transferred to the dedicated comparison reference value

cache register (SCMCR, SCMDR). As shown in Figure 18-12, it is a timing diagram of the double buffer mode when the internal trigger event 0 triggers the capture input.

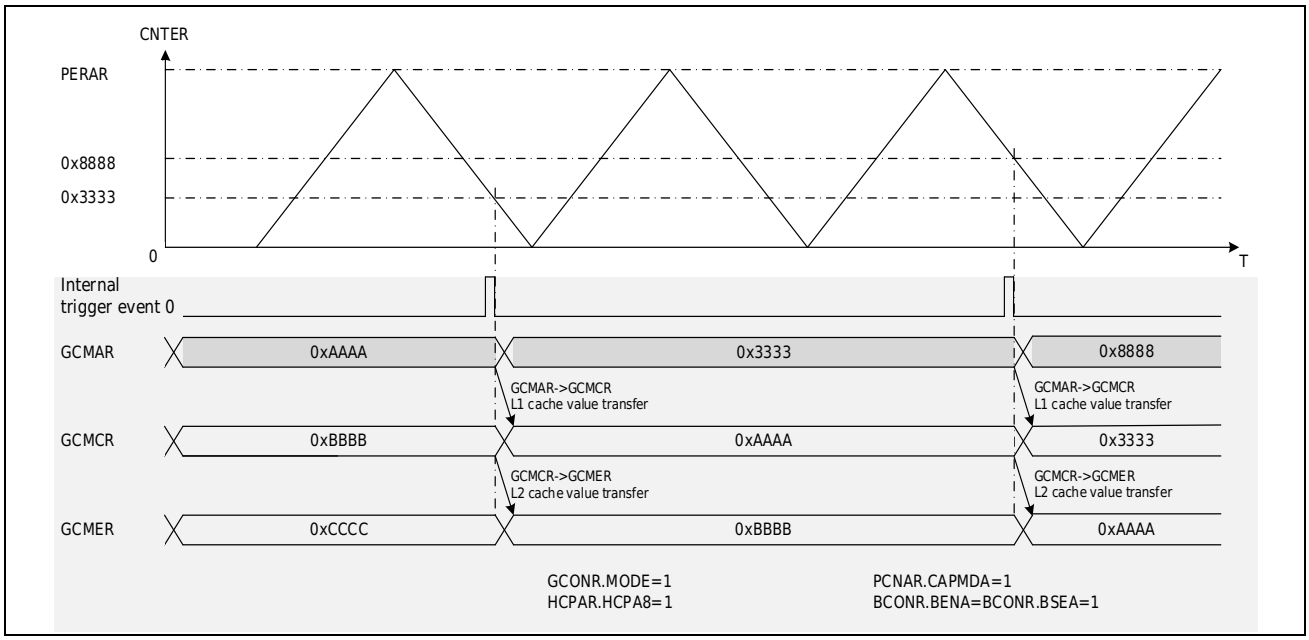


Figure 18-12 Double Buffer Capture Input Timing

18.3.12.3 Cache Transfer Node

Compare Output (Sawtooth Wave) Buffer Transfer

When the cache function valid (BCONR.BENA<B><P><SPA><SPB>=DCONR.DTBENU<D>=1) && comparison output count (PCNA<B>R.CAPMDA<B>=0) && sawtooth counting mode (GCONR.MODE=0), the cache transfer of the Universal period reference value, Universal comparison reference value, dedicated comparison reference value, and dead time reference value occurs at the overflow point of the up-counting or the underflow point of the down-counting.

As shown in Figure 18-13, this is the cache action when the sawtooth wave counts up.

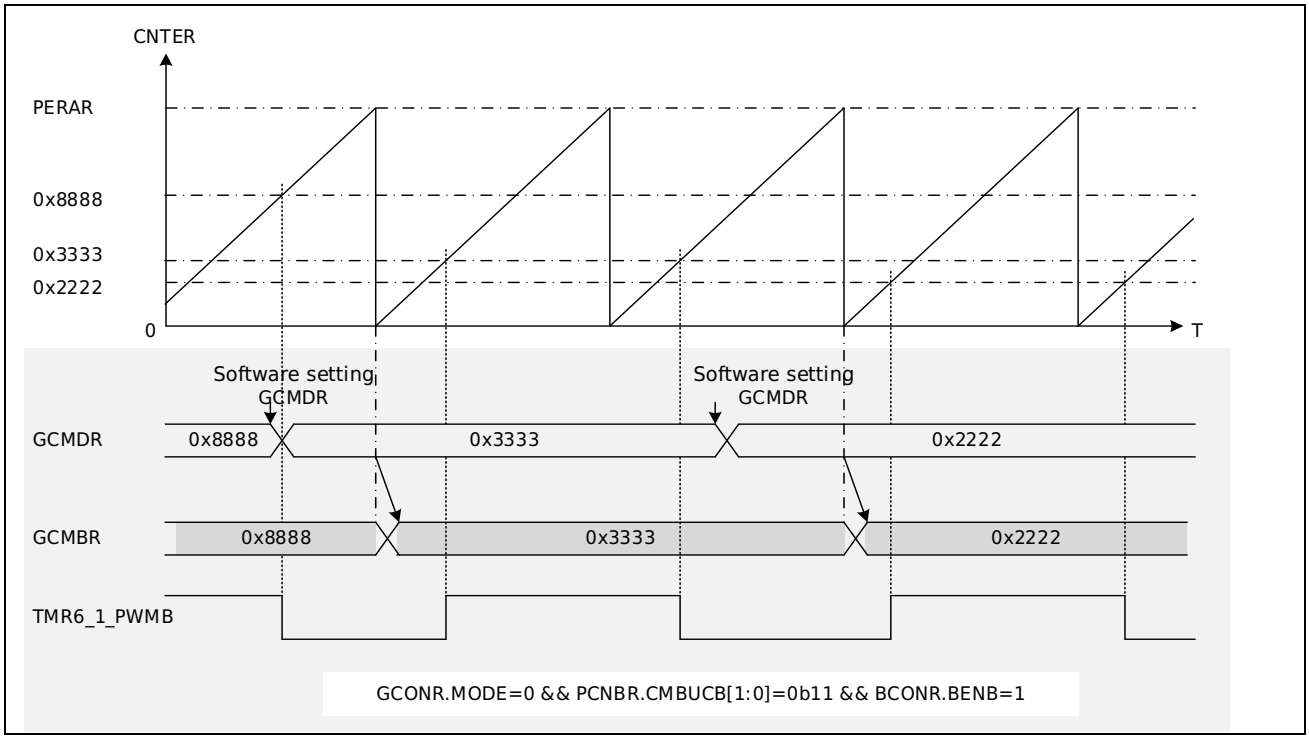


Figure 18-13 Counting Buffer Action in Sawtooth Wave Mode

Note:

- *In sawtooth counting mode, if a reset action occurs, it is also regarded as a count overflow. Each benchmark value will cause a cache transfer according to the corresponding cache action setting (single buffer, double buffer, etc.).*
- *In hardware counting mode, if a reset action occurs, it is also regarded as a count overflow. The Universal cycle reference value and the Universal comparison reference value will generate a cache transfer according to the corresponding cache action setting status (single cache, double cache, overflow transfer, underflow transfer, etc.), and other reference values will not generate cache transfer.*

Comparison Output (Triangle Wave) Buffer Transfer

When the cache function valid (BCONR.BENA<B><P><SPA><SPB>=DCONR.DTBENU<D>=1) && comparison output count (PCNA<B>R.CAPMDA<B>=0) && triangle wave counting mode (GCONR.MODE=1), the cache transfer time point of the Universal period reference value, Universal comparison reference value, dedicated comparison reference value, and dead time reference value is determined by the corresponding register control bits. When BCONR.BTRUA<B><P><SPA><SPB>=1 or DCONR.DTBTRU=1, the corresponding cache transfer occurs when the counter counts to the peak of the triangular wave; when BCONR.BTRDA<B><P><SPA><SPB>=1 or DCONR.DTBTRD=1, the corresponding cache transfer occurs when the counter counts to the valley of the triangular wave.

As shown in Figure 18-14, it is a schematic diagram of the cache action when the triangle wave counts to the valley point, As shown in the Figure 18-15, it is a schematic diagram of the cache action occurring when the triangle wave counts to the peak point and the valley point.

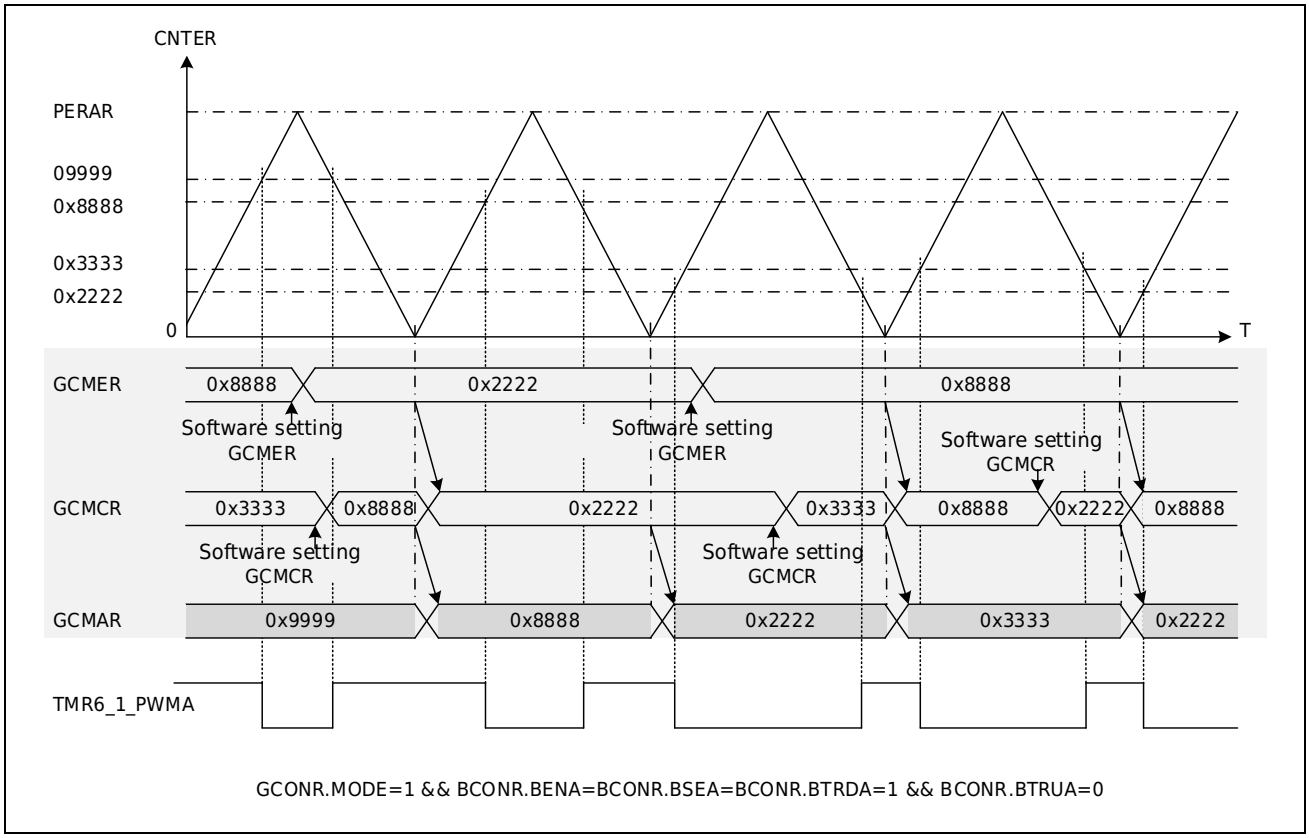


Figure 18-14 Count Buffer Action in Triangle Wave Mode

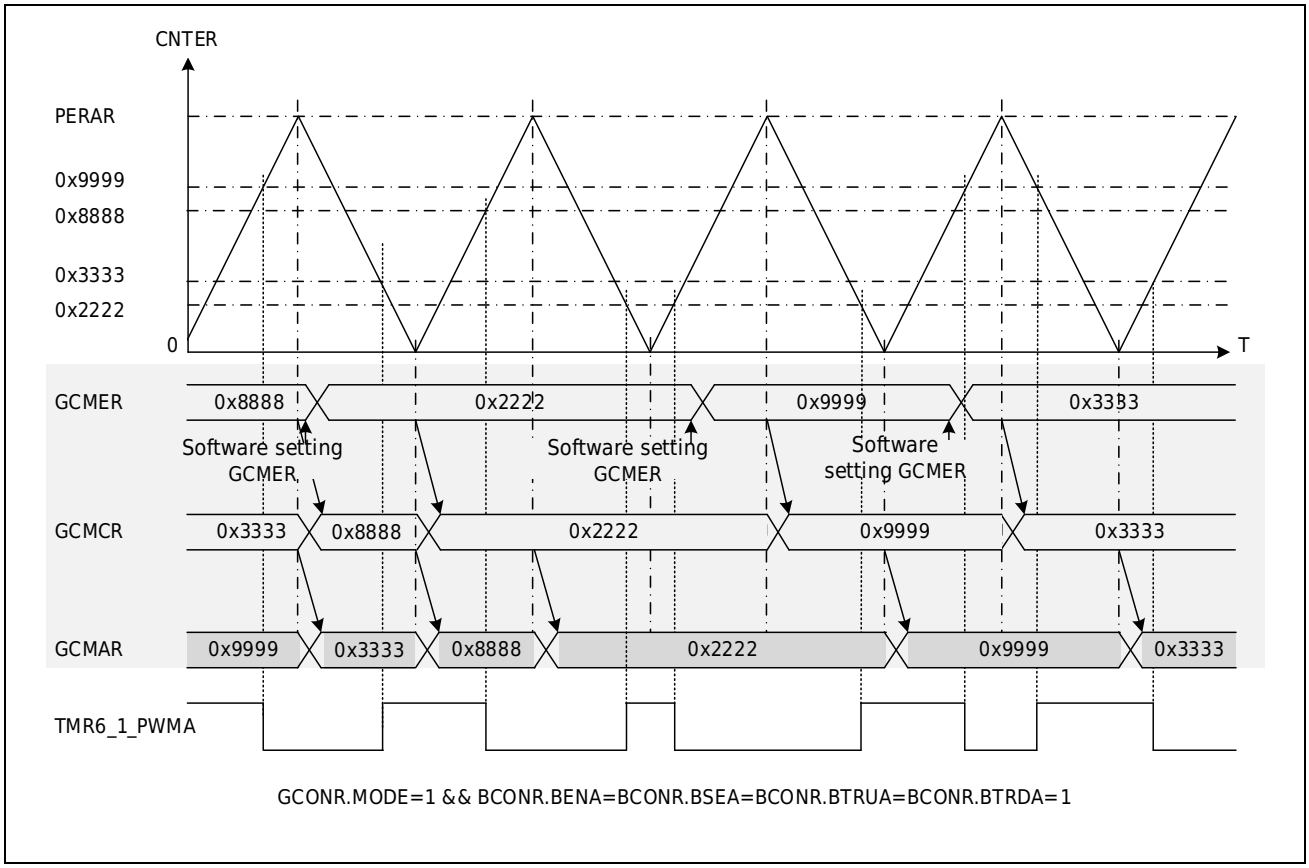


Figure 18-15 Count Buffer Action in Triangle Wave Mode

Capture Tnput Buffer Transfer

When the capture input action (PCNA<B>R.CAPMDA<B>=1) valid, the common comparison reference value supports the cache function. If the buffer setting valid (BCONR.BENA<B>=1), a buffer transfer will occur at the capture input action point. The capture input action can select single buffer function or double buffer function (set by BCONR.BSEA<B>).

18.3.13 Digital Filtering

TMR6_<t>_PWMA, TMR6_<t>_PWMB, TMR6_TRIGA~D port inputs all have digital filtering function. TMR6_<t>_PWMA and TMR6_<t>_PWMB enable the filtering function of the corresponding port by setting the relevant enable bits of the filtering control register (FCNGR). The filtering reference clock when the filtering is effective can also be set through the filtering control register (FCNGR). TMR6_TRIGA~D ports are ports shared by a group of units. The filtering function of the corresponding port is enabled by setting the relevant enable bits of the filtering control register (FCNTR). The filtering reference clock when the filtering is effective can also be set through the filtering control register (FCNTR).

When the filtered sampling reference clock is sampled to the same level three times on the port, the level is transferred to the module as an effective level. A level less than three times consistent will be filtered out as external interference and not transmitted to the module. Its action is as shown in Figure 18-16 the example.

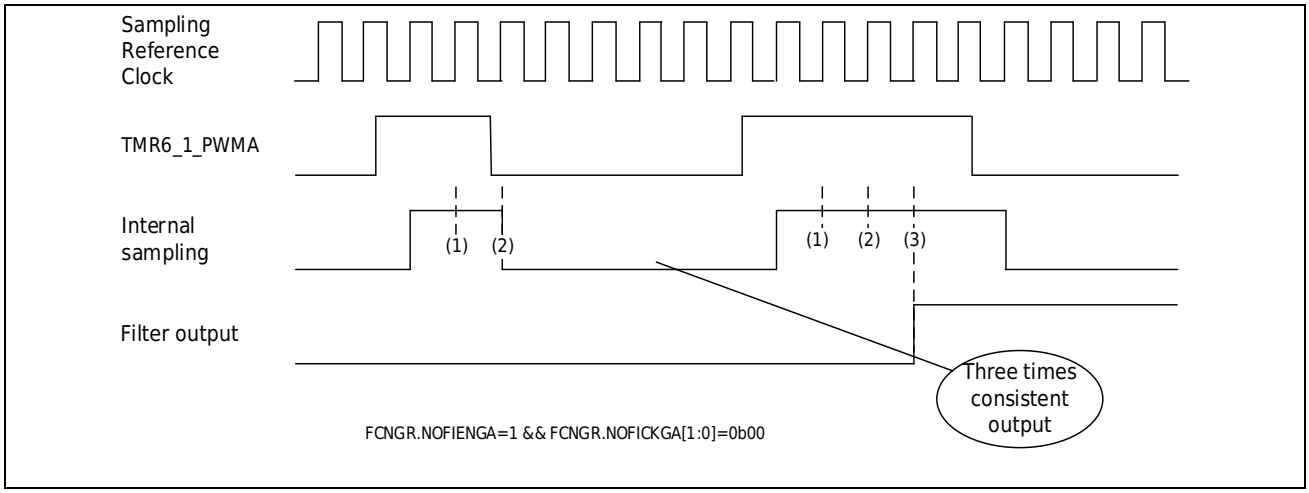


Figure 18-16 Filtering function of the capture input port

18.3.14 Universal PWM Output

18.3.14.1 Single-edge Aligned Independent PWM Output

In sawtooth counting mode (GCONR.MODE=0), the two ports TMR6_<t>_PWMA and TMR6_<t>_PWMB of each unit can output PWM waves independently. If the same level change is set at the count overflow point, single-side aligned independent PWM output can be achieved. As shown in Figure 18-17 below.

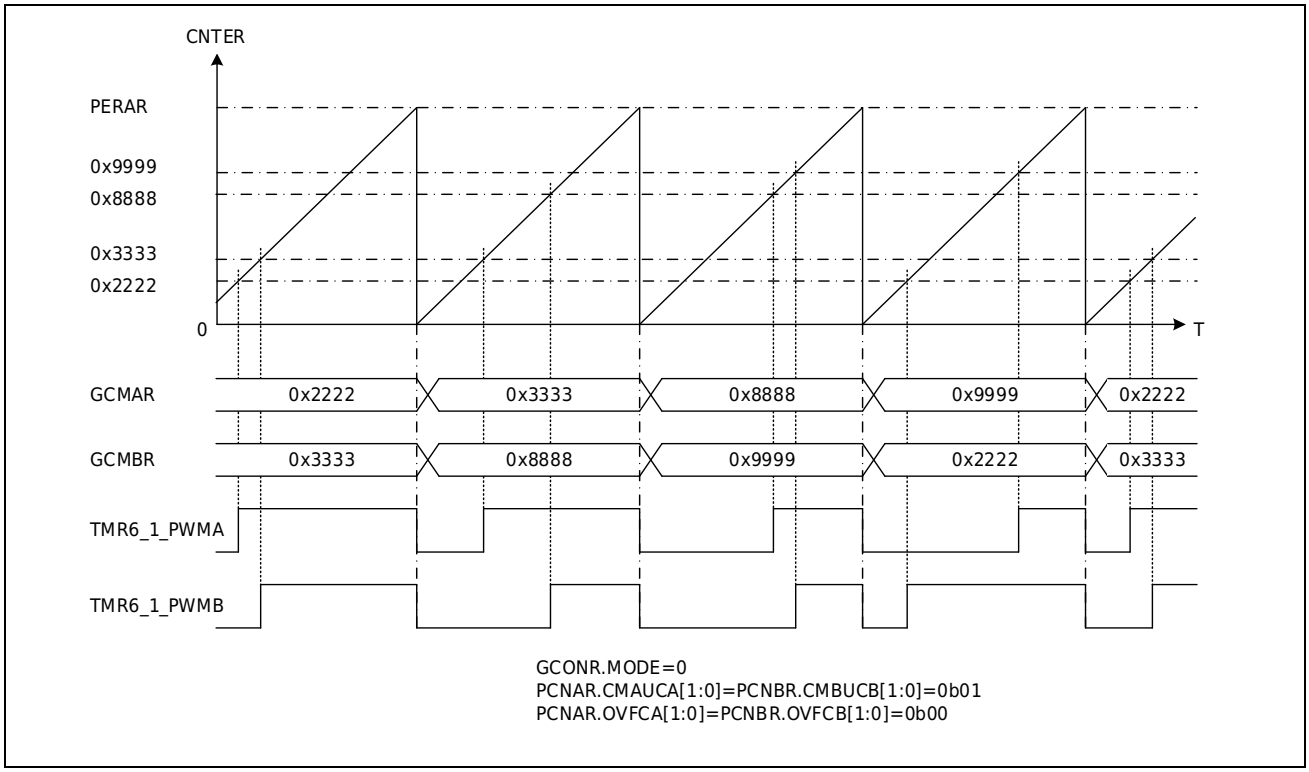


Figure 18-17 Single-edge Aligned Independent PWM

18.3.14.2 Symmetrical Independent PWM Output on Both Sides

In the triangle wave counting mode (GCONR.MODE=1), the two ports TMR6_<t>_PWMA and TMR6_<t>_PWMB of each unit can output PWM waves independently. If the set level changes at the count comparison match point and the set level does not change at the count peak and valley points, bilateral symmetrical independent PWM output can be achieved. As shown in Figure 18-18, the TMR6_<t>_PWMA port and the TMR6_<t>_PWMB port implement bilateral symmetrical independent output PWM.

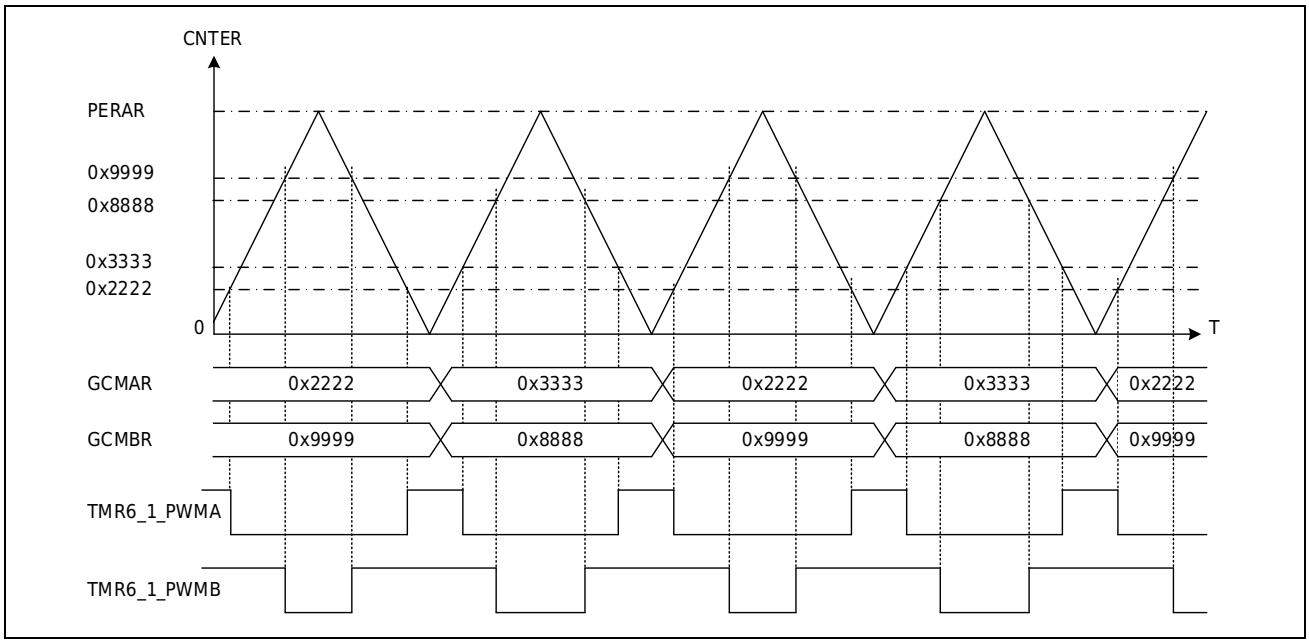


Figure 18-18 Bilateral Symmetrical Independent PWM

18.3.14.3 Bilateral Symmetrical Complementary PWM Output

In the triangle wave counting mode (GCONR.MODE=1), by setting the level state of the port at count start, compare match, and count overflow, a pair of complementary PWM waveforms can be output on the TMR6_<t>_PWMA and TMR6_<t>_PWMB ports.

According to the different ways of assigning the GCMBR reference value, the output of bilateral symmetrical complementary PWM can be divided into "software-set GCMBR complementary PWM output" and "hardware-set GCMBR complementary PWM output".

Software Setting GCMBR Complementary PWM Output

The software setting method means that in the triangle wave mode, the Universal Comparison Reference Value Register (GCMBR) used for the waveform output of the TMR6_<t>_PWMB port is directly written by the CPU, etc., and has no direct relationship with the value of GCMAR.

Figure 18-19 Example of outputting GCMBR complementary PWM wave for software setting.

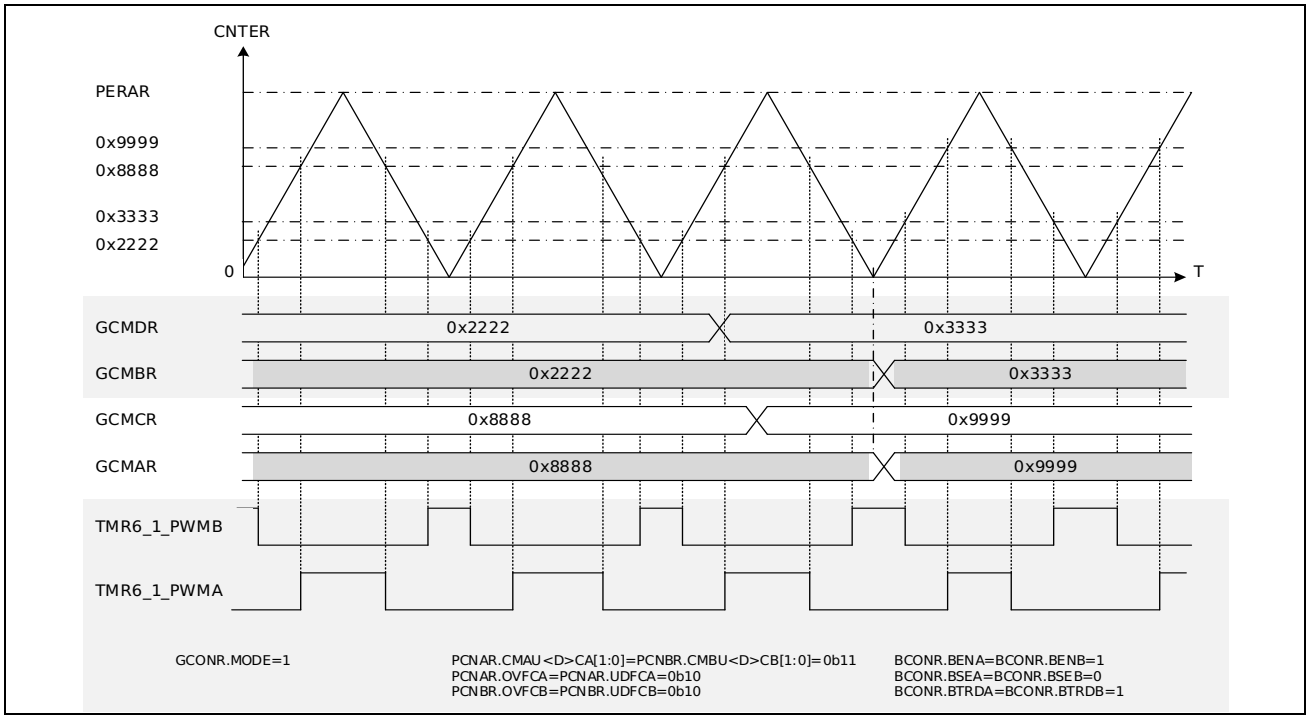


Figure 18-19 Software Setting of GCMBR Complementary PWM Wave Output

Hardware Setting GCMBR Complementary PWM Output

The hardware setting method means that in the triangle wave mode, the value of the Universal Comparison Reference Value Register (GCMBR) used for the waveform output of the TMR6_<t>_PWMB port is determined by the value operation of the Universal Comparison Reference Value Register (GCMAR) and the dead time reference value register (DTU<D>AR).

Figure 18-20 Example of setting GCMBR complementary PWM wave output for hardware.

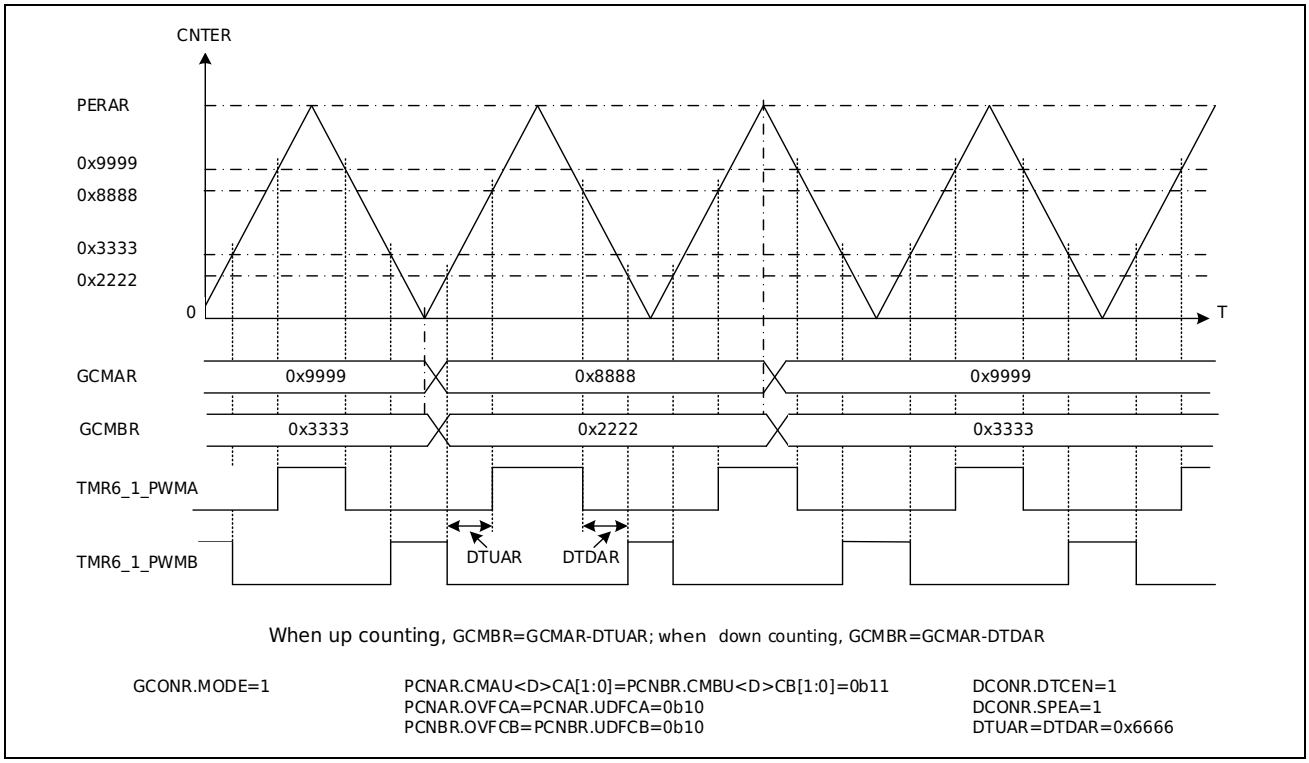


Figure 18-20 Hardware Setting GCMBR Complementary PWM Wave Output

18.3.14.4 Double-sided Asymmetric PWM Output

In the triangle wave counting mode (GCONR.MODE=1), the two ports TMR6_<t>_PWMA and TMR6_<t>_PWMB of each unit can output PWM waves independently, and each port can make corresponding level output changes according to the reference of the Universal Comparison Reference Value Register (GCMAR, GCMBR). If the port level change of TMR6_<t>_PWMA is controlled by the comparison results of GCMAR and GCMBR reference values during the up-counting and down-counting periods of the counter, bilateral asymmetric PWM output can be achieved on the TMR6_<t>_PWMA port. As shown in Figure 18-21, it is the double-sided asymmetric PWM output of TMR6_<t>_PWMA port.

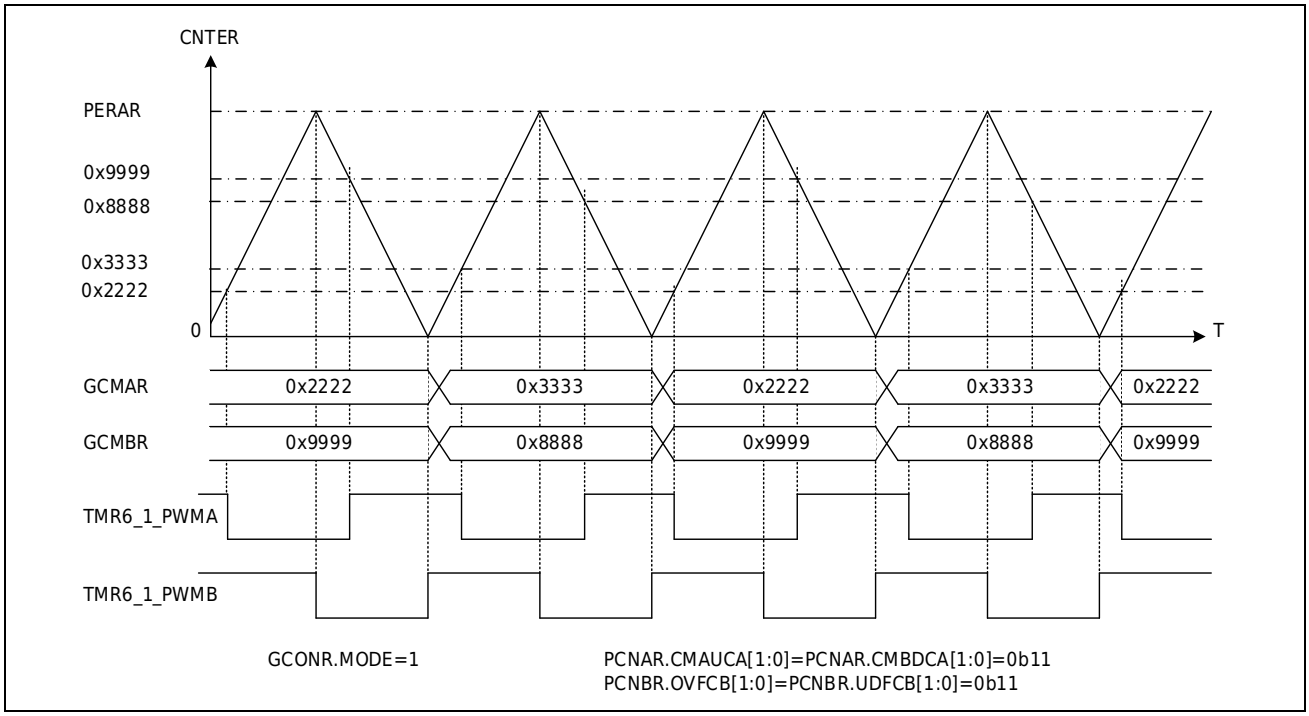


Figure 18-21 Double-sided Asymmetric PWM Output

18.3.14.5 Multi-phase PWM Output between Units

The TMR6_<t>_PWMA and TMR6_<t>_PWMB ports of each unit can output 2-phase PWM waves. By combining multiple units and combining software and hardware synchronization actions, multi-phase PWM wave output can be achieved. As shown in Figure 18-22, the combination of unit 1, unit 3, unit 1, and unit 4 outputs 8-phase single-side aligned independent PWM; As shown in Figure 18-23, the combination of unit 1, unit 3, unit 1, and unit 4 outputs 4-phase double-side symmetrical complementary PWM.

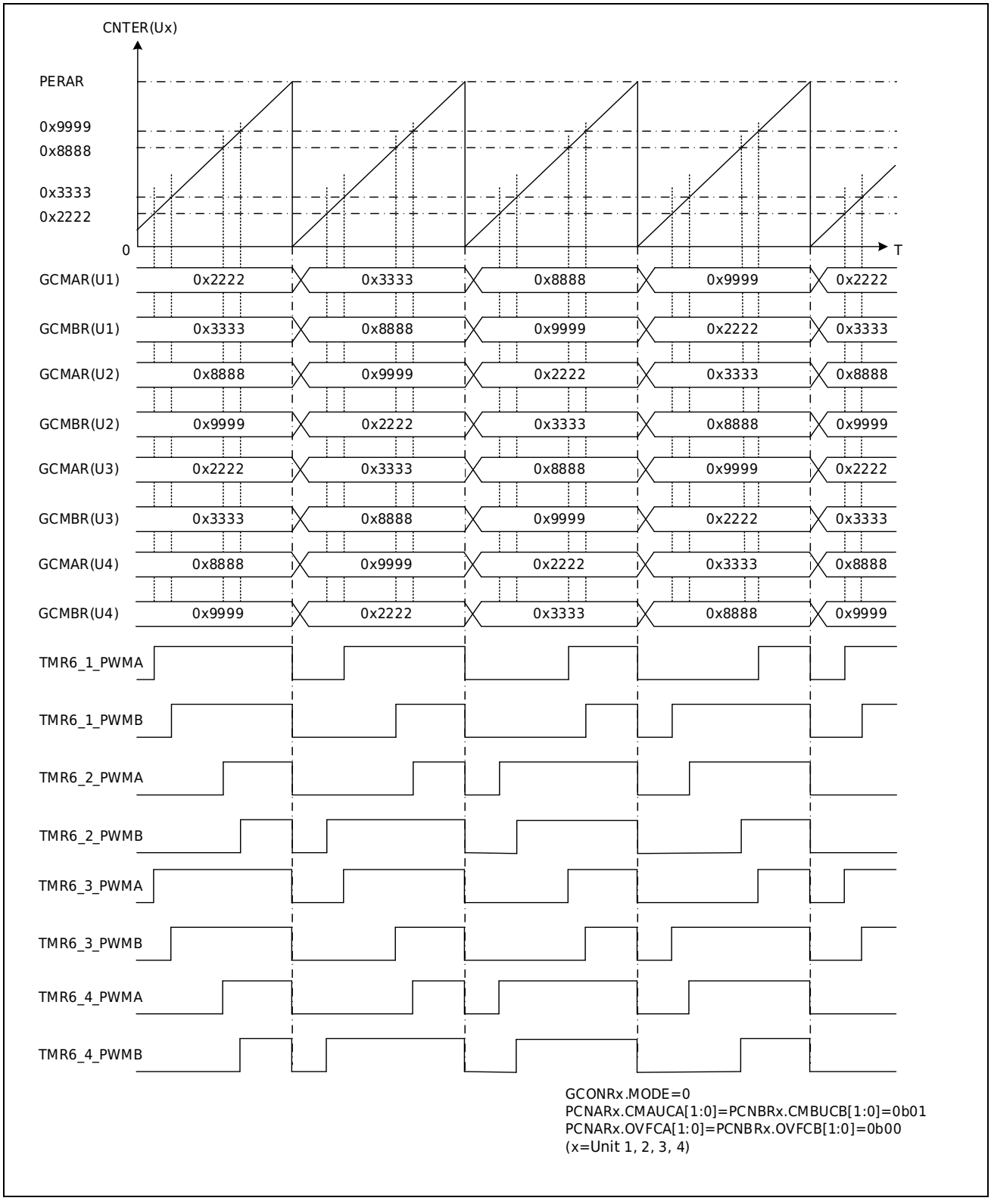


Figure 18-22 Phase Single Edge Aligned Independent PWM

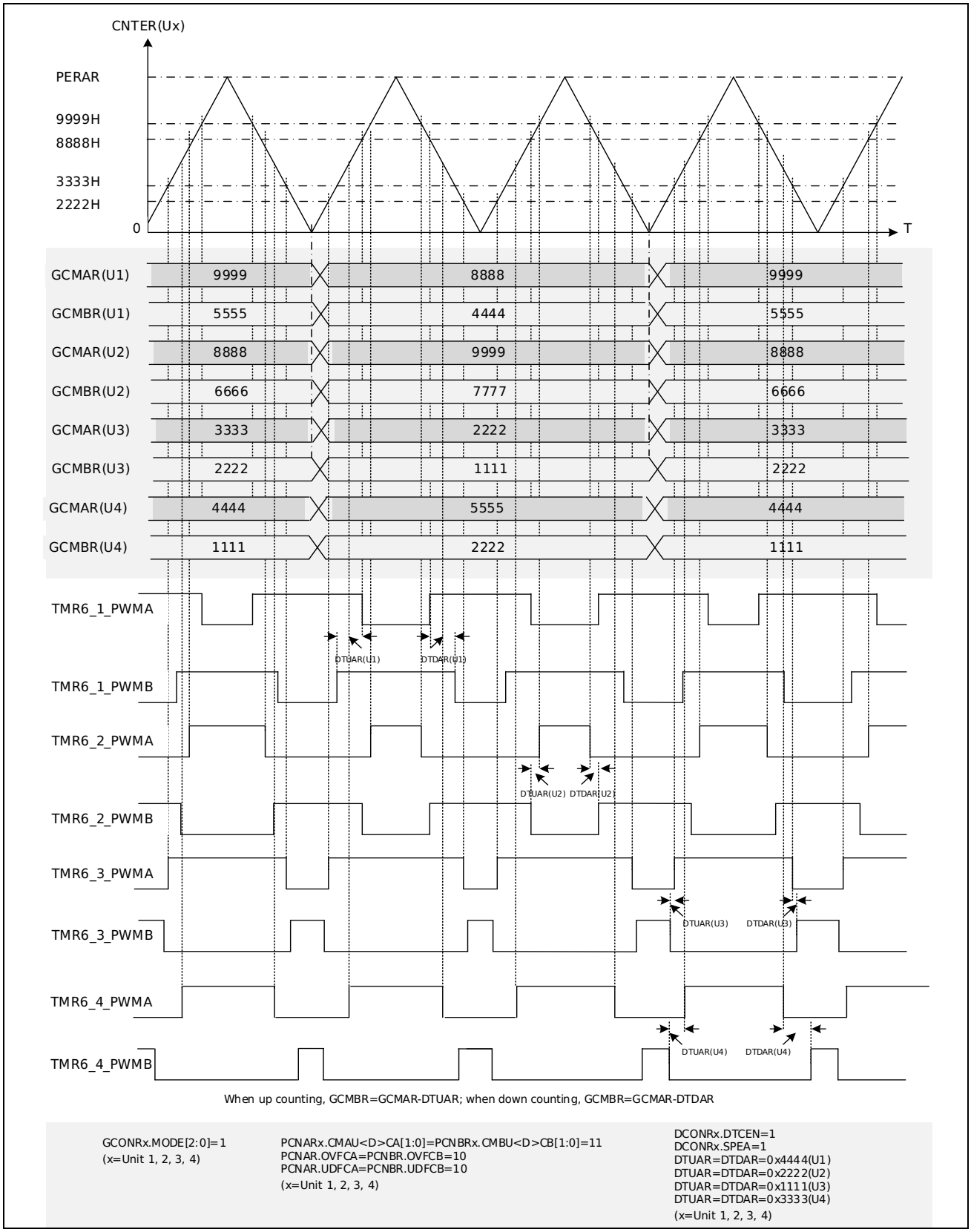


Figure 18-23 4-phase Bilateral Symmetrical Complementary PWM with Dead Time

18.3.15 Periodic Interval Response

The two dedicated comparison reference value registers (SCMAR, SCMBR) of Timer6 can respectively output the special comparison match interrupt A signal and the special comparison match interrupt B signal to INTC to generate the corresponding interrupt when the counting comparison matches; same time, they can output the special comparison match The event A signal and the dedicated comparison match event B signal are used to associate actions with other modules, and are mostly used to start the ADC, etc.

The interrupt and event request signal can generate an effective request signal every several cycles, that is, realize periodic interval response. This function is enabled by setting the VPERR.PCNTE[1:0] bits and VPERR.SPPERIA/B bits of the valid period register (VPERR). Set the VPERR.PCNTS[2:0] bit to specify how many cycles the request signal is valid for once. In other cycles, even if the count value is equal to the value of the dedicated comparison reference value register SCMAR or SCMBR, a valid request signal will not be output. .

After this function is enabled, the period match interrupt and period match event in each waveform mode are only output in the valid period of the dedicated compare match interrupt and event output (the period of STFLR.VPERNUM=0 in the figure below). Figure 18-24 shown is an example of the operation of the periodic interval valid request signal.

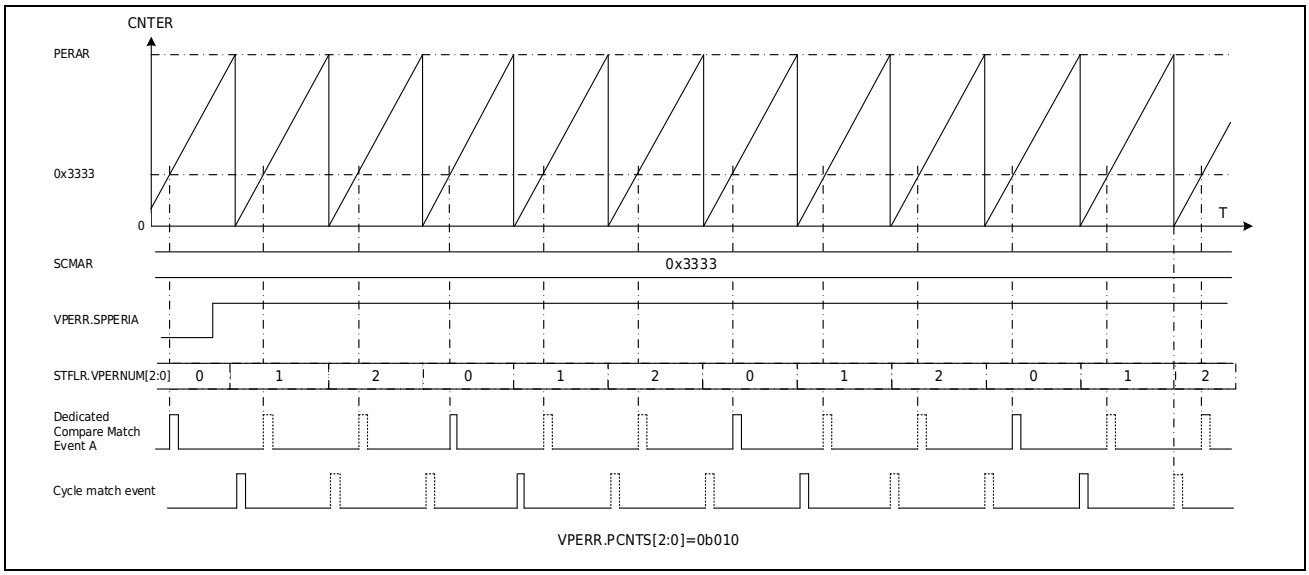


Figure 18-24 Cycle Interval Valid Request Signal Action

18.3.16 Orthogonal Coding Count

By considering TMR6_<t>_PWMA input as AIN input, TMR6_<t>_PWMB input as BIN input, and any one of TMR6_TRIGA~D inputs as ZIN input, Timer6 can implement quadrature encoding counting of three inputs.

The single action of AIN and BIN of one unit can realize the position counting mode; the combined action of AIN, BIN and ZIN of two units can realize the revolution counting mode, one unit is used for position counting, and the other is used for revolution counting.

In revolution counting mode, the two units of Timer6 can be arbitrarily combined to be selected as position counting unit or revolution counting unit, but when the Z-phase shielding function valid, the combination is fixed, and units 1 and 2 are combined, with unit 1 as the position counting unit and unit 2 as the revolution counting unit, to realize position counting and revolution counting respectively.

The counting conditions of AIN and BIN are realized by setting the orthogonal relationship between TMR6_<t>_PWMA and TMR6_<t>_PWMB in the hardware increment event selection register (HCUPR) and the hardware decrement event selection register (HCDOR); the input action of ZIN is realized by setting the hardware clear event selection register (HCLRR) of the position unit to clear the position timer of the position counting unit, and by setting the hardware increment event selection register (HCUPR) of the revolution unit to count the revolution timer of the revolution counting unit..

18.3.16.1 Position Counting Mode

The quadrature encoding position counting mode refers to realizing the basic counting function, phase difference counting function and direction counting function according to the input of AIN and BIN.

Basic Count

The basic counting action is to count according to the input clock of the AIN or BIN port, as shown in Figure 18-25 below.

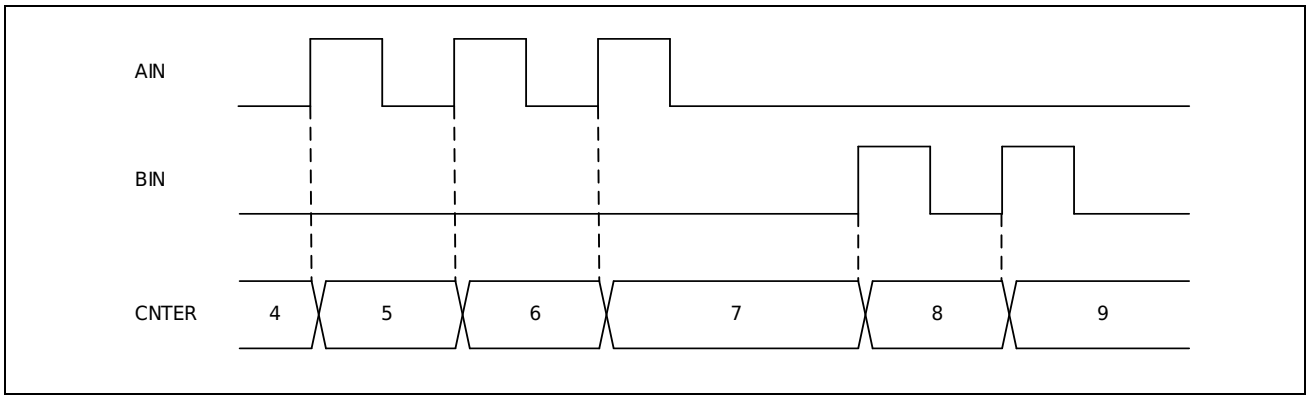


Figure 18-25 Position Mode - Basic Counting

Phase Difference Count

Phase difference counts are counted according to the phase relationship between AIN and BIN. According to different settings, 1-fold counting, 2-fold counting, 4-fold counting, etc. can be realized, as shown in the following Figure 18-26 ~ Figure 18-28.

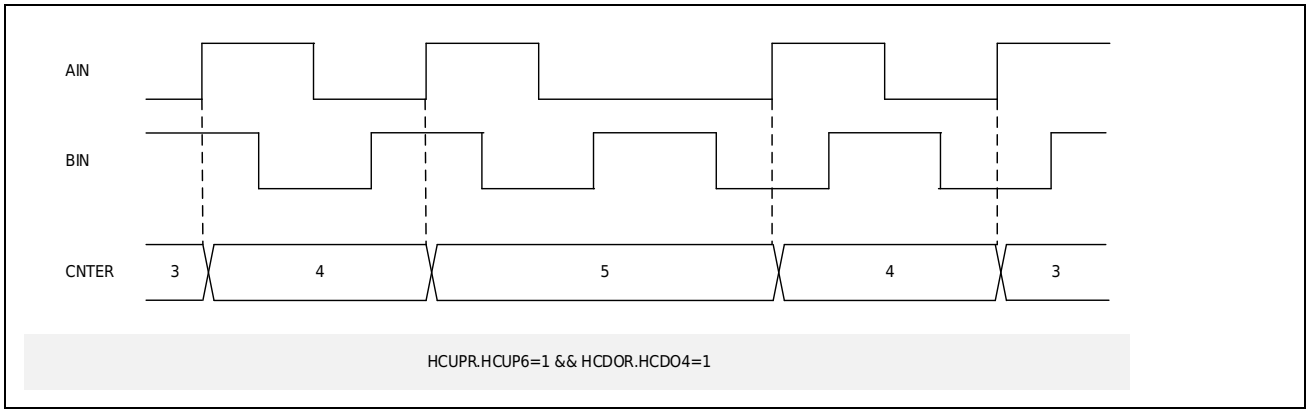


Figure 18-26 Position Counting Mode - Phase Difference Counting (1 times counting)

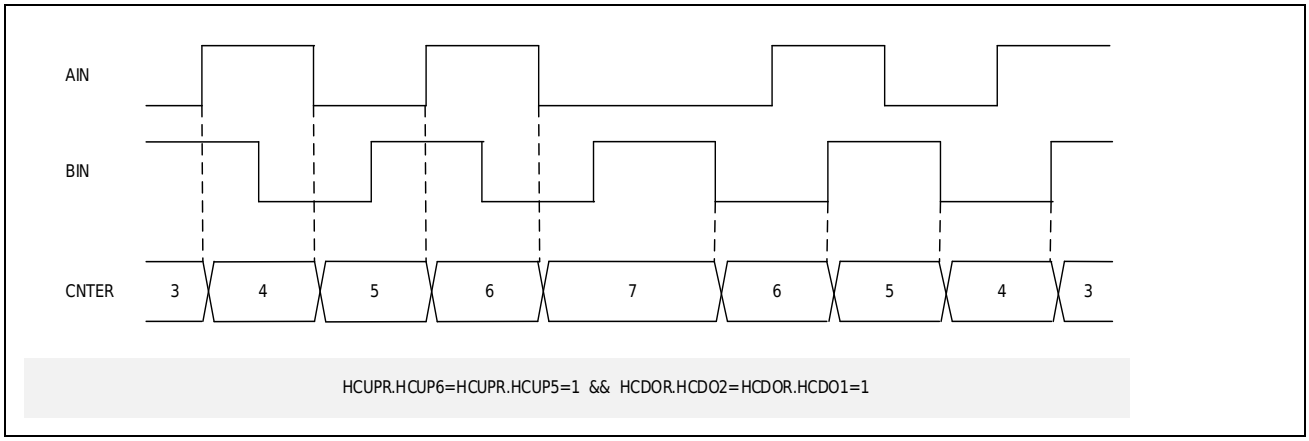


Figure 18-27 Position Counting Mode - Phase Difference Counting (2 times counting)

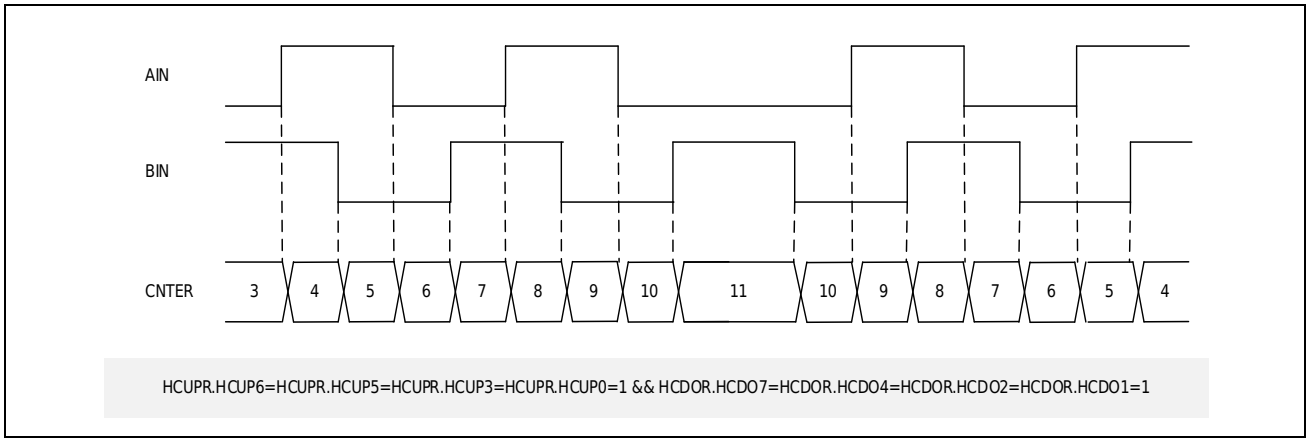


Figure 18-28 Position Counting mode - Phase Difference Counting (4 times counting)

Direction Count

Direction counting refers to setting the input state of AIN as direction control, and using the input of BIN as clock counting, as shown in the Figure 18-29 below.

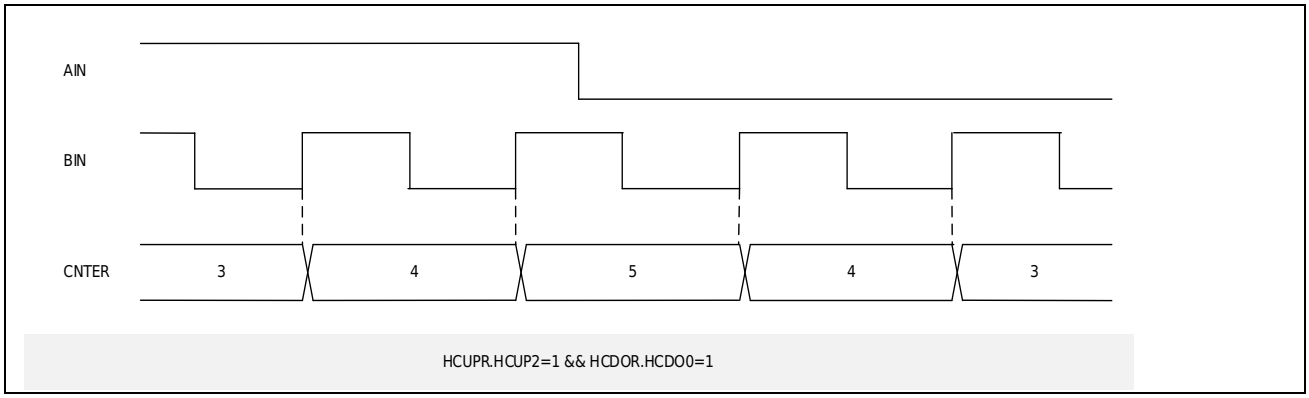


Figure 18-29 Position Counting Mode - Direction Counting

18.3.16.2 Revolution Counting Mode

Orthogonal encoding revolution counting mode refers to the addition of ZIN input events on the basis of AIN and BIN counting to realize the judgment of the number of revolutions, etc. In the revolution counting mode, according to the counting method of the revolution timer, the Z-phase counting function, the position overflow counting function and the mixed counting function can be realized.

Z Phase Count

Z-phase counting refers to the counting action that the revolution counting unit counts according to the input of ZIN, and the position counting unit is cleared at the same time. As shown in Figure 18-30 below.

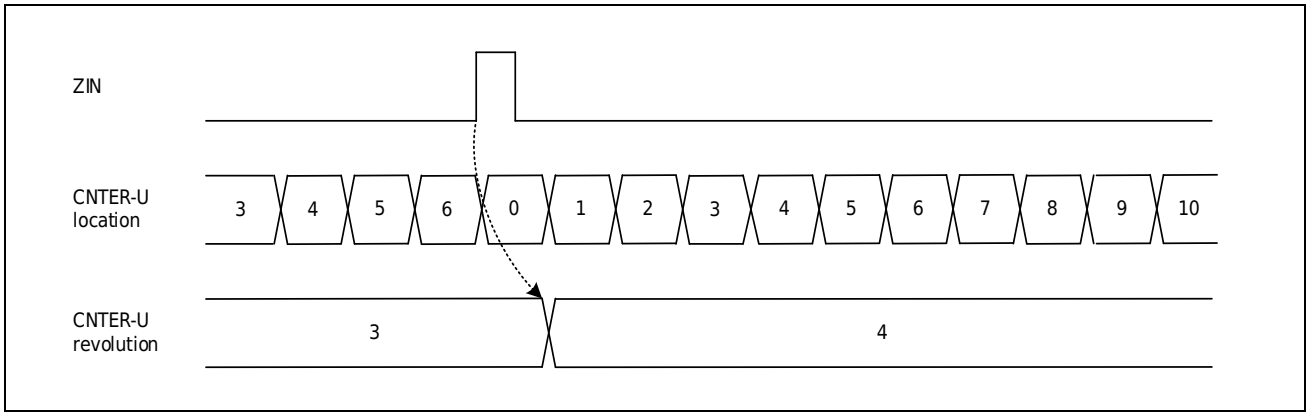


Figure 18-30 Revolution counting mode - Z phase counting

Position Overflow Counting

Position overflow counting means that when the counting of the position counting unit overflows or underflows, an overflow event is generated, thereby triggering the timer of the revolution counting

unit to count once (in this counting mode, the input of ZIN does not perform the counting action of the revolution counting unit and the clearing action of the position counting unit).

The overflow event of the position counting unit is gated through the internal trigger event interface to realize the counting of the revolution counting unit, and the position overflow counting can be realized. The increment (decrement) event selection register (HCUPR or HCDOR) of the revolution counting unit selects 1 bit of Bit8~Bit11, and sets the event number in the corresponding hardware trigger event selection register (TMR6_TRGSELx, x=0~3)) to the overflow or underflow event of the position counting unit. For specific register contents, please refer to the [Timer6 Hardware Trigger Event Selection Register (TMR6_TRGSELx) (x=0~3)] section, and for specific event numbers, please refer to the [Interrupt controller (INTC)] section. As shown in Figure 18-31 below.

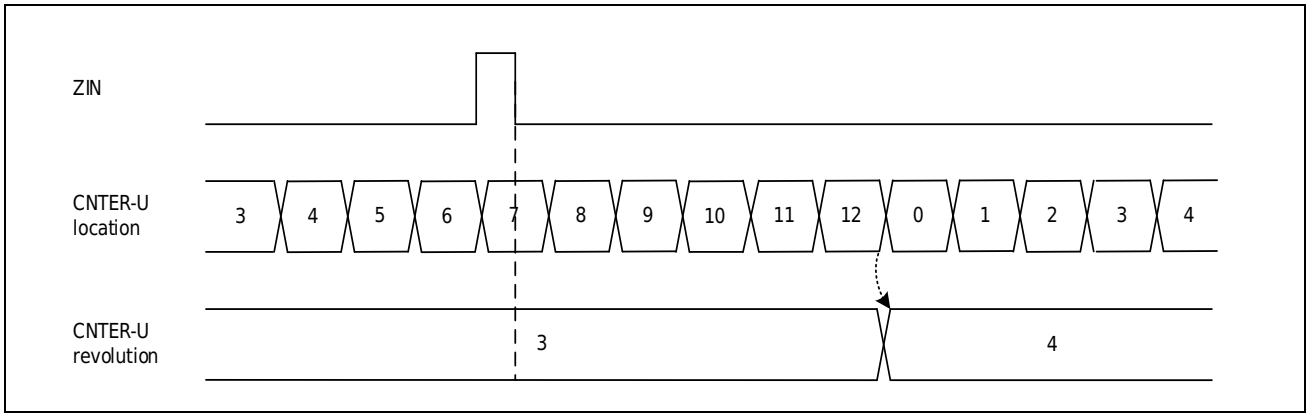


Figure 18-31 Revolution Counting mode - Position Overflow Counting

Mixed Counting

Mixed counting refers to the counting action that combines the above two counting methods of Z-phase counting and position overflow counting, and its realization method is also a combination of the above two counting methods. As shown in Figure 18-32 below.

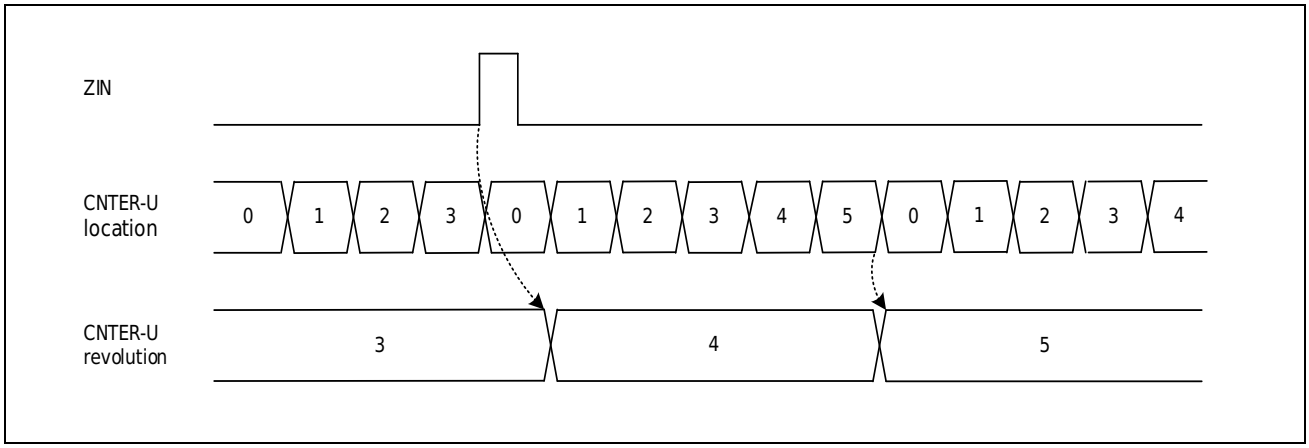


Figure 18-32 Revolution Counting Mode - Mixed Counting

18.3.16.3 Z-phase Motion Shielding

In the Z-phase counting function or mixed counting function of the revolution counting mode, it can be set within a few cycles after the overflow point or underflow point of the position timer (GCONR.ZMSKVAL[0:1] setting), the The effective input of ZIN is masked, and the counting of the revolution counting unit and the clearing of the position counting unit are not performed.

When the GCONR.ZMSKPOS of the Universal Control register (GCONR) of the position counting unit is 1, the Z-phase shielding function of the position counting unit is enabled, and the number of cycles of Z-phase shielding is set by GCONR. ZMSKVAL; the Universal Control register of the revolution counting unit (When GCONR.ZMSKREV of GCONR) is 1, the Z-phase shielding function of the revolution counting unit is enabled.

When Figure 18-33 is the mixed counting mode of revolution counting, there is a ZIN phase input within 4 counting cycles after the position counting unit overflows, the action of the ZIN phase input is invalid, that is, the revolution counting unit does not count, and the position counting unit does not clear; after that Then the ZIN phase input works normally.

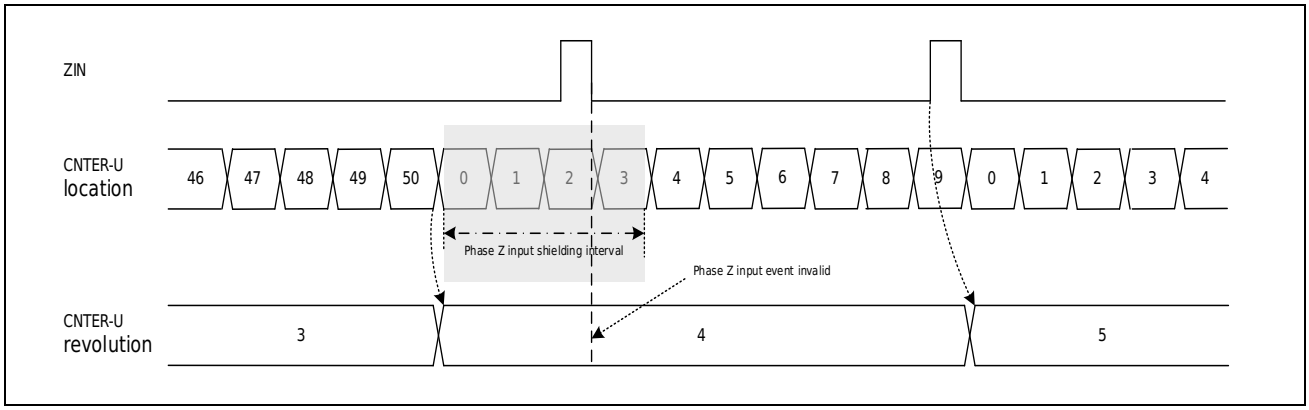


Figure 18-33 Revolution Counting mode - Mixed Counting Z-phase Shielding Action Example 1

When Figure 18-34 is the mixed counting mode of revolution counting., the counting diary changes in the third cycle after the posenting unit overflows, and the mask ING Cycle of 4 Cycles Set at this time Becomes Invalid (The Actual Zin Phase Masking Function Maintautas 3 Cycles), Start counting down. After the counting underflow occurs in the position counting unit, the ZIN phase shielding function is re-opened and becomes invalid after 4 cycles. During the masking period of the ZIN phase, the input function of the ZIN phase is invalid, that is, the revolution counting unit does not count, and the position counting unit does not clear; the subsequent ZIN phase input operates normally.

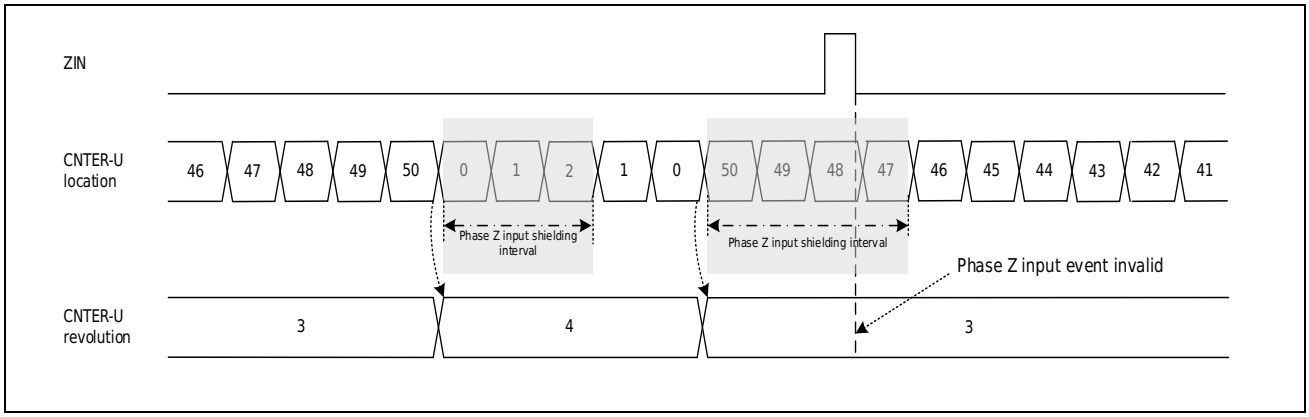


Figure 18-34 Revolution Counting Mode - Mixed Counting Z-phase Shielding Action Example 2

18.3.17 EMB Control

Timer6 can protect and control the output state of the port, and fix the port state to a preset safe state when an abnormality occurs. All units have a common port output control interface. Each unit selects the EMB event connection interface to be used by setting the port control register (PCNA<B>R.EMBSA<B>). This interface connects a group of EMB events output by the EMB module. At the same time, the abnormal condition events gated on the interface can be set from the EMB side (See chapter [Emergency Brake Module (EMB)]). When abnormal conditions are detected on these interfaces, the control of the universal PWM output can be realized.

During the normal output period of the port, if the EMB event from the EMB is detected, the output state of the port can be changed to a preset state. When an EMB abnormal event occurs, the Universal PWM output port can change its state to output high impedance, output low level or output high level (determined by the setting of PCNA<B>R.EMBCA<B>). For example, if PCNAR.EMBCA=01 is set, then during the normal output of the TMR6_<t>_PWMA port, if an EMB event occurs, the output on the TMR6_<t>_PWMA port becomes a high-impedance state.

After the EMB event invalid (the abnormal event connected to Timer6 from the EMB module disappears and the signal becomes the normal level), the output of the PWM port can automatically restore to normal output. At this time, you can choose to restore the normal PWM output immediately (PCNA<B>R.EMBRA<B>=00; this method is called One Shot release) or restore the normal PWM output after counting to the next overflow point (PCNA<B>R.EMBRA<B>=0b01, 0b10, 0b11; this method is called Cycle By Cycle release) through the setting of the port control register.

Note:

When the event release mode of EMB is configured as state release (configured by EMB_RLSSEL register), and the event source of EMB is configured as external port or the comparison result of internal voltage comparator valid, it must be ensured that the level width of the comparison result of external port and internal voltage comparator is greater than 3 count clock cycles, otherwise a glitch of up to 2 count clock cycles will appear on PWM output. When the event release mode of EMB is configured as software release (configured through the EMB_RLSSEL register), and the event

source of EMB is configured as PWM output port in phase, a glitch of up to 2 count clock cycles will occur before PWM in-phase output occurs and PWM enters the safe state.

18.3.18 Function Summary Table

In the sawtooth mode and triangular wave A and B modes of Timer6, the summary table of the main functions is shown in the Table 18-3 below.

Table 18-3 Comparison of Functions in Different Modes

PWM Output Function				Sawtooth Wave	Triangle Wave	Related Main Registers
Independent PWM output	Port Status Control	At startup		Support	Support	PCNA<B>R.STACA<B>
		When stopped		Support	Support	PCNA<B>R.STPCA<B>
		When overflowing		Support	Support	PCNA<B>R.OVFCA<B>
		When underflow		Support	Support	PCNA<B>R.UDFCA<B>
		When counting matches (Up Counting)		Support	Support	PCNA<B>R.CMA<B>UCA<B>
		When counting matches (Down Counting)		Support	Support	PCNA<B>R.CMA<B>DCA<B>
	Cache transfer	Period base value	Single cache	Support	Support	PERBR->PERAR
			Double buffer	Support	Support	PERCR->PERBR PERBR->PERAR
		Comparison benchmark value	Single cache	Support	Support	GCMDR->GCMBR GCMCR->GCMAR
			Double buffer	Support	Support	GCMFR->GCMDR GCMER->GCMCR GCMDR->GCMBR GCMCR->GCMAR
	EMB			Support	Support	PCNA<B>R.EMBCA<B>
Complementary PWM output	Port Status Control	At startup		Support	Support	PCNA<B>R.STACA<B>
		When stopped		Support	Support	PCNA<B>R.STPCA<B>
		When overflowing		Support	Support	PCNA<B>R.OVFCA<B>
		When underflow		Support	Support	PCNA<B>R.UDFCA<B>
		When counting matches (Up Counting)		Support	Support	PCNA<B>R.CMA<B>UCA<B>
		When counting matches (Down Counting)		Support	Support	PCNA<B>R.CMA<B>DCA<B>
	Cache transfer	Period base value	Single cache	Support	Support	PERBR->PERAR
			Double buffer	Support	Support	PERCR->PERBR PERBR->PERAR
		Comparison benchmark value	Single cache	Support	Support	GCMDR->GCMBR GCMCR->GCMAR
			Double buffer	Support	Support	GCMFR->GCMDR GCMER->GCMCR GCMDR->GCMBR GCMCR->GCMAR
		Dead zone reference value	Single cache	Support	Support	DTUBR->DTUAR DTDBR->DTDAR

PWM Output Function			Sawtooth Wave	Triangle Wave	Related Main Registers
	PWM output without dead zone		Support	Support	GCMAR=GCMBR
	PWM output with dead zone	Software method	Support	Support	GCMAR≠GCMBR
		Hardware method	not support	Support	GCMBR=GCMAR-DTUAR GCNBR=GCMAR-DTDAR
	EMB		Support	Support	PCNA<B>R.EMBCA<B>

18.4Interrupt and Events

18.4.1 Interrupt

Timer6 contains 6 universal-purpose counting compare match interrupts (including 2 capture input interrupts), 2 dedicated counting compare match interrupts, 2 count cycle match interrupts, and 1 dead-time error interrupt.

18.4.1.1 Count Comparison Match Interrupt

There are six universal-purpose comparison reference registers (GCMAR-GCMFR), which can be compared with the count value to generate a comparison match. When the count compare matches, the STFLR.CMAF~STFLR.CMFF bits in the status flag register (STFLR) will be set to 1 respectively. At this time, if the corresponding bit in INTENA~INTENF of the interrupt control register (ICONR) is set to 1 to enable interrupt, the corresponding interrupt request (TMR6_<t>_GCMA~F) will also be triggered.

The capture input action occurs when the capture input valid condition selected by the hardware capture event selection register (HCPAR, HCPBR) occurs. At this time, if the INTENA or INTENB bit of the interrupt control register (ICONR) is set to 1 to enable interrupt, the corresponding interrupt request (TMR6_<t>_GCMA~B) is triggered.

Two dedicated comparison reference registers (SCMAR~SCMBR) can also be compared with the count value to generate a comparison match. When the count compare matches, the STFLR.CMSPAF~CMSPBF bits in the status flag register (STFLR) will be set to 1 respectively. At this time, if the corresponding bit in the INTENSAU<D>or INTENSBU<D> of the interrupt control register (ICONR) is set to 1 to enable the interrupt, the corresponding interrupt request (TMR6_<t>_SCMA~B) will also be triggered.

18.4.1.2 Count Cycle Match Interrupt

When the sawtooth wave counts up to the overflow point, the sawtooth wave counts down to the underflow point, the triangle wave counts to the valley point or the triangle wave counts to the peak point, the STFLR. 1. At this time, if the ICONR.INTENOVF or ICONR.INTENUDF bit of the interrupt control register (ICONR) is set to enable interrupt, the count cycle match interrupt (TMR6_<t>_GOVF and TMR6_<t>_GUDF) can be triggered at the corresponding time point.

18.4.1.3 Dead Time Error Interrupt

When the value of the dead time reference register (DTU<D>AR) is loaded into the universal comparison reference register (GCMBR), if it exceeds the cycle limit, a dead time error will be generated, and the STFLR of the status flag register (STFLR). The DTEF bit will be set to 1. At this time, if the INTENDTE bit of the interrupt control register (ICONR) is set to enable interrupt, the dead time error interrupt (TMR6_<t>_GDTE) will be triggered at this moment.

18.4.2 Events

During the clock counting process, if a period match event (overflow or underflow point of a sawtooth wave, counting peak or valley point of a triangle wave), a Universal count comparison match event, or a special count comparison match event occurs, a corresponding event output signal will be generated to select and trigger other modules, such as ADC, DMA, etc.

The following figure shows the operation example of the universal compare match interrupt A~F && events A~F, the dedicated compare match interrupt A~B && events A~B, and the period match interrupt && events of unit 1.

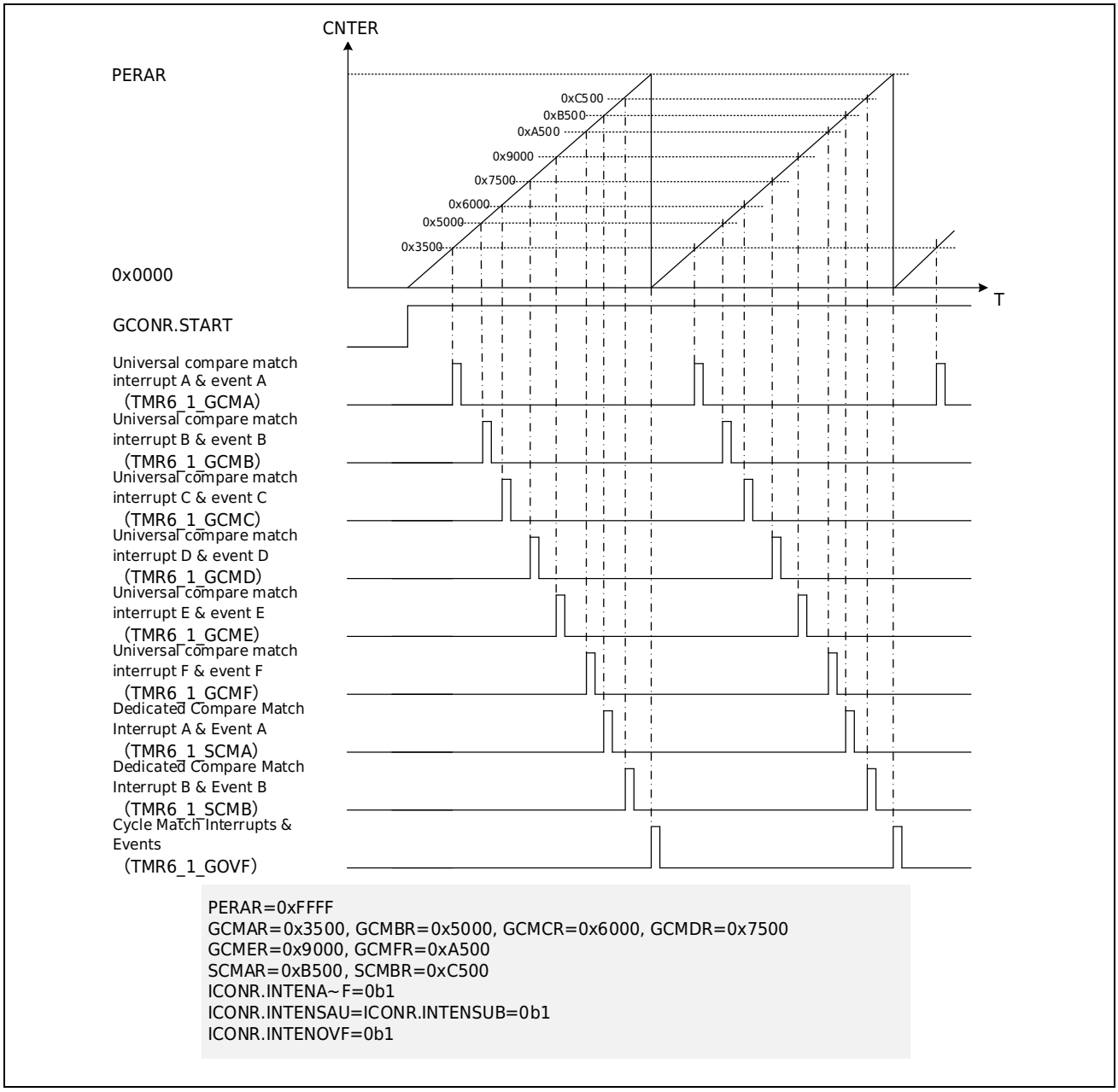


Figure 18-35 Example of Interrupt & Event Output in Sawtooth Wave Mode

18.5Application Examples

The following describes the basic settings of Timer6 related registers in several typical application situations for user reference.

18.5.1 Basic Counting and Interrupt Actions

1. Setting the Universal Period Reference Value (PERAR);
2. Set the required comparison reference value, including universal comparison reference value (GCMAR~GCMFR), special comparison reference value (SCMAR~SCMBR), etc.;
3. Set the required interrupt enable bits, including count overflow interrupt (ICONR.INTENOVF), count underflow interrupt (ICONR.INTENUDF), count match interrupt (ICONR.INTENA~F, ICONR.INTENSAU, ICONR.INTENSAD, ICONR. INTENSBU, ICONR.INTENSBD), etc.;
4. Set internal count clock frequency division (GCONR.CKDIV[3:0]);
5. Set waveform mode (GCONR.MODE);
6. Set the counting direction (only required when sawtooth wave mode GCONR.MODE=0);
7. Start counter (GCONR.START=1);

18.5.2 Comparison Output and Interrupt Action

1. Setting the Universal Period Reference Value (PERAR);
2. Set the comparison reference value of each channel, including universal comparison reference value A (GCMAR), universal comparison reference value B (GCMBR);
3. Set the required interrupt enable bit, including count overflow interrupt (ICONR.INTENOVF), count underflow interrupt (ICONR.INTENUDF), count match interrupt (ICONR.INTENA~B), etc.;
4. Set the port output status of each channel in different counting states (refer to the related control of bit17~bit0 of PCNAR or PCNBR);
5. Set internal count clock frequency division (GCONR.CKDIV[3:0]);
6. Set waveform mode (GCONR.MODE);
7. Set the counting direction (only required when sawtooth wave mode GCONR.MODE=0);
8. Set the comparison output mode of each channel (PCNAR.CAPMDA=0, PCNBR.CAPMDB=0);
9. Set each channel output enable (PCNAR.OUTENA=1, PCNBR.OUTENB=1);
10. Start counter (GCONR.START=1);

18.5.3 Capture Input and Interrupt Actions

1. Setting the Universal Period Reference Value (PERAR);
2. Set the required interrupt enable bit, including count overflow interrupt (ICONR.INTENOVF), count underflow interrupt (ICONR.INTENUDF), capture input interrupt (ICONR.INTENA~B), etc;
3. Set the capture input external conditions of each channel (refer to all valid control bits of HCPAR or HCPBR. The effective control bits are independent of each other, and multiple can be selected as the capture input conditions of a certain channel at the same time);
4. Set internal count clock frequency division (GCONR.CKDIV[3:0]);
5. Set waveform mode (GCONR.MODE);
6. Set the counting direction (only required when sawtooth wave mode GCONR.MODE=0);
7. Set capture input mode (PCNAR.CAPMDA=1, PCNBR.CAPMDB=1);
8. Start counter (GCONR.START=1);
9. Wait for the capture input condition to be generated, read the capture input value of the corresponding channel (GCMAR or GCMBR) or wait for the corresponding interrupt to be generated.

18.5.4 Buffer Transfer Action (Period Reference Value)

1. Set desired common period reference values (PERAR, PERBR, PERCR);
2. Set single and double buffer transfer mode (BCONR.BSEP);
3. Set the cache transfer time point (BCONR.BTRUP, BCONR.BTRDP, these two control bits are independent of each other and can be selected at the same time as the cache transfer time point) (this setting is only valid in triangle wave mode and invalid in sawtooth wave mode);
4. Set internal count clock frequency division (GCONR.CKDIV[3:0]);
5. Set waveform mode (GCONR.MODE);
6. Set the counting direction (only required when sawtooth wave mode GCONR.MODE=0);
7. Set the cache function to be valid (BCONR.BENP=1);
8. Start counter (GCONR.START=1);
9. Wait for the corresponding cache transmission time point, and the cache action occurs (PERBR->PERAR (when BCONR.BSEP=0), PERCR->PERBR->PERAR (when BCONR.BSEP=1)).

18.5.5 Cache Transfer Action (Common Benchmark Value)

1. Set the desired common comparison base value (GCMAR, GCMCR, GCMER, GCMBR, GCMDR, GCMFR);
2. Set the single and double buffer transmission mode of each channel (BCONR.BSEA, BCONR.BSEB);
3. Set the buffer transfer time point of each channel (BCONR.BTRUA, BCONR.BTRDA, BCONR.BTRUB, BCONR.BTRDB, the two control bits of each channel are independent of each other and can be selected at the same time as the buffer transfer time point) (this setting is only valid in triangle wave mode and invalid in sawtooth wave mode);
4. Set internal count clock frequency division (GCONR.CKDIV[3:0]);
5. Set waveform mode (GCONR.MODE);
6. Set the counting direction (only required when sawtooth wave mode GCONR.MODE=0);
7. Set the buffer function of each channel to be valid (BCONR.BENA=1, BCONR.BENB=1);
8. Start counter (GCONR.START=1);
9. Wait for the corresponding buffer transmission time point set by each channel, and the buffer action occurs (GCMCR->GCMAR (when BCONR.BSEA=0), GCMER->GCMCR->GCMAR (when BCONR.BSEA=1), GCMDR->GCMBR (when BCONR.BSEB=0), GCMFR->GCMDR->GCMBR (when BCONR.BSEB=1));

18.5.6 Cache Transfer Action (Dedicated Benchmark Value)

1. Set the required specific comparison reference value (SCMAR, SCMCR, SCMER, SCMBR, SCMDR, SCMFR);
2. Set the single and double buffer transmission mode of each channel (BCONR.BSESPA, BCONR.BSESPB);
3. Set the buffer transfer time point of each channel (BCONR.BTRUSPA, BCONR.BTRDSPA, BCONR.BTRUSPB, BCONR.BTRDSPB, the two control bits of each channel are independent of each other and can be selected at the same time as the buffer transfer time point) (this setting is only valid in triangle wave mode and invalid in sawtooth wave mode);
4. Set internal count clock frequency division (GCONR.CKDIV[3:0]);
5. Set waveform mode (GCONR.MODE);
6. Set the counting direction (only required when sawtooth wave mode GCONR.MODE=0);
7. Set the buffer function of each channel to be valid (BCONR.BENSPA=1, BCONR.BENSPB=1)
8. Start counter (GCONR.START=1);

9. Wait for the corresponding buffer transmission time point set by each channel, and the buffer action occurs (SCMCR->SCMAR (when BCONR.BSESPA=0), SCMER->SCMCR->SCMAR (when BCONR.BSESPA=1), SCMDR- >SCMBR (when BCONR.BSESPB=0), SCMFR->SCMDR->SCMBR (when BCONR.BSESPB=1))

18.5.7 Buffer Transfer Action (Deadband Reference Value)

1. Set the desired dead time reference (DTUAR, DTUBR, DTDAR, DTDBR)
2. Set the cache transfer time point (DCONR.DTBTRU, DCONR.DTBTRD, these two control bits are independent of each other and can be selected at the same time as the cache transfer time point) (this setting is only valid in triangle wave mode and invalid in sawtooth wave mode);
3. Set internal count clock frequency division (GCONR.CKDIV[3:0]);
4. Set waveform mode (GCONR.MODE);
5. Set the counting direction (only required when sawtooth wave mode GCONR.MODE=0);
6. Set the cache function to be valid (DCONR.DTBENU=1, DCONR.DTBEND=1);
7. Set the hardware dead zone function to be valid (DCONR.DTCEN=1);
8. Start counter (GCONR.START=1);
9. Wait for the corresponding cache transmission time point, and the cache action occurs (DTUBR->DTUAR, DTDBR->DTDAR).

18.5.8 Synchronous Start Action (Software Method)

1. Referto steps 1 to 6 in the [Basic Counting and Interrupt Actions] section to set up each unit that needsto be started synchronously;
2. Start the counter synchronously (set the corresponding bit of the SSTAR register to 1, and each unit corresponds to a register bit)

18.5.9 Synchronous Start Action (Hardware Method)

1. Setting the Universal Period Reference Value (PERAR);
2. Set the required comparison reference value, including universal comparison reference value (GCMAR~GCMFR), special comparison reference value (SCMAR~SCMBR), etc.;
3. Set the required interrupt enable bits, including count overflow interrupt (ICONR.INTENOVF), count underflow interrupt (ICONR.INTENUDF), count match interrupt (ICONR.INTENA~F, ICONR.INTENSAU, ICONR.INTENSAD, ICONR. INTENSBU, ICONR.INTENSBD), etc.;
4. Set the hardware startup conditions (selected by HSTAR.HSTAx, x=8~31);
5. Set hardware startup enable (HSTAR.STAS=1);

6. Repeat steps 1 to 5 above to set each unit that needs to be started synchronously (the setting of step 4 must be consistent in each unit that needs to be started synchronously);
7. Wait for the set trigger event to occur, and confirm that the counters of each unit start synchronously.

18.5.10 Quadrature Encoder Counting Action (2 Phases)

1. Setting the Universal Period Reference Value (PERAR);
2. Set the required comparison reference value, including universal comparison reference value (GCMAR~GCMFR), special comparison reference value (SCMAR~SCMBR), etc.;
3. Set the required interrupt enable bits, including count overflow interrupt (ICONR.INTENOVF), count underflow interrupt (ICONR.INTENUDF), count match interrupt (ICONR.INTENA~F, ICONR.INTENSAU, ICONR.INTENSAD, ICONR. INTENSBU, ICONR.INTENSBD), etc.;
4. Set the required hardware count-up condition (selected by HCUPR.HCUPx, x=0~7);
5. Set the required hardware down counting condition (selected by HCDOR.HCDOx, x=0~7);
6. Start counter (GCONR.START=1);
7. Wait for the set quadrature encoding counting event to be generated, and confirm that the counter counts normally

18.5.11 Quadrature Encoder Counting Action (3 Phases)

1. Refer to steps 1 to 5 in chapter [Quadrature Encoder Counting Action (2 Phases)] to set the position counting unit;
2. Set the hardware clearing condition of the position counting unit (selected by HCLRR.HCLRx, x=16~31);
3. Set position counter unit hardware clear enable (HCLRR.CLEARS=1);
4. Set the universal period reference value (PERAR) of the revolution counter unit;
5. Set the comparison reference value of revolution counting unit, including universal comparison reference value (GCMAR~GCMFR), special comparison reference value (SCMAR~SCMBR), etc.;
6. Set the required interrupt enable bits, including count overflow interrupt (ICONR.INTENOVF), count underflow interrupt (ICONR.INTENUDF), count match interrupt (ICONR.INTENA~F, ICONR.INTENSAU, ICONR.INTENSAD, ICONR. INTENSBU, ICONR.INTENSBD), etc.;
7. Set the hardware up counting condition 1 of the revolution counting unit (ZIN phase input) (selected by HCUPR.HCUPx, x=16~31, the set event here should be consistent with the event set by the position counting unit in step b);

8. Set the hardware up counting condition 2 of the revolution counting unit (the overflow event input of the position counting unit) (select the internal hardware trigger event 0 through HCUPR.HCUP8);
9. Set the hardware down counting condition of the revolution counting unit (the underflow event input of the position counting unit) (select internal hardware trigger event 1 through HCDOR.HCDO9);
10. Set the trigger source number in TMR6_TRGSEL0 to the count overflow event of the position counting unit (for details of the register content, refer to the [Timer6 Hardware Trigger Event Selection Register (TMR6_TRGSELx) (x=0~3)] section, and for the overflow event number, refer to the [Interrupt controller (INTC)] section);
11. Set the trigger source number in TMR6_TRGSEL1 to the count underflow event of the position counting unit (refer to the [Timer6 Hardware Trigger Event Selection Register (TMR6_TRGSELx) (x=0~3)] section for the register content and the [Interrupt controller (INTC)] section for the underflow event number);
12. Start revolution counting unit counter (GCONR.START=1);
13. Start position counting unit counter (GCONR.START=1);
14. Wait for the set AIN, BIN, ZIN phase count event to occur, and confirm that the counter counts normally.

18.5.12 Single PWM Output

1. Refer to the settings in steps 1-10 of the [Comparison Output and Interrupt Action] section (the output states of the two PWM channels TMR6_<t>_PWMA and TMR6_<t>_PWMB in each unit can be set independently to form two independent single PWM outputs).

18.5.13 Complementary PWM Output (Software Dead Time)

1. Setting the Universal Period Reference Value (PERAR);
2. Set universal comparison reference value A (GCMAR) and universal comparison reference value B (GCMBR);
3. Set the required interrupt enable bit, including count overflow interrupt (ICONR.INTENOVF), count underflow interrupt (ICONR.INTENUDF), count match interrupt (ICONR.INTENA~B), etc.;
4. Set the port output status in different counting states (refer to the related control of bit17~bit0 of PCNAR and PCNBR, combined with the setting values of GCMAR and GCMBR, it is necessary to ensure that the two PWM outputs form a complementary dead zone);
5. Set internal count clock frequency division (GCONR.CKDIV[3:0]);
6. Set the waveform mode to triangle wave mode (GCONR.MODE=1);

7. Set the comparison output mode (PCNAR.CAPMDA=0, PCNBR.CAPMDB=0);
8. Set output enable (PCNAR.OUTENA=1, PCNBR.OUTENB=1);
9. Start counter (GCONR.START=1);

18.5.14 Complementary PWM Output (Hardware Dead Zone)

1. Setting the Universal Period Reference Value (PERAR);
2. Set universal comparison reference value A (GCMAR), dead time reference value (DTUAR, DTDAR);
3. Set the required interrupt enable bits, including count overflow interrupt (ICONR.INTENOVF), count underflow interrupt (ICONR.INTENUDF), count match interrupt (ICONR.INTENA~B), dead zone error interrupt (ICONR.INTENDTE), etc.
4. Set the port output status in different counting states (refer to the related control of bit17~bit0 of PCNAR and PCNBR, combined with the setting values of GCMAR, DTUAR and DTDAR, it is necessary to ensure that the two PWM outputs form a complementary dead zone);
5. Set internal count clock frequency division (GCONR.CKDIV[3:0]);
6. Set the waveform mode to triangle wave mode (GCONR.MODE=1);
7. Set the comparison output mode of each channel (PCNAR.CAPMDA=0, PCNBR.CAPMDB=0);
8. Set each channel output enable (PCNAR.OUTENA=1, PCNBR.OUTENB=1);
9. Set the hardware dead zone function to be valid (DCONR.DTCEN=1);
10. Start counter (GCONR.START=1);

18.5.15 EMB Monitoring and Interrupt Action

1. Refer to steps 1-8 in section [Complementary PWM Output (Software Dead Time)] or steps 1-9 in section [Complementary PWM Output (Hardware Dead Zone)] to set the complementary PWM output action;
2. Set the state of the PWM port when the EMB event occurs (PCNAR.EMBCA, PCNBR.EMBCB) (select the corresponding protection state according to different system applications);
3. Set the time point when the PWM port resumes normal output when the EMB event becomes invalid (PCNAR.EMBRA, PCNBR.EMBRB);
4. Set and select the EMB event source input from the EMB module (PCNAR.EMBSA, PCNBR.EMBSB);
5. Set the relevant registers of the EMB module (including EMB interrupt permission register (EMB_INTEN0~3), EMB control register 1/2 (EMB_CTL1/2_0~3), EMB control PWM output

release mode selection register (EMB_RLSSEL0~3), etc.);

6. Start the counter (GCONR.START=1), and the EMB module monitors the system status in real time.

18.6Register Description

Table 18-4 Timer6 Base Address

Name	Base Address	Description
Timer6_1	0x4003 9000	Timer6_1 base address
Timer6_2	0x4003 9400	Timer6_2 base address
Timer6_3	0x4003 9800	Timer6_3 base address
Timer6_4	0x4003 9C00	Timer6_4 base address
Timer6_COMMON	0x4003 9300	Timer6 Common Register Base Address

Table 18-5 Timer6 Register List

Register	Description	Offset Address	Bit Width	Reset Value
TMR6_CNTER	Universal count register	0x0000	32	0x0000 0000
TMR6_UPDAR	Universal Refresh Value Register	0x0004	32	0x0000 0000
TMR6_PERAR	Universal purpose period reference register A	0x0040	32	0x0000 FFFF
TMR6_PERBR	Universal purpose period reference register B	0x0044	32	0x0000 FFFF
TMR6_PERCR	Universal Period Reference Register C	0x0048	32	0x0000 FFFF
TMR6_GCMAR	Universal purpose comparison reference value register A	0x0080	32	0x0000 FFFF
TMR6_GCMBR	Universal purpose comparison reference register B	0x0084	32	0x0000 FFFF
TMR6_GCMCR	Universal purpose comparison reference value register C	0x0088	32	0x0000 FFFF
TMR6_GCMDR	Universal comparison reference value register D	0x008C	32	0x0000 FFFF
TMR6_GCMER	Universal comparison reference value register E	0x0090	32	0x0000 FFFF
TMR6_GCMFR	Universal comparison reference value register F	0x0094	32	0x0000 FFFF
TMR6_SCMAR	Dedicated comparison reference value register A	0x00C0	32	0x0000 FFFF
TMR6_SCMBR	Dedicated comparison reference register B	0x00C4	32	0x0000 FFFF
TMR6_SCMCR	Dedicated comparison reference value register C	0x00C8	32	0x0000 FFFF
TMR6_SCMDR	Dedicated comparison reference value register D	0x00CC	32	0x0000 FFFF
TMR6_SCMER	Dedicated comparison reference value register E	0x00D0	32	0x0000 FFFF
TMR6_SCMFR	Dedicated comparison reference value register F	0x00D4	32	0x0000 FFFF
TMR6_DTUAR	Dead Time Reference Register UA	0x0100	32	0x0000 FFFF
TMR6_DTDAR	Dead Time Reference Register DA	0x0104	32	0x0000 FFFF
TMR6_DTUBR	Dead Time Reference Register UB	0x0108	32	0x0000 FFFF
TMR6_DTDBR	DEAD TIME REFERENCE REGISTER DB	0x010C	32	0x0000 FFFF
TMR6_GCONR	Universal control register	0x0140	32	0x0000 0002
TMR6_ICONR	Interrupt control register	0x0144	32	0x0000 0000

Register	Description	Offset Address	Bit Width	Reset Value
TMR6_BCONR	Cache control register	0x0148	32	0x0000 0000
TMR6_DCONR	Dead Band Control Register	0x014C	32	0x0000 0000
TMR6_PCNAR	Port Control Register A	0x0154	32	0x0000 0000
TMR6_PCNBR	Port Control Register B	0x0158	32	0x0000 0000
TMR6_FCNGR	Filter Control Register G	0x015C	32	0x0000 0000
TMR6_VPERR	Valid period register	0x0160	32	0x0000 0000
TMR6_STFLR	Status flag register	0x0164	32	0x8000 0000
TMR6_HSTAR	Hardware Boot Event Select Register	0x0180	32	0x0000 0000
TMR6_HSTPR	Hardware Stop Event Select Register	0x0184	32	0x0000 0000
TMR6_HCLRR	Hardware clear event select register	0x0188	32	0x0000 0000
TMR6_HUPDR	Hardware Refresh Event Selection Register	0x018C	32	0x0000 0000
TMR6_HCPAR	Hardware Capture Event Select Register A	0x0190	32	0x0000 0000
TMR6_HCPBR	Hardware Capture Event Select Register B	0x0194	32	0x0000 0000
TMR6_HCUPR	Hardware Adder Event Selection Register	0x0198	32	0x0000 0000
TMR6_HCDOR	Hardware Decrease Event Selection Register	0x019C	32	0x0000 0000

Table 18-6 Timer6 Common Register List

Register	Description	Offset Address	Bit Width	Reset Value
TMR6_FCNTR	Common Filter Control Register	0x00EC	32	0x0000 0000
TMR6_SSTAR	Common Software Synchronous Start Control register	0x00F0	32	0x0000 0000
TMR6_SSTPR	Common Software Sync Stop Control Register	0x00F4	32	0x0000 0000
TMR6_SCLRR	Common Software Synchronous Clear cControl register	0x00F8	32	0x0000 0000
TMR6_SUPDR	Common Software Synchronous Refresh Control Register	0x00FC	32	0x0000 0000

Note:

1. *The software synchronization registers (TMR6_SSTAR, TMR6_SSTPR, TMR6_SCLLR, TMR6_SUPDR) are 4 unit-independent registers and are shared by the 4 unit Timer6s.*
2. *The filter control register (TMR6_FCNTR) is a unit-independent register shared by all four units of Timer6.*

18.6.1 Universal Counter Value Register (TMR6_CNTER)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
CNT[15:0]															

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b16	Res	Reserved Bit	Read as "0"	R
b15~b0	CNT[15:0]	Count value	Current timer count **Note:** 1. In sawtooth wave upward counting mode, the initial value of CNT is set to 0; in sawtooth wave downward counting mode, the initial value of CNT is set to pear; 2. The initial value of the triangle wave mode count CNT is set to 1;	RW

18.6.2 Universal Refresh Value Register (TMR6_UPDAR)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
UPDA[15:0]															

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b16	Res	Reserved Bit	Read as "0"	R
b15~b0	UPDA[15:0]	Refresh value	Set the refresh value to be updated into the timer	RW

18.6.3 Universal Period Reference Value Register (TMR6_PER, m=A~C)

Reset value: 0x0000 FFFF

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
PERA-C[15:0]															

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b16	Res	Reserved Bit	Read as "0"	R
b15~b0	PERA-C[15:0]	Count cycle value	Set the counting cycle value and corresponding cache value of each round of counting	RW

18.6.4 Universal Comparison Reference Value Register (TMR6_GCMmR, m=A~F)

Reset value: 0x0000 FFFF

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
GCMA-F[15:0]															

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b16	Res	Reserved Bit	Read as "0"	R
b15~b0	GCMA-F[15:0]	Count baseline	The comparison reference value is set, and the matching signal is valid when it is equal to the count value	RW

18.6.5 Dedicated Comparison Reference Value Register (TMR6_SCMmR, m=A~F)

Reset value: 0x0000 FFFF

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
SCMA-F[15:0]															

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b16	Res	Reserved Bit	Read as "0"	R
b15~b0	SCMA-F[15:0]	Dedicated baseline value	Set comparison reference value and cache value	RW

18.6.6 Dead Time Reference Value Register (TMR6_DTmnR, m=D, U, n=A, B)

Reset value: 0x0000 FFFF

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
DTUA-B[15:0]/DTDA-B[15:0]															

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b16	Res	Reserved Bit	Read as "0"	R
b15~b0	DTU/DA-B[15:0]	Dead time value	Dead time setting value and cache value	RW

18.6.7 Universal Control Register (TMR6_GCONR)

Reset value: 0x0000 0002

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res												ZMSK VAL[1:0]		ZMSK POS	ZMSK REV
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res							OV STP	CKDIV[3:0]				Res	MODE	DIR	START

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b20	Res	Reserved Bit	Read at “0”, write at “0”	RW
b19~b18	ZMSKVAL[1:0]	Phase Z input muting cycle number	Quadrature encoding phase Z input masked count period value 00: Phase Z input shielding function is invalid 01: The Z-phase input is masked within 4 counting cycles after the position count overflows or underflows 10: The Z-phase input is masked within 8 counting cycles after the position count overflows or underflows 11: The Z-phase input is masked within 16 counting cycles after the position count overflows or underflows	RW
b17	ZMSKPOS	Z phase input position timer selection	0: When the Z phase is input, the unit is used as a position timer, and the position timer clear function works normally during the masking period 1: When the Z phase is input, the unit is used as a position timer, and the position timer clearing function is masked during the masking period	RW
b16	ZMSKREV	Z phase input revolution timer selection	0: When the Z phase is input, the unit acts as a revolution timer, and the revolution timer counting function works normally during the shielding period 1: When the Z phase is input, the unit is used as a revolution timer, and the counting function of the revolution timer is shielded during the shielding period	RW
b15~b9	Res	Reserved Bit	Read at “0”, write at “0”	RW
b8	OVSTP	Count overflow stop control	0: Counter continues counting after overflow or underflow 1: Counter stops counting after overflowing or underflowing	RW
b7~b4	CKDIV[2:0]	Counting clock selection	0000：PCLK0 0001：PCLK0/2 0010：PCLK0/4 0011：PCLK0/8 0100：PCLK0/16 0101：PCLK0/32 0110：PCLK0/64 0111：PCLK0/128 1000：PCLK0/256 1001：PCLK0/512 1010：PCLK0/1024	RW
b3	Res	Reserved Bit	Read at “0”, write at “0”	RW
b2	MODE	Counting mode	0: Sawtooth mode 1: Triangular wave mode	RW
b1	DIR	Counting direction	0: count down 1: count up	RW
b0	START	Timer start	0: Timer off 1: Timer Startup **Note:** This bit will automatically become 0 when the software stop condition or hardware stop condition is valid	RW

18.6.8 Interrupt Control Register (TMR6_ICONR)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res												INTENSBD	INTENSBU	INTENSAD	INTENSAU
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res							INTENDTE	INTENUDF	INTENOVF	INTENF	INTENE	INTEND	INTENC	INTENB	INTENA

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b20	Res	Reserved Bit	Read at “0”, write at “0”	RW
b19	INTENSBD	Dedicated down count interrupt enable B	0: During down counting, when the SCMBR register is equal to the count value, the interrupt is invalid 1: During down counting, when the SCMBR register is equal to the count value, the interrupt is enabled	RW
b18	INTENSBU	Dedicated up-counting interrupt enable B	0: During up-counting, when the SCMBR register is equal to the count value, the interrupt is invalid 1: During up-counting, when the SCMBR register is equal to the count value, the interrupt is enabled	RW
b17	INTENSAD	Dedicated down count interrupt enable A	0: During down counting, when the SCMAR register is equal to the count value, the interrupt is invalid 1: During down counting, when the SCMAR register is equal to the count value, the interrupt is enabled	RW
b16	INTENSAU	Dedicated up-counting interrupt enable A	0: During up-counting, when the SCMAR register is equal to the count value, the interrupt is invalid 1: During up-counting, when the SCMAR register is equal to the count value, the interrupt is enabled	RW
b15~b9	Res	Reserved Bit	Read at “0”, write at “0”	RW
b8	INTENDTE	Dead time error interrupt enable	0: When the dead time is wrong, the interrupt is invalid 1: When the dead time is wrong, the interrupt is enabled	RW
b7	INTENUDF	Underflow interrupt enable	0: When the counter underflows, the interrupt is invalid. 1: When the counter underflows, the interrupt is enabled	RW
b6	INTENOVF	Overflow interrupt enable	0: When the counter overflows, the interrupt is invalid. 1: When the counter overflows, the interrupt is enabled	RW
b5	INTENF	Count match interrupt enable F	0: When the GCMFR register is equal to the count value, the interrupt is invalid 1: When the GCMFR register is equal to the count value, the interrupt is enabled	RW
b4	INTENE	Count match interrupt enable E	0: When the GCMER register is equal to the count value, the interrupt is invalid 1: When the GCMER register is equal to the count value, the interrupt is enabled	RW
b3	INTEND	Count match interrupt enable D	0: When the GCMDR register is equal to the count value, the interrupt is invalid 1: When the GCMDR register is equal to the count value, the interrupt is enabled	RW
b2	INTENC	Count match interrupt enable C	0: When the GCMCR register is equal to the count value, the interrupt is invalid 1: When the GCMCR register is equal to the count value, the interrupt is enabled	RW
b1	INTENB	Count match interrupt enable B	0: The interrupt is invalid when the CMPARregister is equal to the count value or when a capture input event occurs 1: This interrupt is enabled when the CMPARregister is equal to the count value, or when a capture input event occurs	RW
b0	INTENA	Count Match interrupt enable A	0: The interrupt is invalid when the CMPARregister is equal to the count value or when a capture input event occurs 1: This interrupt is enabled when the CMPARregister is equal to the count value, or when a capture input event occurs	RW

18.6.9 Cache Control Register (TMR6_BCONR)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res								BTRDSPB	BTRUSPB	BSESPB	BENSPB	BTRDSPA	BTRUSPA	BSESPA	BENSPA
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res				BTRDP	BTRUP	BSEP	BENP	BTRDB	BTRUB	BSEB	BENB	BTRDA	BTRUA	BSEA	BENA

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b24	Res	Reserved Bit	Read at “0”, write at “0”	RW
b23	BTRDSPB	Dedicated comparison reference value buffer transfer time setting DB	0: Cached value does not pass when triangulation mode counts to valley point 1: When the triangle wave counting mode counts to the valley point, a buffer value transfer occurs **Note:** In sawtooth wave mode, this bit control is not required	RW
b22	BTRUSPB	Dedicated comparison reference value buffer transfer time setting UB	0: Cached value does not pass when triangulation mode counts to peak. 1: When the triangle wave counting mode reaches the peak, a buffer value transfer occurs **Note:** In sawtooth wave mode, this bit control is not required	RW
b21	BSESPB	Dedicated Baseline Value Cache Transfer Option B	0: Single buffer transfer (SCMDR->SCMBR) 1: Double buffer transfer (SCMFR->SCMDR->SCMBR)	RW
b20	BENSPB	Dedicated baseline cache transfer B	0: Buffer transmission is invalid 1: Buffer transfer enable	RW
b19	BTRDSPA	Dedicated comparison reference value buffer transfer time setting DA	0: Cached value does not pass when triangulation mode counts to valley point 1: When the triangle wave counting mode counts to the valley point, a buffer value transfer occurs **Note:** In sawtooth wave mode, this bit control is not required	RW
b18	BTRUSPA	Dedicated comparison reference value buffer transfer time setting UA	0: Cached value does not pass when triangulation mode counts to peak. 1: When the triangle wave counting mode reaches the peak, a buffer value transfer occurs **Note:** In sawtooth wave mode, this bit control is not required	RW
b17	BSESPA	Dedicated baseline value buffer transfer option A	0: single buffer transfer (SCMCR->SCMAR) 1: Double buffer transfer (SCMER->SCMCR->SCMAR)	RW
b16	BENSPA	Dedicated baseline cache transfer A	0: Buffer transmission is invalid 1: Buffer transfer enable	RW
b15~b12	Res	Reserved Bit	Read at “0”, write at “0”	RW
b11	BTRDP	Cycle value buffer transfer time setting D	0: Cached value does not pass when triangulation mode counts to valley point 1: When the triangle wave counting mode counts to the valley point, a buffer value transfer occurs **Note:** In sawtooth wave mode, this bit control is not required	RW
b10	BTRUP	Cycle value buffer transfer time setting U	0: Cached value does not pass when triangulation mode counts to peak. 1: When the triangle wave counting mode reaches the peak, a buffer value transfer occurs **Note:** In sawtooth wave mode, this bit control is not required	RW
b9	BSEP	Period value buffer transfer selection	0: single buffer transfer (PERBR->PERAR) 1: Double buffer transfer (PERCR->PERBR->PERAR) **Note:** The transmission time point has nothing to do with the counting mode, only at the overflow point, underflow point of the sawtooth wave or the trough of the triangular wave	RW
b8	BENP	Period value buffer transfer	0: Buffer transmission is invalid 1: Buffer transfer enable	RW
b7	BTRDB	Universal comparison value buffer transfer time setting DB	0: Cached value does not pass when triangulation mode counts to valley point 1: When the triangle wave counting mode counts to the valley point, a buffer value transfer occurs	RW

			Note: In sawtooth wave mode, this bit control is not required	
b6	BTRUB	Universal comparison value buffer transfer time setting UB	0: Cached value does not pass when triangulation mode counts to peak. 1: When the triangle wave counting mode reaches the peak, a buffer value transfer occurs **Note:** In sawtooth wave mode, this bit control is not required	RW
b5	BSEB	Universal comparison value buffer transfer option B	When comparing output functions: 0: single buffer transfer (GCMDR->GCMBR) 1: Double buffer transfer (GCMFR->GCMDR->GCMBR) When capturing the input function: 0: single buffer transfer (GCMBR->GCMDR) 1: Double buffer transfer (GCMBR->GCMDR->GCMFR)	RW
b4	BENB	Universal comparison value buffer transfer B	0: Buffer transmission is invalid 1: Buffer transfer enable	RW
b3	BTRDA	Universal comparison value buffer transfer time setting DA	0: Cached value does not pass when triangulation mode counts to valley point 1: When the triangle wave counting mode counts to the valley point, a buffer value transfer occurs **Note:** In sawtooth wave mode, this bit control is not required	RW
b2	BTRUA	Universal comparison value buffer transfer time setting UA	0: Cached value does not pass when triangulation mode counts to peak. 1: When the triangle wave counting mode reaches the peak, a buffer value transfer occurs **Note:** In sawtooth wave mode, this bit control is not required	RW
b1	BSEA	Universal comparison value buffer transfer option A	When comparing output functions: 0: single buffer transfer (GCMCR->GCMAR) 1: Double buffer transfer (GCMER->GCMCR->GCMAR) When capturing the input function: 0: single buffer transfer (GCMAR->GCMCR) 1: Double buffer transfer (GCMAR->GCMCR->GCMER)	RW
b0	BENA	Universal comparison value buffer transfer A	0: Buffer transmission is invalid 1: Buffer transfer enable	RW

18.6.10 Dead Band Control Register (TMR6_DCONR)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res								DTB TRD	DTB TRU	DTB END	DTB ENU	Res		SEPA	DTC EN

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b8	Res	Reserved Bit	Read at “0”, write at “0”	RW
b7	DTBTRD	Dead time value buffer transfer time setting D	0: When the triangle wave counting mode counts to the valley point, the dead zone buffer value is not transmitted 1: When the triangle wave counting mode counts to the valley point, a dead zone buffer value transfer occurs **Note:** In sawtooth wave mode, this bit control is not required	RW
b6	DTBTRU	Dead time value buffer transfer time setting U	0: When the triangle wave counting mode counts to the peak point, the dead zone buffer value is not transmitted 1: When the triangle wave counting mode counts to the peak point, a dead zone buffer value transfer occurs **Note:** In sawtooth wave mode, this bit control is not required	RW
b5	DTBEND	Dead Time Value Cache Transfer D	0: Buffer transmission is invalid 1: Buffer transfer enable (DTDBR->DTDAR)	RW
b4	DTBENU	Dead time value cache transfer U	0: Buffer transmission is invalid 1: Buffer transfer enable (DTUBR->DTUAR)	RW
b3~b2	Res	Reserved Bit	Read at “0”, write at “0”	RW
b1	SEPA	Separate settings	0: DTUAR and DTDAR are set separately 1: The value of DTDAR is automatically equal to the value of DTUAR	RW
b0	DTCEN	Dead zone function	0: Dead zone function is invalid 1: The dead zone function is valid	RW

18.6.11 Port Control Register A (TMR6_PCNAR)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
CAPMDA	Res		OUTENA	Res		EMBSA[1:0]		EMBRA[1:0]		EMBCA[1:0]		Res		FORCA[1:0]	
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
CMBDCA[1:0]		CMBUCA[1:0]		CMADCA[1:0]		CMAUCA[1:0]		UDFCA[1:0]		OVFCA[1:0]		STPCA[1:0]		STACA[1:0]	

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31	CAPMDA	Functional mode selection A	0: Comparison output function 1: Capture input function	RW
b30~b29	Res	Reserved Bit	Read at “0”, write at “0”	RW
b28	OUTENA	Output enable A	0: TMR6_<t>_PWMA port output invalid when Timer6 is in function 1: TMR6_<t>_PWMA port output valid when Timer6 function is in progress	RW
b27~b26	Res	Reserved Bit	Read at “0”, write at “0”	RW
b25~b24	EMBSA[1:0]	When an EMB event occurs, the port status is set to A	00: Select EMB Group6 control event to be valid 01: Select the Group7 control event of EMB to be valid Other values are not allowed to be set	RW
b23~b22	EMBRA[1:0]	Select A for EMB release mode	00: When the EMB control event selected by EMBSA[1:0] invalid, the TMR6_<t>_PWMA port is released immediately (One Shot) 01: When the EMB control event selected by EMBSA[1:0] invalid, the TMR6_<t>_PWMA port is released when the counter counts to overflow (Cycle By Cycle 1) 10: When the EMB control event selected by EMBSA[1:0] invalid, the TMR6_<t>_PWMA port is released when the counter counts to underflow (Cycle By Cycle 2) 11: When the EMB control event selected by EMBSA[1:0] invalid, the TMR6_<t>_PWMA port is released when the counter counts to overflow or underflow (Cycle By Cycle 3)	RW
b21~b20	EMBCA[1:0]	EMB event port status set to A	00: When EMBSA[1:0] selects EMB control event, TMR6_<t>_PWMA port outputs normally 01: When EMBSA[1:0] selects EMB control event, TMR6_<t>_PWMA port outputs high impedance state 10: When EMBSA[1:0] selects EMB control event, TMR6_<t>_PWMA port outputs low level 11: When EMBSA[1:0] selects EMB control event, TMR6_<t>_PWMA port outputs high level	RW
b19~b18	Res	Reserved Bit	Read at “0”, write at “0”	RW
b17~b16	FORCA[1:0]	Force port status setting A	0x: Set invalid 10: The next cycle starts, and the TMR6_<t>_PWMA port output is set to a low level 11: The next cycle starts, and the TMR6_<t>_PWMA port output is set to a high level **Note 1:** The next cycle refers to the hardware counting mode or the zigzag wave counting to the overflow point or underflow point, the triangular wave counting to the valley point **Note 2:** This register bit can be used to achieve 0% or 100% control of PWM output duty cycle.	RW
b15~b14	CMBDCA[1:0]	Count down && when comparison value B matches, port state is set to A	00: During down counting, when the timer count value is equal to GCMBR, the TMR6_<t>_PWMA port output is set to low level 01: During down counting, when the timer count value is equal to GCMBR, the TMR6_<t>_PWMA port output is set to high level 10: During down counting, when the timer count value is equal to GCMBR, the TMR6_<t>_PWMA port output maintains the previous state 11: During down counting, when the timer count value is equal to GCMBR, the TMR6_<t>_PWMA port output is set to the inverted level	RW
b13~b12	CMBUCA[1:0]	Count up && when comparison value B matches, port state is set to A	00: During the up counting period, when the timer count value is equal to GCMBR, the TMR6_<t>_PWMA port output is set to low level 01: During the up counting period, when the timer count value is equal to GCMBR, the TMR6_<t>_PWMA port output is set to high	RW

			level 10: During up counting, when the timer count value is equal to GCMBR, the TMR6_<t>_PWMA port output maintains the previous state 11: During the up counting period, when the timer count value is equal to GCMBR, the TMR6_<t>_PWMA port output is set to the inverted level	
b11~b10	CMADCA[1:0]	Count down && When comparison value A matches, port state is set to A	00: During down counting, when the timer count value is equal to GCMAR, the TMR6_<t>_PWMA port output is set to low level 01: During down counting, when the timer count value is equal to GCMAR, the TMR6_<t>_PWMA port output is set to high level 10: During down counting, when the timer count value is equal to GCMAR, the TMR6_<t>_PWMA port output maintains the previous state 11: During down counting, when the timer count value is equal to GCMAR, the TMR6_<t>_PWMA port output is set to the inverted level	RW
b9~b8	CMAUCA[1:0]	Count up && When the comparison value A matches, the port state is set to A	00: During the up counting period, when the timer count value is equal to GCMAR, the TMR6_<t>_PWMA port output is set to low level 01: During the up counting period, when the timer count value is equal to GCMAR, the TMR6_<t>_PWMA port output is set to high level 10: During up counting, when the timer count value is equal to GCMAR, the TMR6_<t>_PWMA port output maintains the previous state 11: During the up counting period, when the timer count value is equal to GCMAR, the TMR6_<t>_PWMA port output is set to the inverted level	RW
b7~b6	UDFCA[1:0]	Count underflow port status setting A	00: When the count underflows, the TMR6_<t>_PWMA port output is set to low level 01: When the count underflows, the TMR6_<t>_PWMA port output is set to high level 10: When the count underflows, the TMR6_<t>_PWMA port output maintains the previous state 11: When the count underflows, the TMR6_<t>_PWMA port output is set to the inverted level	RW
b5~b4	OVFCA[1:0]	Count overflow port status setting A	00: When the count overflows, the TMR6_<t>_PWMA port output is set to a low level 01: When the count overflows, the TMR6_<t>_PWMA port output is set to high level 10: When the count overflows, the TMR6_<t>_PWMA port output maintains the previous state 11: When the count overflows, the TMR6_<t>_PWMA port output is set to the inverted level	RW
b3~b2	STPCA[1:0]	Count stop port state setting A	00: When counting stops, the TMR6_<t>_PWMA port output is set to low level 01: When counting stops, the TMR6_<t>_PWMA port output is set to high level 10: When counting stops, the TMR6_<t>_PWMA port output maintains the previous state 11: When counting stops, the TMR6_<t>_PWMA port output maintains the previous state	RW
b1~b0	STACA[1:0]	Count start port status setting A	00: When counting starts, the TMR6_<t>_PWMA port output is set to low level 01: When counting starts, the TMR6_<t>_PWMA port output is set to high level 10: When counting starts, the TMR6_<t>_PWMA port output maintains the previous state 11: When counting starts, the TMR6_<t>_PWMA port output maintains the previous state	RW

18.6.12 Port Control Register B (TMR6_PCNBR)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
CAPMDB	Res		OUTENB	Res		EMBSB[1:0]		EMBRB[1:0]		EMBCB[1:0]		Res		FORCB[1:0]	

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
CMBDCB[1:0]		CMBUCB[1:0]		CMADCB[1:0]		CMAUCB[1:0]		UDFCB[1:0]		OVFCB[1:0]		STPCB[1:0]		STACB[1:0]	

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31	CAPMDB	Function mode selection B	0: Comparison output function 1: Capture input function	RW
b30~b29	Res	Reserved Bit	Read at “0”, write at “0”	RW
b28	OUTENB	Output enable B	0: TMR6_<t>_PWMB port output invalid when Timer6 is in function 1: TMR6_<t>_PWMB port output valid when Timer6 function is in progress	RW
b27~b26	Res	Reserved Bit	Read at “0”, write at “0”	RW
b25~b24	EMBSB[1:0]	When an EMB event occurs, the port status is set to B	00: Select EMB Group6 control event to be valid 01: Select the Group7 control event of EMB to be valid Other values are not allowed to be set	RW
b23~b22	EMBRB[1:0]	EMB release mode selection B	00: When the EMB control event selected by EMBSB[1:0] invalid, the TMR6_<t>_PWMB port is released immediately (One Shot) 01: When the EMB control event selected by EMBSB[1:0] invalid, the TMR6_<t>_PWMB port is released when the counter counts to overflow (Cycle By Cycle 1) 10: When the EMB control event selected by EMBSB[1:0] invalid, the TMR6_<t>_PWMB port is released when the counter counts to underflow (Cycle By Cycle 2) 11: When the EMB control event selected by EMBSB[1:0] invalid, the TMR6_<t>_PWMB port is released when the counter counts to overflow or underflow (Cycle By Cycle 3)	RW
b21~b20	EMBCB[1:0]	Port status setting B during EMB event	00: When EMBSB[1:0] selects EMB control event, TMR6_<t>_PWMB port outputs normally 01: When EMBSB[1:0] selects EMB control event, TMR6_<t>_PWMB port outputs high impedance state 10: When EMBSB[1:0] selects EMB control event, TMR6_<t>_PWMB port outputs low level 11: When EMBSB[1:0] selects EMB control event, TMR6_<t>_PWMB port outputs high level	RW
b19~b18	Res	Reserved Bit	Read at “0”, write at “0”	RW
b17~b16	FORCB[1:0]	Forced port status setting B	0x: Set invalid 10: The next cycle starts, and the TMR6_<t>_PWMB port output is set to a low level 11: The next cycle starts, and the TMR6_<t>_PWMB port output is set to a high level **Note 1:** The next cycle refers to the hardware counting mode or the zigzag wave counting to the overflow point or underflow point, the triangular wave counting to the valley point **Note 2:** This register bit can be used to achieve 0% or 100% control of PWM output duty cycle.	RW
b15~b14	CMBDCB[1:0]	Count down && compare value B matches when port state is set to B	00: During down counting, when the timer count value is equal to GCMBR, the TMR6_<t>_PWMB port output is set to low level 01: During down counting, when the timer count value is equal to GCMBR, the TMR6_<t>_PWMB port output is set to high level 10: During down counting, when the timer count value is equal to GCMBR, the TMR6_<t>_PWMB port output maintains the previous state 11: During down counting, when the timer count value is equal to GCMBR, the TMR6_<t>_PWMB port output is set to the inverted level	RW
b13~b12	CMBUCB[1:0]	Count up && when comparison value B matches, port state is set to B	00: During the up counting period, when the timer count value is equal to GCMBR, the TMR6_<t>_PWMB port output is set to low level 01: During the up counting period, when the timer count value is equal to GCMBR, the TMR6_<t>_PWMB port output is set to high	RW

			level 10: During up counting, when the timer count value is equal to GCMBR, the TMR6_<t>_PWMB port output maintains the previous state 11: During the up counting period, when the timer count value is equal to GCMBR, the TMR6_<t>_PWMB port output is set to the inverted level	
b11~b10	CMADCB[1:0]	Count down && when comparison value A matches, port state is set to B	00: During down counting, when the timer count value is equal to GCMAR, the TMR6_<t>_PWMB port output is set to low level 01: During down counting, when the timer count value is equal to GCMAR, the TMR6_<t>_PWMB port output is set to high level 10: During down counting, when the timer count value is equal to GCMAR, the TMR6_<t>_PWMB port output maintains the previous state 11: During down counting, when the timer count value is equal to GCMAR, the TMR6_<t>_PWMB port output is set to the inverted level	RW
b9~b8	CMAUCB[1:0]	Count up && when comparison value A matches, port state is set to B	00: During the up counting period, when the timer count value is equal to GCMAR, the TMR6_<t>_PWMB port output is set to low level 01: During the up counting period, when the timer count value is equal to GCMAR, the TMR6_<t>_PWMB port output is set to high level 10: During up counting, when the timer count value is equal to GCMAR, the TMR6_<t>_PWMB port output maintains the previous state 11: During the up counting period, when the timer count value is equal to GCMAR, the TMR6_<t>_PWMB port output is set to the inverted level	RW
b7~b6	UDFCB[1:0]	Count underflow port status setting B	00: When the count underflows, the TMR6_<t>_PWMB port output is set to a low level 01: When the count underflows, the TMR6_<t>_PWMB port output is set to high level 10: When the count underflows, the TMR6_<t>_PWMB port output maintains the previous state 11: When the count underflows, the TMR6_<t>_PWMB port output is set to the inverted level	RW
b5~b4	OVFCB[1:0]	Count overflow port status setting B	00: When the count overflows, the TMR6_<t>_PWMB port output is set to a low level 01: When the count overflows, the TMR6_<t>_PWMB port output is set to high level 10: When the count overflows, the TMR6_<t>_PWMB port output maintains the previous state 11: When the count overflows, the TMR6_<t>_PWMB port output is set to the inverted level	RW
b3~b2	STPCB[1:0]	Count stop port status setting B	00: When counting stops, the TMR6_<t>_PWMB port output is set to low level 01: When counting stops, the TMR6_<t>_PWMB port output is set to high level 10: When counting stops, the TMR6_<t>_PWMB port output maintains the previous state 11: When counting stops, the TMR6_<t>_PWMB port output maintains the previous state	RW
b1~b0	STBCB[1:0]	Count start port status setting B	00: When counting starts, the TMR6_<t>_PWMB port output is set to low level 01: When counting starts, the TMR6_<t>_PWMB port output is set to high level 10: When counting starts, the TMR6_<t>_PWMB port output maintains the previous state 11: When counting starts, the TMR6_<t>_PWMB port output maintains the previous state	RW

18.6.13 Filter Control Register (TMR6_FCNGR)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res									NOFI CKGB[1:0]		NOFI ENGB	Res	NOFI CKGA[1:0]		NOFI ENGA

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b7	Res	Reserved Bit	Read at “0”, write at “0”	RW
b6~b5	NOFICKGB[1:0]	Filter sampling reference clock selection GB	00：PCLK0 01：PCLK0/4 10：PCLK0/16 11：PCLK0/64	RW
b4	NOFIENGB	Capture input port filter GB	0: The filter function of the TMR6_<t>_PWMB input port of this unit invalid 1: The unit TMR6_<t>_PWMB input port filter function is enabled	RW
b3	Res	Reserved Bit	Read at “0”, write at “0”	RW
b2~b1	NOFICKGA[1:0]	Filter sampling reference clock selection GA	00：PCLK0 01：PCLK0/4 10：PCLK0/16 11：PCLK0/64	RW
b0	NOFIENGA	Capture input port filter GA	0: The filter function of the TMR6_<t>_PWMA input port of this unit invalid 1: The unit TMR6_<t>_PWMA input port filter function is enabled	RW

18.6.14 Valid Period Register (TMR6_VPERR)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res											PCNTS[2:0]			PCNTE[1:0]	
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res						SP PERIB	SP PERIA	Res							

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b21	Res	Reserved Bit	Read at “0”, write at “0”	RW
b20~b18	PCNTS[2:0]	Valid period selection	000: Valid cycle selection function is invalid 001: Valid every 1 cycle 010: Valid every 2 cycles 011: Valid every 3 cycles 100: Valid every 4 cycles 101: Valid every 5 cycles 110: Valid every 6 cycles 111: Valid every 7 cycles	RW
b17~b16	PCNTE[1:0]	Active cycle count condition selection	00: Valid period selection function is invalid 01: The sawtooth wave counts the upper and lower overflow points or the triangular wave trough as the counting condition 10: The sawtooth wave counts the upper and lower overflow points or the triangular wave peak as the counting condition 11: The sawtooth wave counts the upper and lower overflow points or the triangular wave trough and peak as the counting condition	RW
b15~b10	Res	Reserved Bit	Read at “0”, write at “0”	RW
b9	SPPERIB	Dedicated signal effective period selection B	0: Valid cycle selection function is invalid 1: Valid cycle selection function is enabled	RW
b8	SPPERIA	Special signal effective period selection A	0: Valid cycle selection function is invalid 1: Valid cycle selection function is enabled	RW
b7~b0	Res	Reserved Bit	Read at “0”, write at “0”	RW

18.6.15 Status Flag Register (TMR6_STFLR)

Reset value: 0x8000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
DIRF	Res			CMBF2	CMAF2	Res		VPERNUM[2:0]			Res				
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res			CMS BDF	CMS BUF	CMS ADF	CMS AUF	DTEF	UDFF	OVFF	CMFF	CMEF	CMDF	CMCF	CMBF	CMAF

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31	DIRF	Counting direction	0: count down 1: count up	R
b30-b28	Res	Reserved Bit	Read at “0”, write at “0”	R
b27	CMBF2	Capture B again	0: No TMR6_<t>_PWMB capture completion action occurs after the count match B flag CMBF is 1 1: After the count match B flag CMBF is 1, TMR6_<t>_PWMB capture completion action occurs again	RW
b26	CMAF2	Capture A again	0: No TMR6_<t>_PWMA capture completion action occurs after the count match A flag CMAF is 1 1: After the count match A flag CMAF is 1, the TMR6_<t>_PWMA capture completion action occurs again	RW
b25-b24	Res	Reserved Bit	Read at “0”, write at “0”	R
b23~b21	VPERNUM[2:0]	Number of cycles	When the effective cycle selection function is enabled, the number of cycles after counting	R
b20~b13	Res	Reserved Bit	Read at “0”, write at “0”	R
b12	CMSBDF	Count down special reference value match B	0: When counting down, the value of the SCMBR register is not equal to the count value 1: When counting down, the value of the SCMBR register is equal to the count value	RW
b11	CMSBUF	Count up special reference value match B	0: When counting up, the value of the SCMBR register is not equal to the count value 1: When counting up, the value of the SCMBR register is equal to the count value	RW
b10	CMSADF	Count down special base value match A	0: When counting down, the value of the SCMAR register is not equal to the count value 1: When counting down, the value of the SCMAR register is equal to the count value	RW
b09	CMSAUF	Up-Count-Specific Baseline Match A	0: When counting up, the value of the SCMAR register is not equal to the count value 1: When counting up, the value of the SCMAR register is equal to the count value	RW
b8	DTEF	Dead time error	0: no dead time error occurred 1: A dead time error has occurred	R
b7	UDFF	Underflow match	0: Sawtooth underflow does not occur or triangle wave counts to the valley point 1: A sawtooth wave underflow occurs or a triangular wave counts to a valley point	RW
b6	OVFF	Overflow match	0: No sawtooth overflow or triangular wave counting to peak 1: Sawtooth overflow occurs or triangular wave counts to peak	RW
b5	CMFF	Count match F	0: The value of the GCMFR register is not equal to the count value 1: The value of the GCMFR register is equal to the count value	RW
b4	CMEF	count match E	0: The value of the GCMER register is not equal to the count value 1: The value of the GCMER register is equal to the count value	RW
b3	CMDF	Count match D	0: The value of the GCMDR register is not equal to the count value 1: The value of the GCMDR register is equal to the count value	RW
b2	CMCF	Count match C	0: The value of the GCMCR register is not equal to the count value 1: The value of the GCMCR register is equal to the count value	RW
b1	CMBF	Count match B	0: The value of the GCMBR register is not equal to the count value, and TMR6_<t>_PWMB capture completion action has not occurred 1: The value of the GCMBR register is equal to the count value, or TMR6_<t>_PWMB capture completion occurs	RW
b0	CMAF	Count Match A	0: The value of the GCMAR register is not equal to the count value, and TMR6_<t>_PWMA capture completion action has not occurred 1: The value of the GCMAR register is equal to the count value, or TMR6_<t>_PWMA capture completion occurs	RW

18.6.16 Hardware Start Event Select Register (TMR6_HSTAR)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res								HSTA 23	HSTA 22	HSTA 21	HSTA 20	HSTA 19	HSTA 18	HSTA 17	HSTA 16
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res				HSTA 11	HSTA 10	HSTA 9	HSTA 8	STAS	Res			HSTA 3	HSTA 2	HSTA 1	HSTA 0

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b24	Res	Reserved Bit	Read at “0”, write at “0”	RW
b23	HSTA23	Hardware startup condition 23	Condition: TMR6_TRIGD port is sampled to the falling edge 0: Hardware startup is invalid when conditions match 1: Hardware startup is valid when conditions match	RW
b22	HSTA22	Hardware startup condition 22	Condition: TMR6_TRIGD port is sampled to the rising edge 0: Hardware startup is invalid when conditions match 1: Hardware startup is valid when conditions match	RW
b21	HSTA21	Hardware startup condition 21	Condition: TMR6_TRIGC port is sampled to the falling edge 0: Hardware startup is invalid when conditions match 1: Hardware startup is valid when conditions match	RW
b20	HSTA20	Hardware startup condition 20	Condition: TMR6_TRIGC port is sampled to the rising edge 0: Hardware startup is invalid when conditions match 1: Hardware startup is valid when conditions match	RW
b19	HSTA19	Hardware startup condition 19	Condition: TMR6_TRIGB port is sampled to the falling edge 0: Hardware startup is invalid when conditions match 1: Hardware startup is valid when conditions match	RW
b18	HSTA18	Hardware startup condition 18	Condition: TMR6_TRIGB port is sampled to the rising edge 0: Hardware startup is invalid when conditions match 1: Hardware startup is valid when conditions match	RW
b17	HSTA17	Hardware startup condition 17	Condition: TMR6_TRIGA port is sampled to the falling edge 0: Hardware startup is invalid when conditions match 1: Hardware startup is valid when conditions match	RW
b16	HSTA16	Hardware startup condition 16	Condition: TMR6_TRIGA port is sampled to the rising edge 0: Hardware startup is invalid when conditions match 1: Hardware startup is valid when conditions match	RW
b15~b12	Res	Reserved Bit	Read at “0”, write at “0”	RW
b11	HSTA11	Hardware startup condition 11	Condition: Internal Hardware Trigger Event 3 Valid 0: Hardware startup is invalid when conditions match 1: Hardware startup is valid when conditions match	RW
b10	HSTA10	Hardware startup condition 10	Condition: Internal Hardware Trigger Event 2 Valid 0: Hardware startup is invalid when conditions match 1: Hardware startup is valid when conditions match	RW
b9	HSTA9	Hardware startup condition 9	Condition: Internal Hardware Trigger Event 1Valid 0: Hardware startup is invalid when conditions match 1: Hardware startup is valid when conditions match	RW
b8	HSTA8	Hardware startup condition 8	Condition: Internal Hardware Trigger Event 0Valid 0: Hardware startup is invalid when conditions match 1: Hardware startup is valid when conditions match	RW
b7	STAS	Hardware enable	0: hardware startup is invalid 1: Hardware startup is valid **Note:** When hardware startup is enabled, the SSTAR setting is invalid.	RW
b6~b4	Res	Reserved Bit	Read at “0”, write at “0”	RW
b3	HSTA3	Hardware startup condition 3	Condition: TMR6_<t>_PWMB port is sampled to the falling edge 0: Hardware startup is invalid when conditions match 1: Hardware startup is valid when conditions match	RW
b2	HSTA2	Hardware startup condition 2	Condition: TMR6_<t>_PWMB port is sampled to the rising edge 0: Hardware startup is invalid when conditions match 1: Hardware startup is valid when conditions match	RW
b1	HSTA1	Hardware startup condition 1	Condition: TMR6_<t>_PWMA port is sampled to the falling edge 0: Hardware startup is invalid when conditions match 1: Hardware startup is valid when conditions match	RW
b0	HSTA0	Hardware startup condition 0	Condition: TMR6_<t>_PWMA port is sampled to the rising edge 0: Hardware startup is invalid when conditions match 1: Hardware startup is valid when conditions match	RW

18.6.17 Hardware Stop Event Select Register (TMR6_HSTPR)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res								HSTP 23	HSTP 22	HSTP 21	HSTP 20	HSTP 19	HSTP 18	HSTP 17	HSTP 16
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res				HSTP 11	HSTP 10	HSTP 9	HSTP 8	STPS	Res			HSTP 3	HSTP 2	HSTP 1	HSTP 0

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b24	Res	Reserved Bit	Read at “0”, write at “0”	RW
b23	HSTP23	Hardware stop condition 23	Condition: TMR6_TRIGD port is sampled to the falling edge 0: Hardware stop invalid when condition matches 1: Hardware stops working when conditions match	RW
b22	HSTP22	Hardware stop condition 22	Condition: TMR6_TRIGD port is sampled to the rising edge 0: Hardware stop invalid when condition matches 1: Hardware stops working when conditions match	RW
b21	HSTP21	Hardware stop condition 21	Condition: TMR6_TRIGC port is sampled to the falling edge 0: Hardware stop invalid when condition matches 1: Hardware stops working when conditions match	RW
b20	HSTP20	Hardware stop condition 20	Condition: TMR6_TRIGC port is sampled to the rising edge 0: Hardware stop invalid when condition matches 1: Hardware stops working when conditions match	RW
b19	HSTP19	Hardware stop condition 19	Condition: TMR6_TRIGB port is sampled to the falling edge 0: Hardware stop invalid when condition matches 1: Hardware stops working when conditions match	RW
b18	HSTP18	Hardware stop condition 18	Condition: TMR6_TRIGB port is sampled to the rising edge 0: Hardware stop invalid when condition matches 1: Hardware stops working when conditions match	RW
b17	HSTP17	Hardware stop condition 17	Condition: TMR6_TRIGA port is sampled to the falling edge 0: Hardware stop invalid when condition matches 1: Hardware stops working when conditions match	RW
b16	HSTP16	Hardware stop condition 16	Condition: TMR6_TRIGA port is sampled to the rising edge 0: Hardware stop invalid when condition matches 1: Hardware stops working when conditions match	RW
b15~b12	Res	Reserved Bit	Read at “0”, write at “0”	RW
b11	HSTP11	Hardware stop condition 11	Condition: Internal Hardware Trigger Event 3 Valid 0: Hardware stop invalid when condition matches 1: Hardware stops working when conditions match	RW
b10	HSTP10	Hardware stop condition 10	Condition: Internal Hardware Trigger Event 2 Valid 0: Hardware stop invalid when condition matches 1: Hardware stops working when conditions match	RW
b9	HSTP9	Hardware stop condition 9	Condition: Internal Hardware Trigger Event 1Valid 0: Hardware stop invalid when condition matches 1: Hardware stops working when conditions match	RW
b8	HSTP8	Hardware stop condition 8	Condition: Internal Hardware Trigger Event 0Valid 0: Hardware stop invalid when condition matches 1: Hardware stops working when conditions match	RW
b7	STPS	Hardware disable	0: hardware stop is invalid 1: Hardware stop is valid **Note:** When hardware stop is enabled, the setting of SSTPR is invalid.	RW
b6~b4	Res	Reserved Bit	Read at “0”, write at “0”	RW
b3	HSTP3	Hardware stop condition 3	Condition: TMR6_<t>_PWMB port is sampled to the falling edge 0: Hardware stop invalid when condition matches 1: Hardware stops working when conditions match	RW
b2	HSTP2	Hardware stop condition 2	Condition: TMR6_<t>_PWMB port is sampled to the rising edge 0: Hardware stop invalid when condition matches 1: Hardware stops working when conditions match	RW
b1	HSTP1	Hardware stop condition 1	Condition: TMR6_<t>_PWMA port is sampled to the falling edge 0: Hardware stop invalid when condition matches 1: Hardware stops working when conditions match	RW
b0	HSTP0	Hardware stop condition 0	Condition: TMR6_<t>_PWMA port is sampled to the rising edge 0: Hardware stop invalid when condition matches 1: Hardware stops working when conditions match	RW

18.6.18 Hardware Clear Event Select Register (TMR6_HCLRR)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res								HCLE 23	HCLE 22	HCLE 21	HCLE 20	HCLE 19	HCLE 18	HCLE 17	HCLE 16
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res				HCLE 11	HCLE 10	HCLE 9	HCLE 8	CLES	Res			HCLE 3	HCLE 2	HCLE 1	HCLE 0

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b24	Res	Reserved Bit	Read at “0”, write at “0”	RW
b23	HCLE23	Hardware zeroing condition 23	Condition: TMR6_TRIGD port is sampled to the falling edge 0: Hardware zeroing is invalid when conditions match 1: Hardware zeroing is valid when conditions match	RW
b22	HCLE22	Hardware zeroing condition 22	Condition: TMR6_TRIGD port is sampled to the rising edge 0: Hardware zeroing is invalid when conditions match 1: Hardware zeroing is valid when conditions match	RW
b21	HCLE21	Hardware zeroing condition 21	Condition: TMR6_TRIGC port is sampled to the falling edge 0: Hardware zeroing is invalid when conditions match 1: Hardware zeroing is valid when conditions match	RW
b20	HCLE20	Hardware zeroing condition 20	Condition: TMR6_TRIGC port is sampled to the rising edge 0: Hardware zeroing is invalid when conditions match 1: Hardware zeroing is valid when conditions match	RW
b19	HCLE19	Hardware zeroing condition 19	Condition: TMR6_TRIGB port is sampled to the falling edge 0: Hardware zeroing is invalid when conditions match 1: Hardware zeroing is valid when conditions match	RW
b18	HCLE18	Hardware zeroing condition 18	Condition: TMR6_TRIGB port is sampled to the rising edge 0: Hardware zeroing is invalid when conditions match 1: Hardware zeroing is valid when conditions match	RW
b17	HCLE17	Hardware zeroing condition 17	Condition: TMR6_TRIGA port is sampled to the falling edge 0: Hardware zeroing is invalid when conditions match 1: Hardware zeroing is valid when conditions match	RW
b16	HCLE16	Hardware zeroing condition 16	Condition: TMR6_TRIGA port is sampled to the rising edge 0: Hardware zeroing is invalid when conditions match 1: Hardware zeroing is valid when conditions match	RW
b15~b12	Res	Reserved Bit	Read at “0”, write at “0”	RW
b11	HCLE11	Hardware zeroing condition 11	Condition: Internal Hardware Trigger Event 3 Valid 0: Hardware zeroing is invalid when conditions match 1: Hardware zeroing is valid when conditions match	RW
b10	HCLE10	Hardware zeroing condition 10	Condition: Internal Hardware Trigger Event 2 Valid 0: Hardware zeroing is invalid when conditions match 1: Hardware zeroing is valid when conditions match	RW
b9	HCLE9	Hardware zeroing condition 9	Condition: Internal Hardware Trigger Event 1Valid 0: Hardware zeroing is invalid when conditions match 1: Hardware zeroing is valid when conditions match	RW
b8	HCLE8	Hardware zeroing condition 8	Condition: Internal Hardware Trigger Event0Valid 0: Hardware zeroing is invalid when conditions match 1: Hardware zeroing is valid when conditions match	RW
b7	CLES	Hardware clear enable	0: Hardware reset is invalid 1: Hardware clear is valid **Note:** When the hardware clear is valid, the setting of SCLRR is invalid	RW
b6~b4	Res	Reserved Bit	Read at “0”, write at “0”	RW
b3	HCLE3	Hardware zeroing condition 3	Condition: TMR6_<t>_PWMB port is sampled to the falling edge 0: Hardware zeroing is invalid when conditions match 1: Hardware zeroing is valid when conditions match	RW
b2	HCLE2	Hardware zeroing condition 2	Condition: TMR6_<t>_PWMB port is sampled to the rising edge 0: Hardware zeroing is invalid when conditions match 1: Hardware zeroing is valid when conditions match	RW
b1	HCLE1	Hardware zeroing condition 1	Condition: TMR6_<t>_PWMA port is sampled to the falling edge 0: Hardware zeroing is invalid when conditions match 1: Hardware zeroing is valid when conditions match	RW
b0	HCLE0	Hardware zeroing condition 0	Condition: TMR6_<t>_PWMA port is sampled to the rising edge 0: Hardware zeroing is invalid when conditions match 1: Hardware zeroing is valid when conditions match	RW

18.6.19 Hardware Refresh Event Select Register (TMR6_HUPDR)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res								HUPD23	HUPD22	HUPD21	HUPD20	HUPD19	HUPD18	HUPD17	HUPD16
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res				HUPD11	HUPD10	HUPD9	HUPD8	UPDS	Res			HUPD3	HUPD2	HUPD1	HUPD0

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b24	Res	Reserved Bit	Read at “0”, write at “0”	RW
b23	HUPD23	Hardware refresh condition 23	Condition: TMR6_TRIGD port is sampled to the falling edge 0: When the condition is matched, hardware refresh invalid 1: When the conditions are met, hardware refresh valid	RW
b22	HUPD22	Hardware refresh condition 22	Condition: TMR6_TRIGD port is sampled to the rising edge 0: When the condition is matched, hardware refresh invalid 1: When the conditions are met, hardware refresh valid	RW
b21	HUPD21	Hardware refresh condition 21	Condition: TMR6_TRIGC port is sampled to the falling edge 0: When the condition is matched, hardware refresh invalid 1: When the conditions are met, hardware refresh valid	RW
b20	HUPD20	Hardware refresh condition 20	Condition: TMR6_TRIGC port is sampled to the rising edge 0: When the condition is matched, hardware refresh invalid 1: When the conditions are met, hardware refresh valid	RW
b19	HUPD19	Hardware refresh condition 19	Condition: TMR6_TRIGB port is sampled to the falling edge 0: When the condition is matched, hardware refresh invalid 1: When the conditions are met, hardware refresh valid	RW
b18	HUPD18	Hardware refresh condition 18	Condition: TMR6_TRIGB port is sampled to the rising edge 0: When the condition is matched, hardware refresh invalid 1: When the conditions are met, hardware refresh valid	RW
b17	HUPD17	Hardware refresh condition 17	Condition: TMR6_TRIGA port is sampled to the falling edge 0: When the condition is matched, hardware refresh invalid 1: When the conditions are met, hardware refresh valid	RW
b16	HUPD16	Hardware refresh condition 16	Condition: TMR6_TRIGA port is sampled to the rising edge 0: When the condition is matched, hardware refresh invalid 1: When the conditions are met, hardware refresh valid	RW
b15~b12	Res	Reserved Bit	Read at “0”, write at “0”	RW
b11	HUPD11	Hardware refresh condition 11	Condition: Internal hardware trigger event 3 is valid 0: When the condition is matched, hardware refresh invalid 1: When the conditions are met, hardware refresh valid	RW
b10	HUPD10	Hardware refresh condition 10	Condition: Internal hardware trigger event 2 is valid 0: When the condition is matched, hardware refresh invalid 1: When the conditions are met, hardware refresh valid	RW
b9	HUPD9	Hardware refresh condition 9	Condition: Internal hardware trigger event 1 is valid 0: When the condition is matched, hardware refresh invalid 1: When the conditions are met, hardware refresh valid	RW
b8	HUPD8	Hardware refresh condition 8	Condition: Internal hardware trigger event 0 is valid 0: When the condition is matched, hardware refresh invalid 1: When the conditions are met, hardware refresh valid	RW
b7	UPDS	Hardware refresh enable	0: Hardware refresh invalid 1: Hardware refresh valid **Note:** When hardware refresh is enabled, the SUPDR setting disable.	RW
b6~b4	Res	Reserved Bit	Read at “0”, write at “0”	RW
b3	HUPD3	Hardware refresh condition 3	Condition: TMR6_<t>_PWMB port is sampled to the falling edge 0: When the condition is matched, hardware refresh invalid 1: When the conditions are met, hardware refresh valid	RW
b2	HUPD2	Hardware refresh condition 2	Condition: TMR6_<t>_PWMB port is sampled to the rising edge 0: When the condition is matched, hardware refresh invalid 1: When the conditions are met, hardware refresh valid	RW
b1	HUPD1	Hardware refresh condition 1	Condition: TMR6_<t>_PWMA port is sampled to the falling edge 0: When the condition is matched, hardware refresh invalid 1: When the conditions are met, hardware refresh valid	RW
b0	HUPD0	Hardware refresh condition 0	Condition: TMR6_<t>_PWMA port is sampled to the rising edge 0: When the condition is matched, hardware refresh invalid 1: When the conditions are met, hardware refresh valid	RW

18.6.20 Hardware Capture Event Select Register A (TMR6_HCPAR)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res						HCPA 25	HCPA 24	HCPA 19	HCPA 18	HCPA 17	HCPA 16	HCPA 19	HCPA 18	HCPA 17	HCPA 16

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res				HCPA 11	HCPA 10	HCPA 9	HCPA 8	Res				HCPA 3	HCPA 2	HCPA 1	HCPA 0

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b26	Res	Reserved Bit	Read at “0”, write at “0”	RW
b25	HCPA25	Hardware Capture A Condition 25	Condition: TMR6_<t>_XOR signal is sampled to the falling edge 0: When the condition is matched, the hardware capture A is invalid 1: When the condition is matched, the hardware capture A is valid	RW
b24	HCPA24	Hardware Capture A Condition24	Condition: TMR6_<t>_XOR signal is sampled to the rising edge 0: When the condition is matched, the hardware capture A is invalid 1: When the condition is matched, the hardware capture A is valid	RW
b23	HCPA23	Hardware Capture A Condition23	Condition: TMR6_TRIGD port is sampled to the falling edge 0: When the condition is matched, the hardware capture A is invalid 1: When the condition is matched, the hardware capture A is valid	RW
b22	HCPA22	Hardware Capture A Condition22	Condition: TMR6_TRIGD port is sampled to the rising edge 0: When the condition is matched, the hardware capture A is invalid 1: When the condition is matched, the hardware capture A is valid	RW
b21	HCPA21	Hardware Capture A Condition21	Condition: TMR6_TRIGC port is sampled to the falling edge 0: When the condition is matched, the hardware capture A is invalid 1: When the condition is matched, the hardware capture A is valid	RW
b20	HCPA20	Hardware Capture A Condition20	Condition: TMR6_TRIGC port is sampled to the rising edge 0: When the condition is matched, the hardware capture A is invalid 1: When the condition is matched, the hardware capture A is valid	RW
b19	HCPA19	Hardware Capture A Condition19	Condition: TMR6_TRIGB port is sampled to the falling edge 0: When the condition is matched, the hardware capture A is invalid 1: When the condition is matched, the hardware capture A is valid	RW
b18	HCPA18	Hardware Capture A Condition18	Condition: TMR6_TRIGB port is sampled to the rising edge 0: When the condition is matched, the hardware capture A is invalid 1: When the condition is matched, the hardware capture A is valid	RW
b17	HCPA17	Hardware Capture A Condition17	Condition: TMR6_TRIGA port is sampled to the falling edge 0: When the condition is matched, the hardware capture A is invalid 1: When the condition is matched, the hardware capture A is valid	RW
b16	HCPA16	Hardware Capture A Condition16	Condition: TMR6_TRIGA port is sampled to the rising edge 0: When the condition is matched, the hardware capture A is invalid 1: When the condition is matched, the hardware capture A is valid	RW
b15~b12	Res	Reserved Bit	Read at “0”, write at “0”	RW
b11	HCPA11	Hardware Capture A Condition 11	Condition: Internal Hardware Trigger Event 3 Valid 0: When the condition is matched, the hardware capture A is invalid 1: When the condition is matched, the hardware capture A is valid	RW

b10	HCPA10	Hardware Capture A Condition 10	Condition: Internal Hardware Trigger Event 2 Valid 0: When the condition is matched, the hardware capture A is invalid 1: When the condition is matched, the hardware capture A is valid	RW
b9	HCPA9	Hardware Capture A Condition 9	Condition: Internal Hardware Trigger Event 1 Valid 0: When the condition is matched, the hardware capture A is invalid 1: When the condition is matched, the hardware capture A is valid	RW
b8	HCPA8	Hardware Capture A Condition 8	Condition: Internal Hardware Trigger Event 0 Valid 0: When the condition is matched, the hardware capture A is invalid 1: When the condition is matched, the hardware capture A is valid	RW
b7~b4	Res	Reserved Bit	Read at “0”, write at “0”	RW
b3	HCPA3	Hardware Capture A Condition 3	Condition: TMR6_<t>_PWMB port is sampled to the falling edge 0: When the condition is matched, the hardware capture A is invalid 1: When the condition is matched, the hardware capture A is valid	RW
b2	HCPA2	Hardware Capture A Condition 2	Condition: TMR6_<t>_PWMB port is sampled to the rising edge 0: When the condition is matched, the hardware capture A is invalid 1: When the condition is matched, the hardware capture A is valid	RW
b1	HCPA1	Hardware Capture A Condition 1	Condition: TMR6_<t>_PWMA port is sampled to the falling edge 0: When the condition is matched, the hardware capture A is invalid 1: When the condition is matched, the hardware capture A is valid	RW
b0	HCPA0	Hardware Capture A Condition 0	Condition: TMR6_<t>_PWMA port is sampled to the rising edge 0: When the condition is matched, the hardware capture A is invalid 1: When the condition is matched, the hardware capture A is valid	RW

Note:

The TMR6_<t>_XOR signal is the signal obtained by XORing the quadrature inputs AIN and BIN of each unit.

18.6.21 Hardware Capture Event Select Register B (TMR6_HCPBR)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res						HCPB 25	HCPB 24	HCP B23	HCP B22	HCP B21	HCP B20	HCPB 19	HCPB 18	HCPB 17	HCPB 16

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res				HCP B11	HCP B10	HCPB 9	HCPB 8	Res				HCPB 3	HCPB 2	HCPB 1	HCPB 0

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b26	Res	Reserved Bit	Read at “0”, write at “0”	RW
b25	HCPB25	Hardware Capture B Condition 25	Condition: TMR6_<t>_XOR signal is sampled to the falling edge 0: When the condition is matched, the hardware capture B is invalid 1: When the condition is matched, the hardware capture B is valid	RW
b24	HCPB24	Hardware Capture B Condition24	Condition: TMR6_<t>_XOR signal is sampled to the rising edge 0: When the condition is matched, the hardware capture B is invalid 1: When the condition is matched, the hardware capture B is valid	RW
b23	HCPB23	Hardware Capture B Condition23	Condition: TMR6_TRIGD port is sampled to the falling edge 0: When the condition is matched, the hardware capture B is invalid 1: When the condition is matched, the hardware capture B is valid	RW
b22	HCPB22	Hardware Capture B Condition22	Condition: TMR6_TRIGD port is sampled to the rising edge 0: When the condition is matched, the hardware capture B is invalid 1: When the condition is matched, the hardware capture B is valid	RW
b21	HCPB21	Hardware Capture B Condition21	Condition: TMR6_TRIGC port is sampled to the falling edge 0: When the condition is matched, the hardware capture B is invalid 1: When the condition is matched, the hardware capture B is valid	RW
b20	HCPB20	Hardware Capture B Condition20	Condition: TMR6_TRIGC port is sampled to the rising edge 0: When the condition is matched, the hardware capture B is invalid 1: When the condition is matched, the hardware capture B is valid	RW
b19	HCPB19	Hardware Capture B Condition19	Condition: TMR6_TRIGB port is sampled to the falling edge 0: When the condition is matched, the hardware capture B is invalid 1: When the condition is matched, the hardware capture B is valid	RW
b18	HCPB18	Hardware Capture B Condition18	Condition: TMR6_TRIGB port is sampled to the rising edge 0: When the condition is matched, the hardware capture B is invalid 1: When the condition is matched, the hardware capture B is valid	RW
b17	HCPB17	Hardware Capture B Condition17	Condition: TMR6_TRIGA port is sampled to the falling edge 0: When the condition is matched, the hardware capture B is invalid 1: When the condition is matched, the hardware capture B is valid	RW
b16	HCPB16	Hardware Capture B Condition16	Condition: TMR6_TRIGA port is sampled to the rising edge 0: When the condition is matched, the hardware capture B is invalid 1: When the condition is matched, the hardware capture B is valid	RW
b15~b12	Res	Reserved Bit	Read at “0”, write at “0”	RW
b11	HCPB11	Hardware Capture B Condition 11	Condition: Internal Hardware Trigger Event 3 Valid 0: When the condition is matched, the hardware capture B is invalid 1: When the condition is matched, the hardware capture B is valid	RW
b10	HCPB10	Hardware Capture B Condition10	Condition: Internal Hardware Trigger Event 2 Valid 0: When the condition is matched, the hardware capture B is	RW

					invalid 1: When the condition is matched, the hardware capture B is valid	
b9	HCPB9	Hardware Capture Condition9	Capture	B	Condition: Internal Hardware Trigger Event 1 Valid 0: When the condition is matched, the hardware capture B is invalid 1: When the condition is matched, the hardware capture B is valid	RW
b8	HCPB8	Hardware Capture Condition8	Capture	B	Condition: Internal Hardware Trigger Event 0 Valid 0: When the condition is matched, the hardware capture B is invalid 1: When the condition is matched, the hardware capture B is valid	RW
b7~b4	Res	Reserved Bit			Read at “0”, write at “0”	RW
b3	HCPB3	Hardware Capture Condition 3	Capture	B	Condition: TMR6_<t>_PWMB port is sampled to the falling edge 0: When the condition is matched, the hardware capture B is invalid 1: When the condition is matched, the hardware capture B is valid	RW
b2	HCPB2	Hardware Capture Condition2	Capture	B	Condition: TMR6_<t>_PWMB port is sampled to the rising edge 0: When the condition is matched, the hardware capture B is invalid 1: When the condition is matched, the hardware capture B is valid	RW
b1	HCPB1	Hardware Capture Condition1	Capture	B	Condition: TMR6_<t>_PWMA port is sampled to the falling edge 0: When the condition is matched, the hardware capture B is invalid 1: When the condition is matched, the hardware capture B is valid	RW
b0	HCPB0	Hardware capture condition0	capture	B	Condition: TMR6_<t>_PWMA port is sampled to the rising edge 0: When the condition is matched, the hardware capture B is invalid 1: When the condition is matched, the hardware capture B is valid	RW

Note:

The TMR6_<t>_XOR signal is the signal obtained by XORing the quadrature inputs AIN and BIN of each unit.

18.6.22 Hardware Increment Event Select Register (TMR6_HCUPR)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res								HCUP23	HCUP22	HCUP21	HCUP20	HCUP19	HCUP18	HCUP17	HCUP16
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res				HCUP11	HCUP10	HCUP9	HCUP8	HCUP7	HCUP6	HCUP5	HCUP4	HCUP3	HCUP2	HCUP1	HCUP0

Bit/Bitfield	Marking	Bit Name		Function	Read and Write
b31~b24	Res	Reserved Bit		Read at “0”, write at “0”	RW
b23	HCUP23	Hardware Condition 23	Added	Condition: TMR6_TRIGD port is sampled to the falling edge 0: Hardware add invalid when condition matches 1: Hardware addition is valid when a condition matches	RW
b22	HCUP22	Hardware Condition 22	Added	Condition: TMR6_TRIGD port is sampled to the rising edge 0: Hardware add invalid when condition matches 1: Hardware addition is valid when a condition matches	RW
b21	HCUP21	Hardware Condition 21	Added	Condition: TMR6_TRIGC port is sampled to the falling edge 0: Hardware add invalid when condition matches 1: Hardware addition is valid when a condition matches	RW
b20	HCUP20	Hardware Condition 20	Added	Condition: TMR6_TRIGC port is sampled to the rising edge 0: Hardware add invalid when condition matches 1: Hardware addition is valid when a condition matches	RW
b19	HCUP19	Hardware Condition 19	Added	Condition: TMR6_TRIGB port is sampled to the falling edge 0: Hardware add invalid when condition matches 1: Hardware addition is valid when a condition matches	RW
b18	HCUP18	Hardware Condition 18	Added	Condition: TMR6_TRIGB port is sampled to the rising edge 0: Hardware add invalid when condition matches 1: Hardware addition is valid when a condition matches	RW
b17	HCUP17	Hardware Condition 17	Added	Condition: TMR6_TRIGA port is sampled to the falling edge 0: Hardware add invalid when condition matches 1: Hardware addition is valid when a condition matches	RW
b16	HCUP16	Hardware Condition 16	Added	Condition: TMR6_TRIGA port is sampled to the rising edge 0: Hardware add invalid when condition matches 1: Hardware addition is valid when a condition matches	RW
b15~b12	Res	Reserved Bit		Read at “0”, write at “0”	RW
b11	HCUP11	Hardware Conditions11	Added	Condition: Internal Hardware Trigger Event 3 Valid 0: Hardware add invalid when condition matches 1: Hardware addition is valid when a condition matches	RW
b10	HCUP10	Hardware Condition 10	Added	Condition: Internal Hardware Trigger Event 2 Valid 0: Hardware add invalid when condition matches 1: Hardware addition is valid when a condition matches	RW
b9	HCUP9	Hardware Condition 9	Added	Condition: Internal Hardware Trigger Event 1Valid 0: Hardware add invalid when condition matches 1: Hardware addition is valid when a condition matches	RW
b8	HCUP8	Hardware Condition 8	Added	Condition: Internal Hardware Trigger Event0Valid 0: Hardware add invalid when condition matches 1: Hardware addition is valid when a condition matches	RW
b7	HCUP7	Hardware Condition 7	Added	Condition: When the TMR6_<t>_PWMB port is high, the TMR6_<t>_PWMA port is sampled to the falling edge 0: Hardware add invalid when condition matches 1: Hardware addition is valid when a condition matches	RW
b6	HCUP6	Hardware Condition 6	Added	Condition: When the TMR6_<t>_PWMB port is high, the TMR6_<t>_PWMA port is sampled to the rising edge 0: Hardware add invalid when condition matches 1: Hardware addition is valid when a condition matches	RW
b5	HCUP5	Hardware Condition 5	Added	Condition: When the TMR6_<t>_PWMB port is low, the TMR6_<t>_PWMA port is sampled to the falling edge 0: Hardware add invalid when condition matches 1: Hardware addition is valid when a condition matches	RW
b4	HCUP4	Hardware Condition 4	Added	Condition: When the TMR6_<t>_PWMB port is low, the TMR6_<t>_PWMA port is sampled to the rising edge 0: Hardware add invalid when condition matches 1: Hardware addition is valid when a condition matches	RW
b3	HCUP3	Hardware Condition 3	Added	Condition: When the TMR6_<t>_PWMA port is high, the TMR6_<t>_PWMB port is sampled to the falling edge 0: Hardware add invalid when condition matches 1: Hardware addition is valid when a condition matches	RW
b2	HCUP2	Hardware Condition 2	Added	Condition: When the TMR6_<t>_PWMA port is high, the TMR6_<t>_PWMB port is sampled to the rising edge	RW

				0: Hardware add invalid when condition matches 1: Hardware addition is valid when a condition matches	
b1	HCUP1	Hardware Condition 1	Added	Condition: When the TMR6_<t>_PWMA port is low, the TMR6_<t>_PWMB port is sampled to the falling edge 0: Hardware add invalid when condition matches 1: Hardware addition is valid when a condition matches	RW
b0	HCUP0	Hardware Condition 0	Added	Condition: When the TMR6_<t>_PWMA port is low, the TMR6_<t>_PWMB port is sampled to the rising edge 0: Hardware add invalid when condition matches 1: Hardware addition is valid when a condition matches	RW

18.6.23 Hardware Decrement Event Select Register (TMR6_HCDOR)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res								HCDO23	HCDO22	HCDO21	HCDO20	HCDO19	HCDO18	HCDO17	HCDO16
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res				HCDO11	HCDO10	HCDO9	HCDO8	HCDO7	HCDO6	HCDO5	HCDO4	HCDO3	HCDO2	HCDO1	HCDO0

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b20	Res	Reserved Bit	Read at “0”, write at “0”	RW
b23	HCDO23	Hardware decrement condition 23	Condition: TMR6_TRIGD port is sampled to the falling edge 0: Hardware decrement is invalid when conditions match 1: Hardware decrements are effective when conditions match	
b22	HCDO22	Hardware decrement condition 22	Condition: TMR6_TRIGD port is sampled to the rising edge 0: Hardware decrement is invalid when conditions match 1: Hardware decrements are effective when conditions match	
b21	HCDO21	Hardware decrement condition 21	Condition: TMR6_TRIGC port is sampled to the falling edge 0: Hardware decrement is invalid when conditions match 1: Hardware decrements are effective when conditions match	
b20	HCDO20	Hardware decrement condition 20	Condition: TMR6_TRIGC port is sampled to the rising edge 0: Hardware decrement is invalid when conditions match 1: Hardware decrements are effective when conditions match	
b19	HCDO19	Hardware decrement condition 19	Condition: TMR6_TRIGB port is sampled to the falling edge 0: Hardware decrement is invalid when conditions match 1: Hardware decrements are effective when conditions match	
b18	HCDO18	Hardware decrement condition 18	Condition: TMR6_TRIGB port is sampled to the rising edge 0: Hardware decrement is invalid when conditions match 1: Hardware decrements are effective when conditions match	
b17	HCDO17	Hardware decrement condition 17	Condition: TMR6_TRIGA port is sampled to the falling edge 0: Hardware decrement is invalid when conditions match 1: Hardware decrements are effective when conditions match	
b16	HCDO16	Hardware decrement condition 16	Condition: TMR6_TRIGA port is sampled to the rising edge 0: Hardware decrement is invalid when conditions match 1: Hardware decrements are effective when conditions match	
b15~b12	Res	Reserved Bit	Read at “0”, write at “0”	RW
b11	HCDO11	Hardware decrement condition11	Condition: Internal Hardware Trigger Event 3 Valid 0: Hardware decrement is invalid when conditions match 1: Hardware decrements are effective when conditions match	RW
b10	HCDO10	Hardware decrement condition 10	Condition: Internal Hardware Trigger Event 2 Valid 0: Hardware decrement is invalid when conditions match 1: Hardware decrements are effective when conditions match	RW
b9	HCDO9	Hardware decrement condition 9	Condition: Internal Hardware Trigger Event 1Valid 0: Hardware decrement is invalid when conditions match 1: Hardware decrements are effective when conditions match	RW
b8	HCDO8	Hardware decrement condition 8	Condition: Internal Hardware Trigger Event0Valid 0: Hardware decrement is invalid when conditions match 1: Hardware decrements are effective when conditions match	RW
b7	HCDO7	Hardware decrement condition 7	Condition: When the TMR6_<t>_PWMB port is high, the TMR6_<t>_PWMA port is sampled to the falling edge 0: Hardware decrement is invalid when conditions match 1: Hardware decrements are effective when conditions match	RW
b6	HCDO6	Hardware decrement condition 6	Condition: When the TMR6_<t>_PWMB port is high, the TMR6_<t>_PWMA port is sampled to the rising edge 0: Hardware decrement is invalid when conditions match 1: Hardware decrements are effective when conditions match	RW

b5	HCDO5	Hardware decrement condition 5	Condition: When the TMR6_<t>_PWMB port is low, the TMR6_<t>_PWMA port is sampled to the falling edge 0: Hardware decrement is invalid when conditions match 1: Hardware decrements are effective when conditions match	RW
b4	HCDO4	Hardware decrement condition 4	Condition: When the TMR6_<t>_PWMB port is low, the TMR6_<t>_PWMA port is sampled to the rising edge 0: Hardware decrement is invalid when conditions match 1: Hardware decrements are effective when conditions match	RW
b3	HCDO3	Hardware decrement condition 3	Condition: When the TMR6_<t>_PWMA port is high, the TMR6_<t>_PWMB port is sampled to the falling edge 0: Hardware decrement is invalid when conditions match 1: Hardware decrements are effective when conditions match	RW
b2	HCDO2	Hardware decrement condition 2	Condition: When the TMR6_<t>_PWMA port is high, the TMR6_<t>_PWMB port is sampled to the rising edge 0: Hardware decrement is invalid when conditions match 1: Hardware decrements are effective when conditions match	RW
b1	HCDO1	Hardware decrement condition 1	Condition: When the TMR6_<t>_PWMA port is low, the TMR6_<t>_PWMB port is sampled to the falling edge 0: Hardware decrement is invalid when conditions match 1: Hardware decrements are effective when conditions match	RW
b0	HCDO0	Hardware decrement condition 0	Condition: When the TMR6_<t>_PWMA port is low, the TMR6_<t>_PWMB port is sampled to the rising edge 0: Hardware decrement is invalid when conditions match 1: Hardware decrements are effective when conditions match	RW

18.6.24 Common Filter Control Register (TMR6_FCNTR)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res	NOFICKTD [1:0]		NOFI ENTD	Res	NOFICKTC [1:0]		NOFI ENTC	Res	NOFICKTB [1:0]		NOFI ENTB	Res	NOFICKTA [1:0]		NOFI ENTA

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b15	Res	Reserved Bit	Read at “0”, write at “0”	RW
b14~b13	NOFICKTD[1:0]	Filter sampling reference clock selection TD	00：PCLK0 01：PCLK0/4 10：PCLK0/16 11：PCLK0/64	RW
b12	NOFIENTD	Capture input port filter TD	0: TMR6_TRIGD port input filter function is invalid 1: TMR6_TRIGD port input filter function is enabled	RW
b11	Res	Reserved Bit	Read at “0”, write at “0”	RW
b10~b9	NOFICKTC[1:0]	Filter sampling reference clock selection TC	00：PCLK0 01：PCLK0/4 10：PCLK0/16 11：PCLK0/64	RW
b8	NOFIENTC	Capture input port filter TC	0: TMR6_TRIGC port input filter function is invalid 1: TMR6_TRIGC port input filter function is enabled	RW
b7	Res	Reserved Bit	Read at “0”, write at “0”	RW
b6~b5	NOFICKTB[1:0]	Filter sampling reference clock selection TB	00：PCLK0 01：PCLK0/4 10：PCLK0/16 11：PCLK0/64	RW
b4	NOFIENTB	Capture input port filter TB	0: TMR6_TRIGB port input filter function is invalid 1: TMR6_TRIGB port input filter function is enabled	RW
b3	Res	Reserved Bit	Read at “0”, write at “0”	RW
b2~b1	NOFICKTA[1:0]	Filter sampling reference clock selection TA	00：PCLK0 01：PCLK0/4 10：PCLK0/16 11：PCLK0/64	RW
b0	NOFIENTA	Capture input port filter TA	0: TMR6_TRIGA port input filter function is invalid 1: TMR6_TRIGA port input filter function is enabled	RW

18.6.25 Common Software Synchronous Start Control Register (TMR6_SSTAR)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res												SSTA4	SSTA3	SSTA2	SSTA1

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b4	Res	Reserved Bit	Read at “0”, write at “0”	RW
b3	SSTA4	Unit 4 software start	0: Software startup is invalid 1: Software startup enable	RW
b2	SSTA3	Unit 3 Software Startup	0: Software startup is invalid 1: Software startup enable	RW
b1	SSTA2	Unit 2 software start	0: Software startup is invalid 1: Software startup enable	RW
b0	SSTA1	Unit 1 software start	0: Software startup is invalid 1: Software startup enable	RW

18.6.26 Common Software Synchronous Stop Control Register (TMR6_SSTPR)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res												SSTP4	SSTP3	SSTP2	SSTP1

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b4	Res	Reserved Bit	Read at “0”, write at “0”	RW
b3	SSTP4	Unit 4 software stop	0: Software stop is invalid 1: Software stop enable	RW
b2	SSTP3	Unit 3 software stop	0: Software stop is invalid 1: Software stop enable	RW
b1	SSTP2	Unit 2 software stop	0: Software stop is invalid 1: Software stop enable	RW
b0	SSTP1	Unit 1 software stop	0: Software stop is invalid 1: Software stop enable	RW

18.6.27 Common Software Synchronous Clear Control Register (TMR6_SCLRR)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res												SCLE4	SCLE3	SCLE2	SCLE1

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b4	Res	Reserved Bit	Read at “0”, write at “0”	RW
b3	SCLE4	Unit 4 software clear	0: Software reset is invalid 1: Software clear enable	RW
b2	SCLE3	Unit 3 software clear	0: Software reset is invalid 1: Software clear enable	RW
b1	SCLE2	Unit 2 software clear	0: Software reset is invalid 1: Software clear enable	RW
b0	SCLE1	Unit 1 software clear	0: Software reset is invalid 1: Software clear enable	RW

18.6.28 Common Software Synchronous Refresh Control Register (TMR6_SUPDR)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res												SUPD4	SUPD3	SUPD2	SUPD1

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b4	Res	Reserved Bit	Read at “0”, write at “0”	RW
b3	SUPD4	Unit 4 software refresh	0: Software refresh invalid 1: Software refresh enable	RW
b2	SUPD3	Unit 3 software update	0: Software refresh invalid 1: Software refresh enable	RW
b1	SUPD2	Unit 2 software update	0: Software refresh invalid 1: Software refresh enable	RW
b0	SUPD1	Unit 1 software update	0: Software refresh invalid 1: Software refresh enable	RW

18.7Precautions

- The CPU cannot write to the CNTER register when the counter is counting, and can only write to the CNTER register when the counter stops.
- When the corresponding cache function of GCMCR~GCMFR valid, its interrupt output and event output are invalid.
- When using the bilateral symmetrical complementary PWM output function, the user needs to calculate and set the relevant reference value and port output status to ensure that the port is in the expected output state.
- In some specific situations, the control events of the counter may appear at the same time. At this time, the change control priority of the counter (CNTER) is shown in Table 18-7 below.

Table 18-7 Counter (CNTER) Control Priority

Timer Control Events		Counter (CNTER) value	Priority
The CPU performs a write operation on CNTER		CPU write value	high
Refresh event (software refresh or hardware refresh)		Values in UPDAR	↓
Clear event (software clear or hardware clear)	Hardware counting or sawtooth mode	Depending on the counting direction, it becomes 0 or PERAR	↓
	In triangle wave mode	0	↓
Hardware Count		Normal count value	↓
Triangle wave mode counting		Normal count value	↓
Sawtooth pattern counting		Normal count value	Low

- In some specific situations, the control events of the PWM port output may appear at the same time. At this time, the output change control priorities of the two ports (TMR6_<t>_PWMA and TMR6_<t>_PWMB) are shown in Table 18-8 and
- Table 18-9 respectively.

Table 18-8 PWMA Port Output Control Priority

PWMA Port Output Control Event		Port Status	Priority
EMB selected event occurs		PCNAR.EMBCA设定状态	high
Set forced port output		PCNAR.FORCA设定状态	↓
When counting starts		PCNAR.STACA设定状态	↓
When the sawtooth wave counts up	CNTER=PERAR	PCNAR.OVFCA设定状态	↓
	CNTER=GCMAR	PCNAR.CMAUCA设定状态	↓
	CNTER=GCMBR	PCNAR.CMBUCA设定状态	↓
When the sawtooth wave counts down	CNTER=0	PCNAR.UDFCA设定状态	↓
	CNTER=GCMAR	PCNAR.CMADCA设定状态	↓
	CNTER=GCMBR	PCNAR.CMBDCA设定状态	↓
When the triangle wave counts up	CNTER=GCMAR	PCNAR.CMAUCA设定状态	↓
	CNTER=GCMBR	PCNAR.CMBUCA设定状态	↓

PWMA Port Output Control Event		Port Status	Priority
	CNTER=PERAR	PCNAR.OVFCA设定状态	↓
When the triangle wave counts down	CNTER=GCMAR	PCNAR.CMADCA设定状态	↓
	CNTER=GCMBR	PCNAR.CMBDCA设定状态	↓
	CNTER=0	PCNAR.UDFCA设定状态	↓
When counting stops		PCNAR.STPCA设定状态	Low

Table 18-9 PWMB Port Output Control Priority

PWMB Port Output Control Event		Port Status	Priority
EMB selected event occurs		PCNBR.EMBCB setting status	high
Set forced port output		PCNBR.FORCB setting status	↓
When counting starts		PCNBR.STACB setting status	↓
When the sawtooth wave counts up	CNTER=PERAR	PCNBR.OVFCB set status	↓
	CNTER=GCMBR	PCNBR.CMBUCB setting status	↓
	CNTER=GCMAR	PCNBR.CMAUCB setting status	↓
When the sawtooth wave counts down	CNTER=0	PCNBR.UDFCB setting status	↓
	CNTER=GCMBR	PCNBR.CMBDCB setting status	↓
	CNTER=GCMAR	PCNBR.CMADCB set status	↓
When the triangle wave counts up	CNTER=GCMBR	PCNBR.CMBUCB setting status	↓
	CNTER=GCMAR	PCNBR.CMAUCB setting status	↓
	CNTER=PERAR	PCNBR.OVFCB set status	↓
When the triangle wave counts down	CNTER=GCMBR	PCNBR.CMBDCB setting status	↓
	CNTER=GCMAR	PCNBR.CMADCB set status	↓
	CNTER=0	PCNBR.UDFCB setting status	↓
When counting stops		PCNBR.STPCB setting status	Low

19 High-resolution PWM (HRPWM)

19.1 Overview

The High Resolution PWM (HRPWM) is a 22-bit timer capable of generating up to 12 high-resolution PWM waveforms. two count modes : Up-count Mode(Sawtooth) and Up-Down Count Mode(Triangle);can generate asymmetric PWM and symmetric PWM; software synchronization and hardware synchronization (synchronous start, clear, capture, etc.) . Supports synchronous output to synchronize off-chip counters; supports Shadow function; supports multiple idle controls: immediate idle, Burst Mode and delayed idle; Spurious events filtering with blanking and windowing modes; supports EMB(Fault) control; supports synchronous DAC; supports interleaved phase shift control between multi-phase units. This series of products HRPWM is equipped with 6 counting units and supports up to 6 groups of high-precision PWM outputs.

Main Features of HRPWM:

- High-resolution timing Units
  - 104 ps resolution, compensated against voltage and temperature variations
  - High-resolution available on all outputs, possibility to adjust duty-cycle, frequency and phase,etc;
  - 6 * 22-bit timing units (each one with an independent counter, has 6 compare units and 2 Special-compares)
  - Up-count Mode(Sawtooth) and Up-Down Count Mode(Triangle)
  - Each timing units represents one complete PWM channel composed of two PWM outputs: HRPWMxA and HRPWMxB; HRGCMPA/B/E/F,SCMPA/B Specify the time at which switching events occur on the HRPWMxA or HRPWMxA output;
  - Supports 2 capture units;
  - Supports Shadow Mode;
- Up to 10 external events, available for any timing unit
  - Programmable polarity and edge sensitivity
  - 5 external events support fast asynchronous mode
  - 5 external events support digital filtering
  - Spurious events filtering with blanking and windowing Modes
- Support EMB function
  - Each unit has 6 EMB groups to choose from
  - Supports One shot or Cycle by Cycle
- Support synchronous output
  - Supports synchronous output signal to synchronize off-chip counter
- Support multiple synchronization methods
  - Software synchronous start, clear, update and stop

  - Hardware synchronization start and reset, hardware synchronization source can be configured as off-chip and on-chip events
  - Unit 1 phase matching event synchronously clears and starts units 2~6
- Versatile PWM output
  - Support dead zone function (resolution is 104ps)
  - Support output polarity selection
  - Support A/B channel switching output function
- Multiple idle processing
  - immediate Idle
  - Burst Mode
  - Delayed Idle
- Support interrupt
  - 6 General purpose compare match(HRGCMPA/B/C/D/E/F) interrupts per unit
  - 2 SCMAR/SCMBR interrupts per unit
  - 1 Period match interrupt per unit
  - 1 zero match interrupt per unit
  - capture interrupts per unit
  - 1 calibration complete interrupt
  - 1 Burst Mode period match interrupt
  - 1 delayed idle trigger interrupt
- Support event output to AOS unit
  - 6 General purpose compare match(HRGCMPA/B/C/D/E/F) events per unit
  - 2 SCMAR/SCMBR events per unit
  - 1 Period match event per unit
  - 1 zero match event per unit
  - 2 capture events per unit
  - 1 external event detected
- Support DAC trigger
  - 3 DAC trigger signals to synchronize DAC

19.2Basic Block Diagram

The basic functions and characteristics of HRPWM are shown in the Table 19-1.

Table 19-1 HRPWM Basic Functions and Features

Couter mode	Up-count Mode(Sawtooth) and Up-Down Count Mode(Triangle)
Basic functions	capture function
	Software synchronization
	Hardware synchronization
	sync output
	Shadow function
	High-resolution PWM output (provides 104ps resolution)
	EMB control
	Burst Mode Controller
	Delayed idle
	External events filter support blanking and windowing modes
Interrupt output	Count compare match interrupt
	Count period match interrupt
	Count zero match interrupt
	Capture Interrupt
	Calibration Complete Interrupt
	Burst Mode period match interrupt
	Delayed idle trigger interrupt
Event output	Count compare match events
	External event detection event
	Count period match event
	Count zero match events
	Capturing Events
DAC trigger signal	DAC sync trigger signal

19.3Functional Description

19.3.1 HRWPM Block Diagram

The basic block diagram of HRPWM is shown in the Figure 19-1. The “<t>” shown in the block diagram represents the unit number, that is, “<t>” is 1 to 6. In the rest of this chapter, “<t>” refers to the unit number, and y is the external event number, that is, y is 1 to 10.

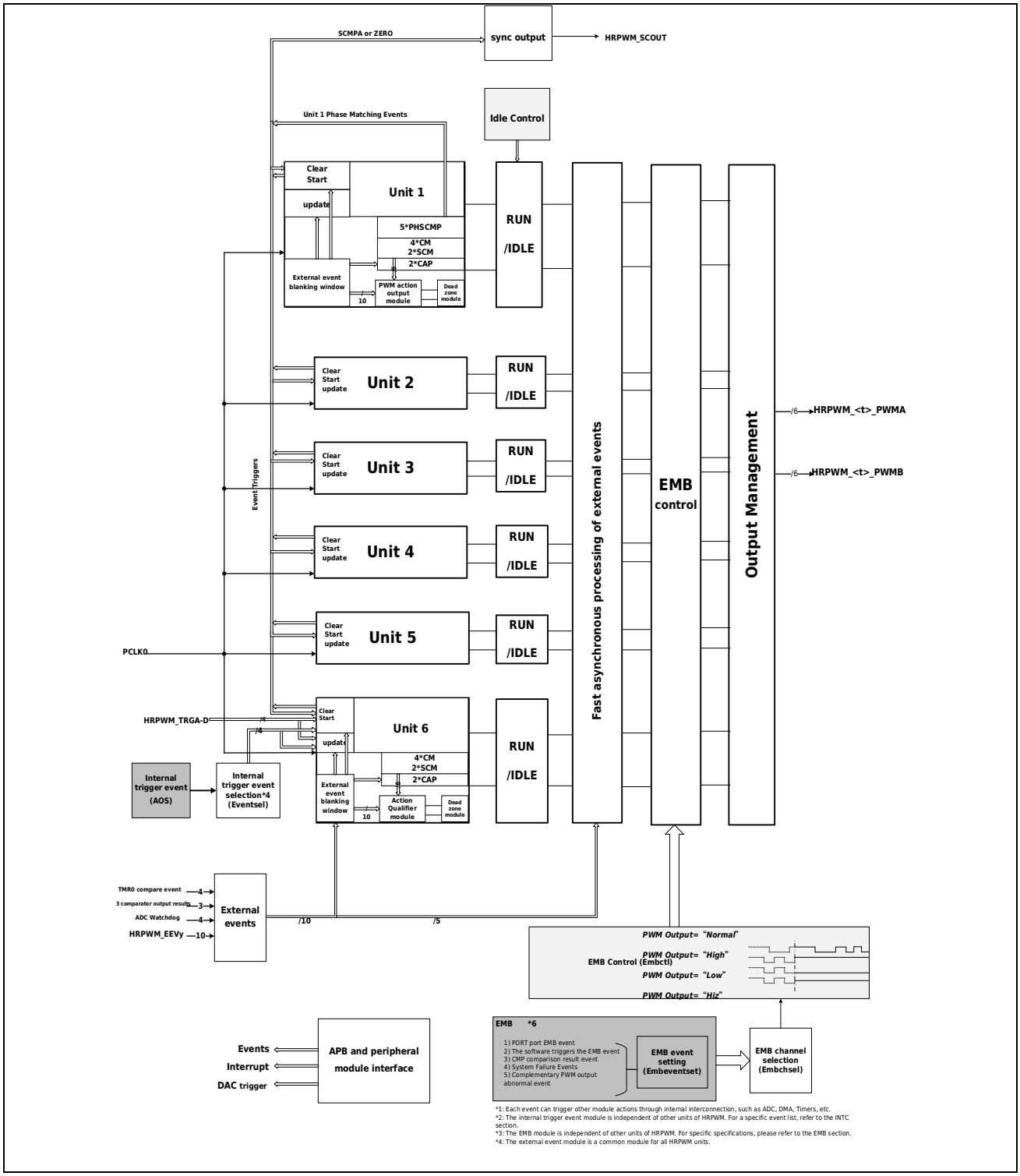


Figure 19-1 HRPWM Basic Block Diagram

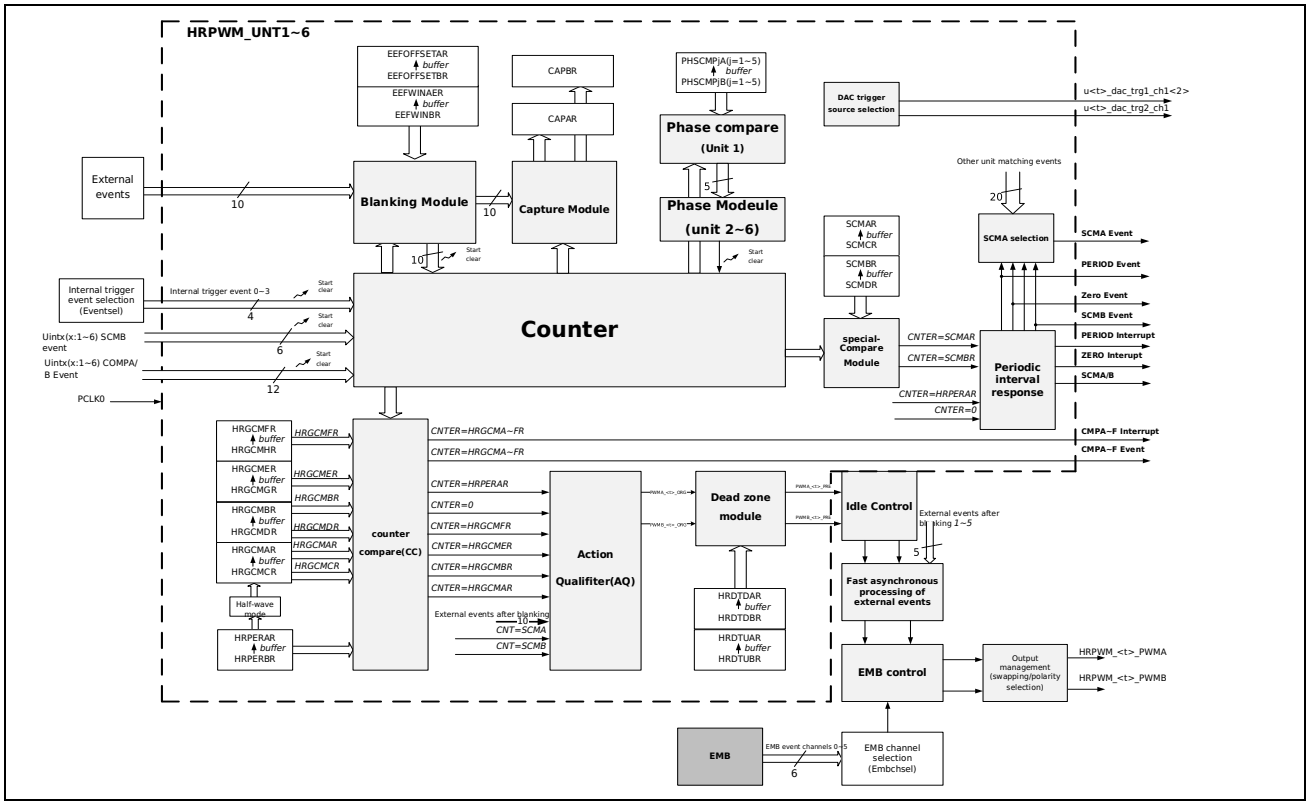


Figure 19-2 HRPWM Basic Block Diagram of Each Unit

Table 19-2 shows the input and output port list of HRPWM.

Table 19-2 HRPWM Port List

Port Name	Direction	Function
HRPWM_<t>_PWMA	out	PWM output port
HRPWM_<t>_PWMB		
HRPWM_TRIGA	in	input port for Hardware synchronous start/clear;
HRPWM_TRIGB		
HRPWM_TRIGC		
HRPWM_TRIGD		
HRPWM_EEVy (y=1~10)	in	External event input port
HRPWM_SCOUT	out	Synchronous output port

19.3.2 Counter Mode

HRPWM has two basic counting waveform modes, sawtooth counter mode and triangle counter mode. The basic waveforms of the two counter modes are shown in Figure 18-2 and Figure 19-4.

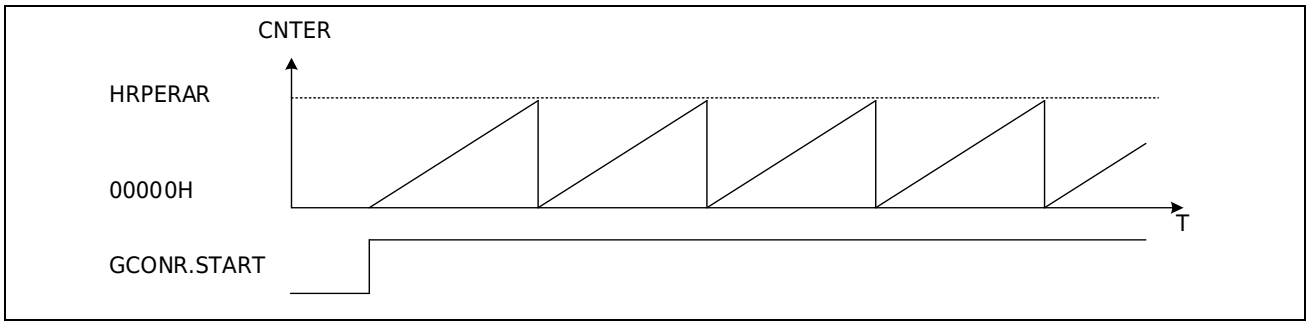


Figure 19-3 Sawtooth Counter Mode

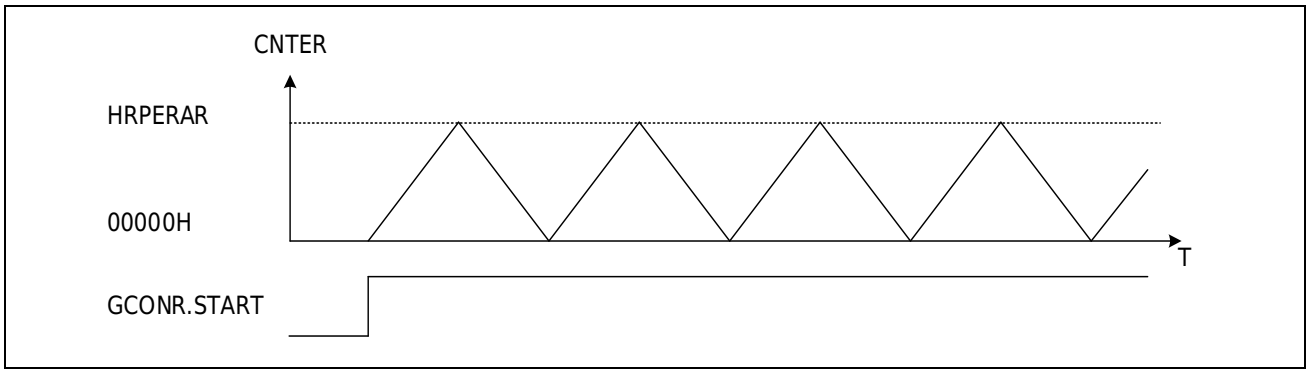


Figure 19-4 Triangle Counter Mode

Note:

In sawtooth counter *mode, when the counter value is greater than the period value, the counter will become 0 and start counting again; in triangle* counter *mode, when the count value is greater than the period value, the counter will jump to the period value and start counting down.*

19.3.3 Counter Clock

The HRPWM calibration module improves the resolution of the PWM waveform by evenly dividingPCLK0 into 64 parts: The HRPWM calibration module evenly divide PCLK0 into up to 64 intermediate steps using an edge positioning logic. The resolution of HRPWM and the minimum frequency of PWM waveform are shown in Table 19-3.

Table 19-3 HRPWM Resolution

Waveform Mode	PCLK0 Frequency	Resolution	HRPERAR	PWM Minimum Frequency
Sawtooth wave	150MHz	104ps	0x3FFFC0	2.29kHz
Triangle wave			0x3FFFC0	1.14kHz

When the EN bit of the high-precision control register HRPWM<t>_CR is 1, high resolution is

effective, which can support the adjustment of high-precision period, high-precision duty cycle, high-precision dead zone and high-precision phase.

In this mode, the following functions do not support high resolution:

- Writing and reading counter values
- Hardware synchronous clear event
- capture function
- Hardware Update
- External event blanking window
- Dedicated baseline value

When the high-precision control register HRPWM<t>_CR.EN is 0, high resolution invalid. In this mode, the values set in the lower six bits of the high-resolution period register, high-precision General Compare Register, high-resolution dead zone register, and high-resolution phase Compare Register are invalid.

19.3.4 Calibration Function

Before enabling HRPWM (HRPWM<t>CR.EN=1), the HRPWM must be calibrated using the calibration Module, otherwise the high-resolution function cannot be used normally.

By configuring the CAL and CALEN bits in the calibration control register HRPWM_CALCR, single calibration or periodic calibration can be performed. The calibration period of periodic calibration can be configured by the HRPWM_CALCR.CALPRD[3:0] register bits. After calibration is completed, the CALENF bit in CALCR is set to 1.

This calibration unit can calibrate PCLK0 at 150M frequency. When the frequency is too low, a calibration error will occur and HRPWM_CALCR .ERRF will be set to 1.

19.3.5 Counting Direction

19.3.5.1 Sawtooth Counting Direction

In sawtooth counter mode, the HRPWM counting direction is up counting.

19.3.5.2 Triangular Counting Direction

In triangle count mode, the counter automatically changes the counting direction when it counts up to the period value -64 or counts down to 64.

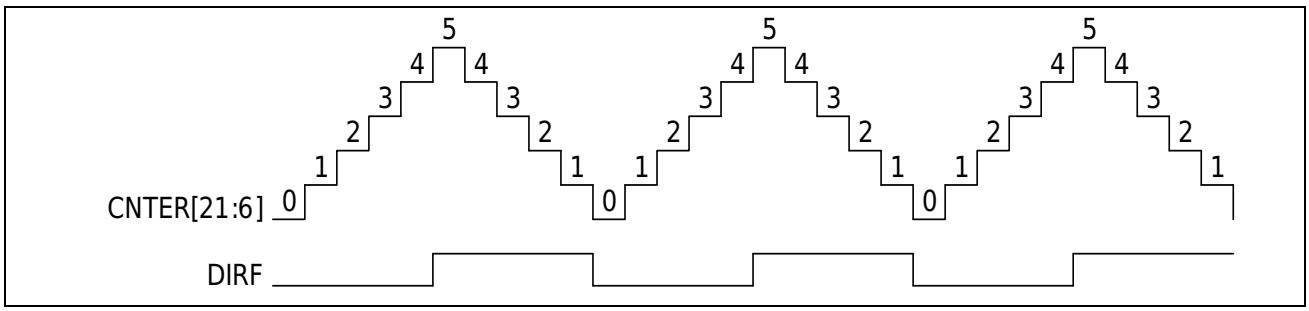


Figure 19-5 Triangle counting direction

19.3.6 Capture Function

Each unit has a capture function and has two sets of capture input registers (HRPWM<t>_CAPAR, HRPWM<t>_CAPBR) to save the captured current count value. Setting the HCPA<B>EN bit in the capture event selection register (HRPWM<t>_HCPA<B>R2) to 1 enables the hardware capture function. When the corresponding capture input condition is set and the condition valid, the current count value is saved in the corresponding capture register (HRPWM<t>_CAPAR, HRPWM<t>_CAPBR), where CAPDIRA/B in the register indicates the counting direction of the counter when capture occurs. 1 indicates that the counting direction is down counting when capture occurs, and 0 indicates that the counting direction is up counting when capture occurs. The capture value of the capture register does not support high precision, and the lower six bits are read as 0. At the same time, the CAPA<B>F bit in the status flag register (HRPWM<t>_STFLR1) is 1 when capture occurs.

The conditions for each capture input of each unit can be internal trigger event input 0~3, port input of HRPWM_TRIGA~D, and external event 1~10 input. The specific condition selection can be set through the hardware capture event selection register (HRPWM<t>_HCPAR1<2>, HRPWM<t>_HCPBR1<2>). Figure 19-6 is an example of the action of capturing input.

In addition to the above hardware capture conditions, a software capture can be triggered by writing 1 to the software synchronization capture control register HRPWM _SCAPR.SCAP<t>A<B> register bit.

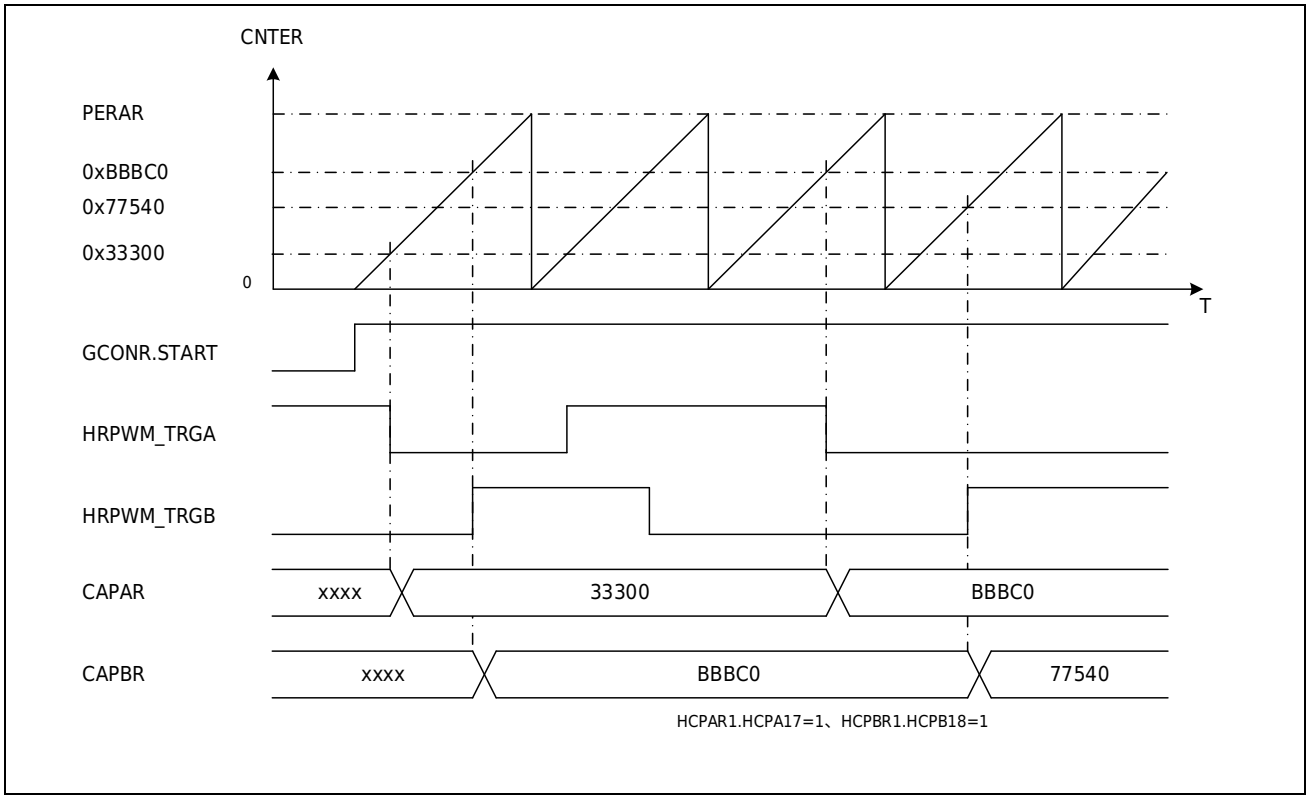


Figure 19-6 Capturing Input Actions

19.3.7 PWM Action Qualifer(AQ)

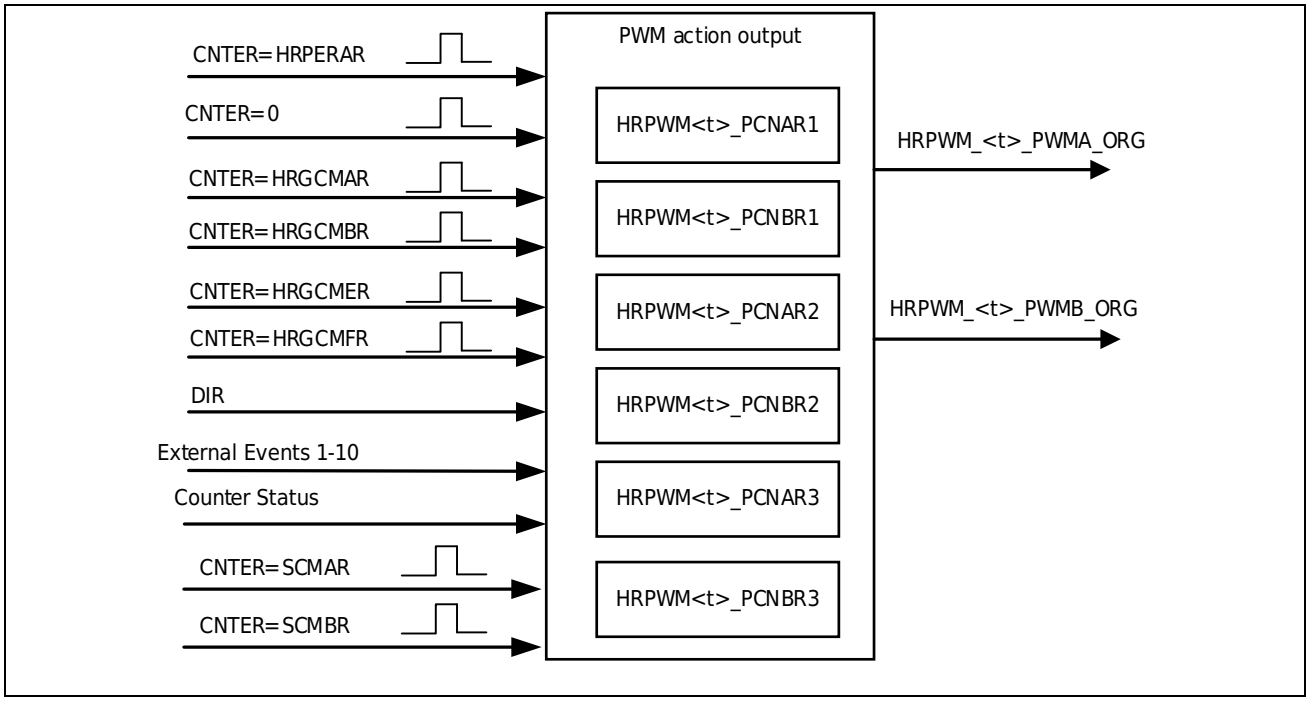


Figure 19-7 PWM Action Qualifer Module

Note:

In the above figure, CNTER indicates the value of the counter, and DIR indicates the counting direction of the counter.

The block diagram of the PWM action output module is shown in Figure 19-8. Each HRPWM unit has two output ports (HRPWM_<t>_PWMA, HRPWM_<t>_PWMB), which can be set to output a specified level when the following events occur.

- CNTER = HRGCMAR: The counter count value is equal to the High-resolution general compare A HRGCMAR
- CNTER = HRGCMBR: The counter count value is equal to the High-resolution general compare B HRGCMBR
- CNTER = HRGCMER: The counter count value is equal to the High-resolution general compare E HRGCMER
- CNTER = HRGCMFR: The counter count value is equal to the High-resolution general compare F HRGCMFR
- CNTER = SCMAR: The counter count value is equal to the special-compare SCMAR
- CNTER = SCMBR: The counter count value is equal to the special-compare SCMBR
- CNTER= HRPERAR: The counter count value is equal to the period value
- Hardware clear: hardware synchronous clear, clear event generated by master unit phase matching, write SCLRR corresponding register bit software clear; Burst Mode control counter clear is not considered as hardware clear; hardware clear is only valid in sawtooth

counter mode.

- CNTER= 0: The counter value is equal to 0
- External events 1~10
- Start counting
- Stop counting

Compare match events as well as external events can be further qualified by the direction of the counter (up or down), which is configured through the port control registers HRPWM<t>_PCNA<B>R1, HRPWM<t>_PCNA<B>R2, HRPWM<t>_PCNA<B>R3. When an event occurs, the status of HRPWM_<t>_PWMA and HRPWM_<t>_PWMB can be configured to be set to 0, set to 1, flipped or completely bypassed. Please refer to Table 18-9 for the priority order when all events occur at the same time.

Note:

After the chip is reset, only the external event in the port control register resets the PWM control to bypass, and other events are reset to 0. If you need to completely bypass an event, please configure the corresponding control register bit to the bypass state; the dedicated match event does not support high precision, and the lower 6 bits of the special-compare Register (HRPWM_<t>_SCMA<B>R) are always read as 0. When high resolution is enabled (HRPWM<t>_CR.EN=1), CNTER=SCMAR/SCMBR will have 1~2 PCLK0 jitters.

Figure 19-8 is an example of the operation of the comparison output.

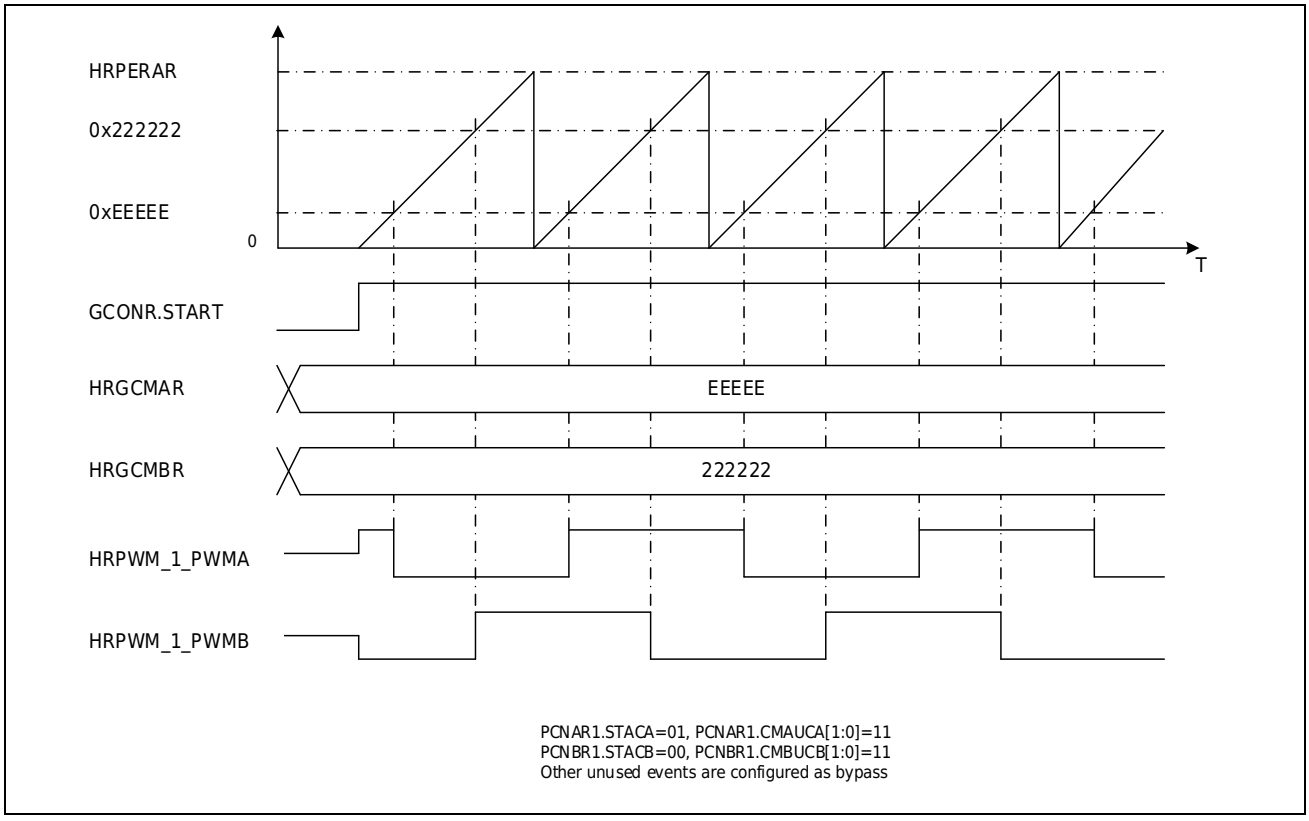


Figure 19-8 Compare Output Action

Half-wave mode

The half-wave mode is used to generate a square wave with a duty cycle of 50% at different frequencies. The half-wave function is enabled by configuring the HALF bit of the Universal control register 1 (HRPWM<t>_GCONR1) to 1. In this mode, the comparison value HRGCMAR automatically changes to (HRPERAR)/2. Do not change the HRPWM<t>_GCONR1.HALF bit after the counter is started. The PWM output example in half-wave mode is shown in the Figure 19-9.

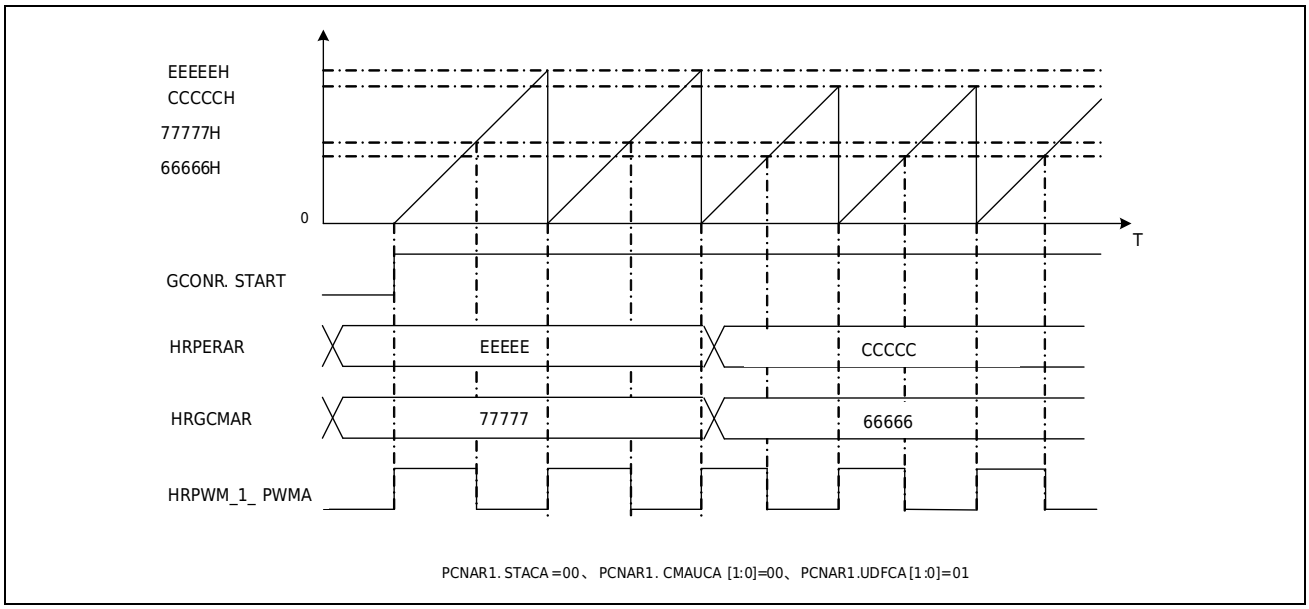


Figure 19-9 Valid PWM OutputLegend in Half-wave Mode

Duty Cycle Limitation

When the high resolution function is enabled (HRPWM<t>_CR.EN=1), there is a limit on the duty cycle of HRPWM.

In sawtooth mode: the compare value in (HRPERAR-64,HRPERAR+64) is forbidding (excluding equal to HRPERAR), otherwise an unexpected waveform will appear. The limits are shown in the Figure 19-10. If you want to set the compare value to be greater than the period value, you must set the compare value to the period value + 1 PCLK0 (HRPERAR+64) to ensure that a compare match does not occur. (When the compare value is greater than the period value, period event is not occurs)

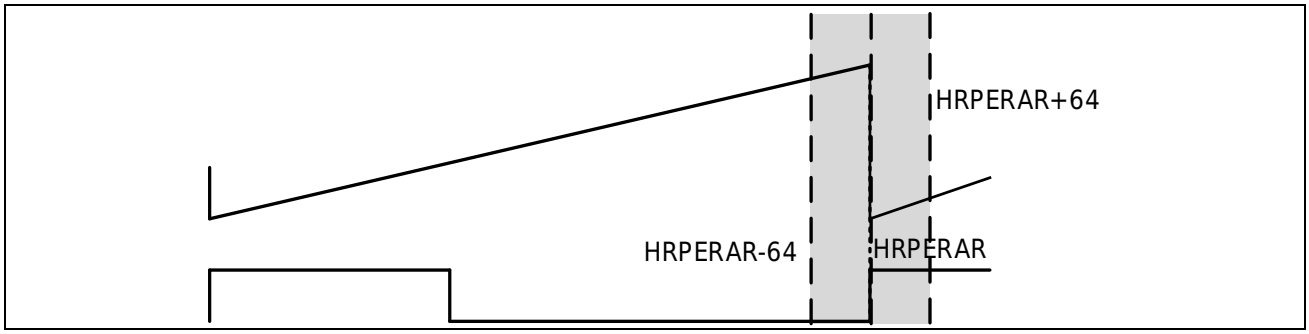


Figure 19-10 Sawtooth Wave Duty Cycle Limit 1

When two compare match events are used to control the same PWM channel, the distance between the two events must be greater than one PCLK0 cycle, otherwise unpredictable waveforms will appear.

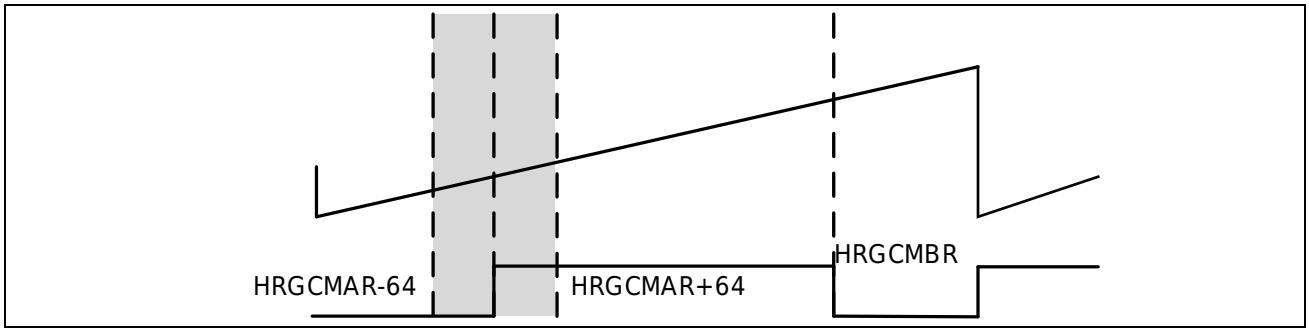


Figure 19-11 Sawtooth Duty Cycle Limit 2

In triangle count mode, t the compare value in (HRPERAR-128,HRPERAR+128) is forbidding, otherwise an unexpected waveform will appear. The limits are shown in the Figure 19-10.

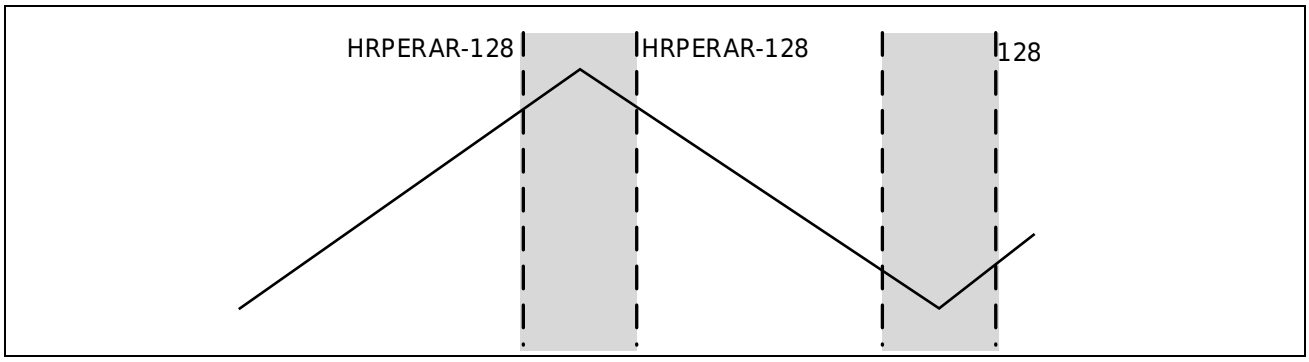


Figure 19-12 Triangle Wave Duty Cycle Limit 1

Similar to the sawtooth wave, when two compare match events control the same PWM channel, the distance between the two events must be greater than one PCLK0 cycle, otherwise an unexpected waveform will appear.

In addition to the limitations set by the comparison value above, HRPWM has further limitations on the PWM output pulse width. When the pulse width is less than or equal to 1 PCLK0 cycle, the narrow pulse width is filtered out. When the pulse width is greater than 1 PCLK0 and less than 3 PCLK0 cycles, the PWM output has a jitter of 1 to 2 PCLK0 cycles.

19.3.8 Dead Zone Module

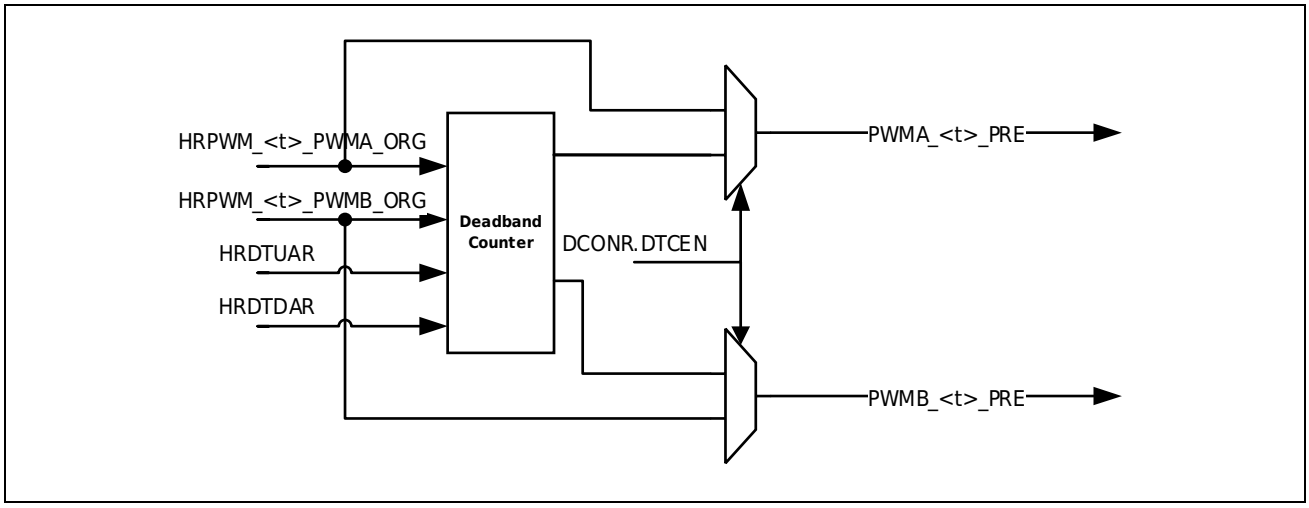


Figure 19-13 Dead Zone Module

The dead zone module is shown in the Figure 19-13. When the dead zone is enabled (HRPWM<t>_DCONR.DTCEN=1), the port level state is set when counting starts, the comparison value is equal to the comparison value, the count value is equal to 0, and the count value is equal to the period value. The PWM action output module generates the channel A internal output signal (HRPWM_<t>_PWMA_ORG), and the set value of the high-precision dead zone time reference value register (HRPWM_DTU<D>AR) is offset through timing to output a pair of complementary PWM waveforms on the PWMA_<t>_PRE and PWMB_<t>_PRE ports in hardware.

In this mode, the polarity on the PWMA_<t>_PRE output port is the same as that on the HRPWM_<t>_PWMA_ORG, and the polarity on the PWMB_<t>_PRE port is opposite to that on the HRPWM_<t>_PWMA_ORG.

If the rising edge of HRPWM_<t>_PWMA_ORG is detected, the PWMB_<t>_PRE output becomes low, the dead zone counter loads the set value of the HRPWM<t>_HRDTUAR register and starts counting down. When the count value becomes 0x0000, the counter stops and makes the PWMA_<t>_PRE output high. If the falling edge of HRPWM_<t>_PWMA_ORG is detected, the PWMA_<t>_PRE output becomes low, the dead zone counter loads the set value of the HRPWM<t>_HRDTDAR register and starts counting down. When the count value becomes 0x0000, the counter stops and makes the PWMB_<t>_PRE output high.

By setting the high-precision dead time registers HRPWM<t>_HRDTUAR and HRPWM<t>_HRDTDAR, the dead time of the output rise and fall changes can be set accordingly. Figure 19-14 is the hardware implementation of the complementary PWM output waveform.

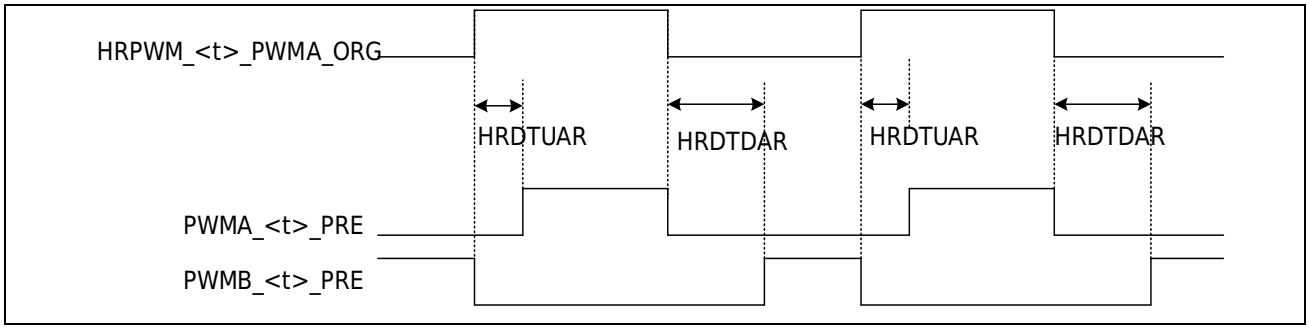


Figure 19-14 Hardware Realizes Complementary PWM Wave Output

When the high pulse width of HRPWM_<t>_PWMA_ORG is less than the dead time set by HRPWM<t>_HRDTUAR, only the PWMB_<t>_PRE output becomes low. The condition for the PWMB_<t>_PRE output level to become high is when the dead time set by the HRPWM<t>_HRDTDAR register passes after the falling edge of HRPWM_<t>_PWMA_ORG. In this case, the PWMA_<t>_PRE output will remain low continuously.

When the low-level pulse width of HRPWM_<t>_PWMA_ORG is less than the dead time set by HRPWM<t>_HRDTDAR, only the PWMA_<t>_PRE output becomes low. The condition for the PWMA_<t>_PRE output level to become high is when the dead time set by the HRPWM<t>_HRDTUAR register passes after the rising edge of HRPWM_<t>_PWMA_ORG. In this case, the PWMB_<t>_PRE output will remain low continuously. As shown in Figure 19-15 below.

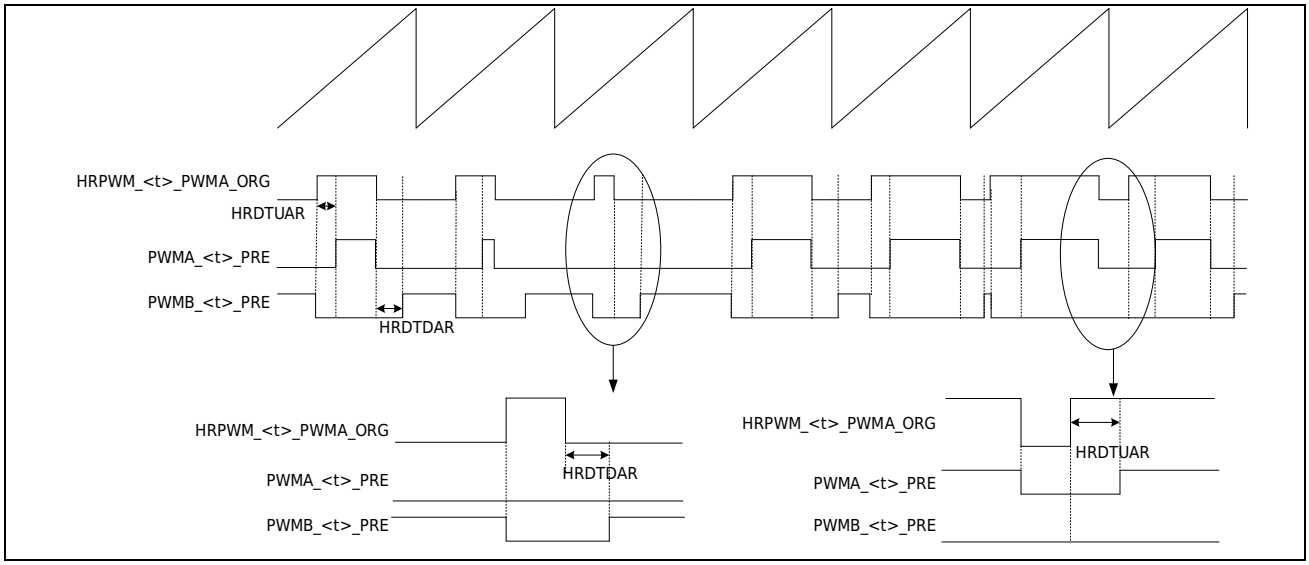


Figure 19-15 Waveform Output Under Dead Zone Mode when Pulse Width is Abnormal

19.3.9 Idle Control

The idle control level has three idle state modes:

- immediate now
- Delayed Idle
- Burst Mode Idle

The output level in the idle state can be selected as high or low by configuring the IDLESA<B> bit in HRPWM<t>_IDLECR.

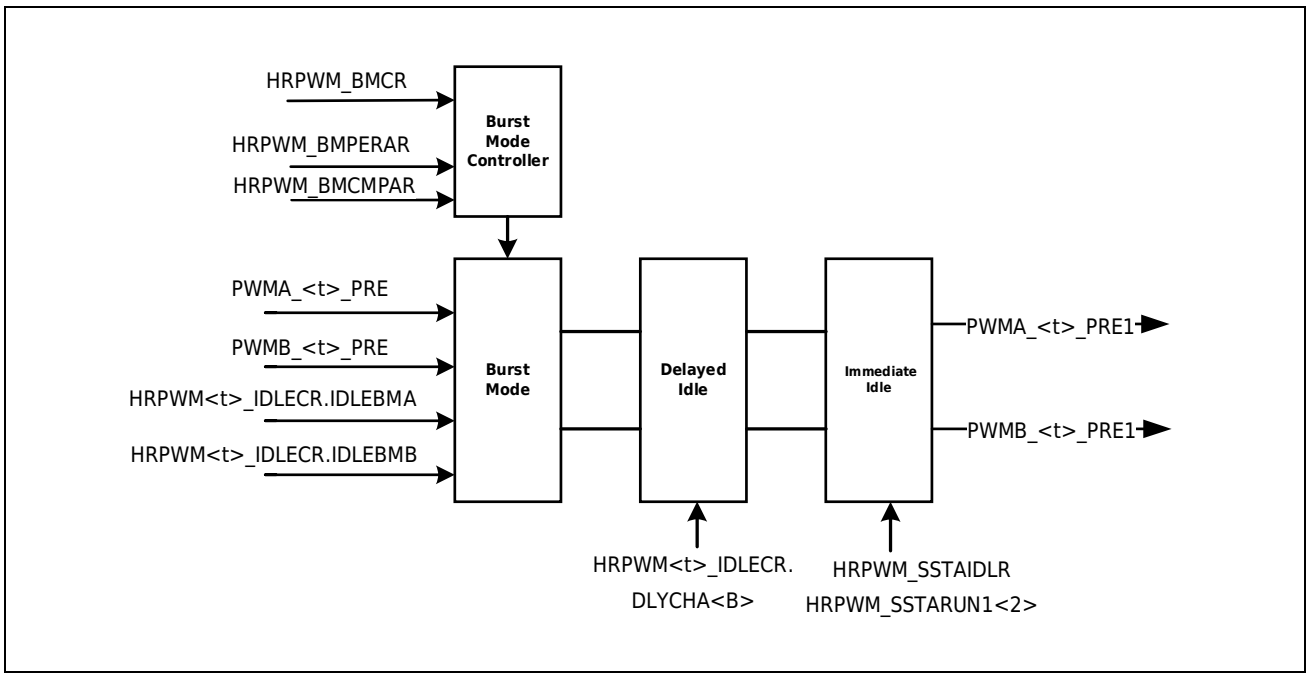


Figure 19-16 Control Basic Block Diagram

19.3.9.1 Idle Now

By writing 1 to the SSTAIDL<t>A<B> bit of the HRPWM_SSTAIDLR register, the corresponding channel can be triggered to enter the immediate idle state. By writing 1 to the SSTARUN<t>A<B> bit of the HRPWM_SSTARUNR1 register, the channel exits the immediate idle state and enters the running state. The PWM status of each unit can be queried through the HRPWM_SSTARUNR1.SSTARUN<t>A<B> bit. When HRPWM_SSTARUNR1.SSTARUN<t>A<B> is 1, it means that the current channel PWM is in the running state, and when it is 0, it is in the idle state. This bit can indicate whether the current channel PWM is in running or idle state during Burst Mode mode. After the chip is reset, HRPWM_SSTARUNR1.SSTARUN<t>A<B> defaults to 0, which is the initial output idle state.

There is another way to Exit immediate idle Exit immediate idle. You can also write 1 to HRPWM_SSTARUNR2.SSTARUN<t>A<B> to exit idle when the counter starts. This trigger signal is only valid when the corresponding unit counter stops. After the trigger signal occurs, the PWM will enter the running state when the first counter start signal encountered thereafter occurs.Refer to Figure 19-17.If an immediate idle is triggered (writing 1 to the SSTAIDL<t>A<B> bit of the HRPWM_SSTAIDLR register) between writing 1 to HRPWM_SSTARUNR2.SSTARUN<t>A<B> and starting the counter, the action of exiting the idle state when the counter is started will be canceled.

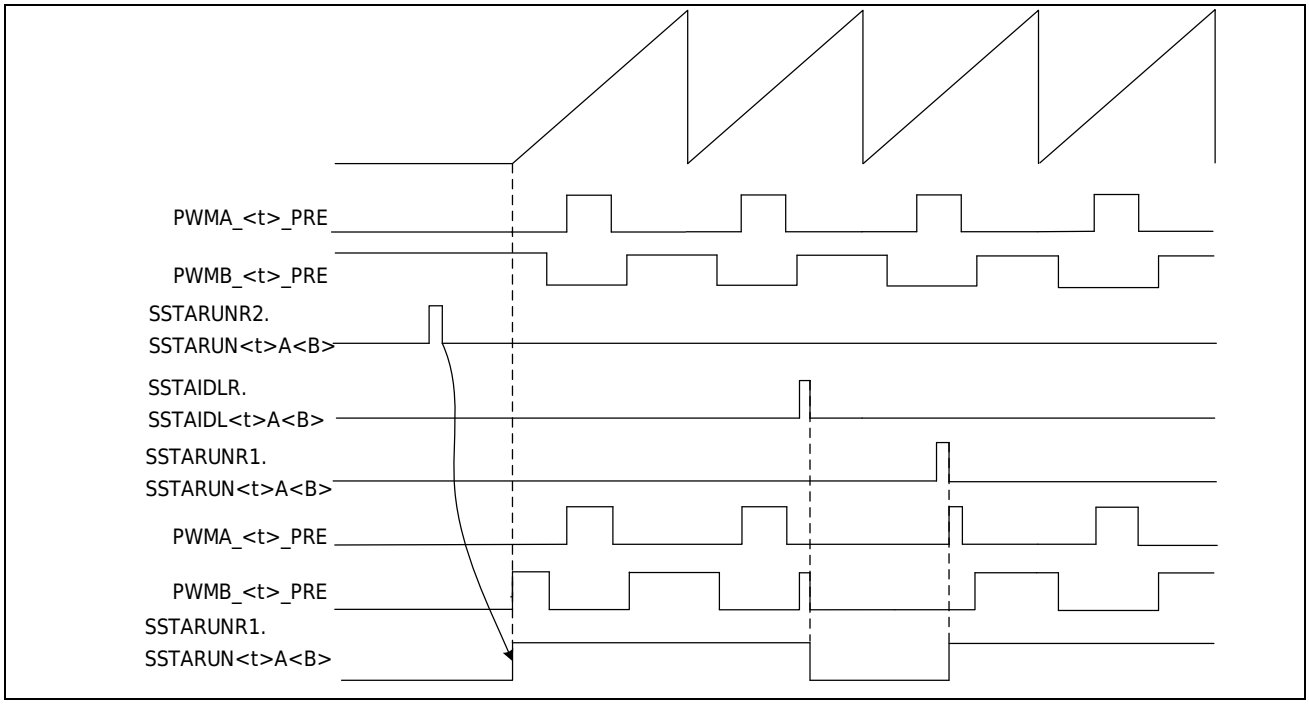


Figure 19-17 Immediate Idle Entry and Exit

19.3.9.2 Delayed Idle

This mode is used to delay PWM from entering the idle state when PWM is turned off. In this mode, the preset idle state is entered when the set idle state is detected to be consistent with the internal PWM (PWMA<B>_<t>_PRE) output state at the Full period point (defined by HRPWM<t>_GCONR1.PRDSEL), thereby ensuring the integrity of the idle state waveform. The delayed idle function is enabled by configuring the HRPWM<t>_IDLECR.DLYPRTEN bit.

Delayed idle mode supports event trigger and software trigger. The trigger source is selected by the HRPWM<t>_IDLECR.DLYEVSEL[2:0] register bit:

1. EEV6, EEV7, EEV8 and EEV9 (filtering per unit)
2. Software Triggering

The software trigger can be triggered by writing 1 to HRPWM<t>_IDLECR.DLYSTRG of each unit or writing 1 to HRPWM_SSTADIDLR. DLYSTRGU<t>.

Output A and output B have separate delayed idle enable bits, which can select to enter or not enter the delayed idle state when the delayed idle trigger source arrives.

Once delayed idle is triggered, when HRPWM_<t>_PWMA<B> enters the idle state, the DLYIDLEA<B> status flag bit in HRPWM<t>_STFLR2 is set to 1, and the idle mode will be maintained permanently until HRPWM_SSTARUNR1.SSTARUN<t>A<B> is written to 1 to trigger exit from delayed idle, and the counter continues to run in the delayed idle state. The entry of Delayed Idle does not affect the state of HRPWM_SSTARUNR1.SSTARUN<t>A<B>. When exiting delayed idle, you need to first confirm that the HRPWM<t>_DLYIDLEA<B> bit is 1, and then write 1 to HRPWM_SSTARUNR1.SSTARUN<t>A<B> to exit delayed idle. After writing 1, it will not exit the idle

state immediately, but after the Full period point (defined by the HRPWM_<t>_GCONR1.PRDSEL bit), it will detect that the set idle state is consistent with the internal PWM (PWMA<B>_<t>_PRE) output state, and then exit the idle state and exit the delayed idle, As shown in Figure 19-18.

once an external event arrives or a software trigger occurs, the DLYPRT flag bit of the HRPWM<t>_STFLR2 register is set and a delayed idle trigger interrupt is generated when the INTENDLYPRT bit of HRPWM<t>_IDLECR is set.

The OASTAT and OBSTAT bits of the HRPWM<t>_STFLR2 register can be used to confirm the state of the PWM output when the external event triggers the delayed idle. When the register bit is 1, it indicates that the PWM is in a high state when the delayed idle is triggered, and when it is 0, it indicates that the PWM is in a low state.

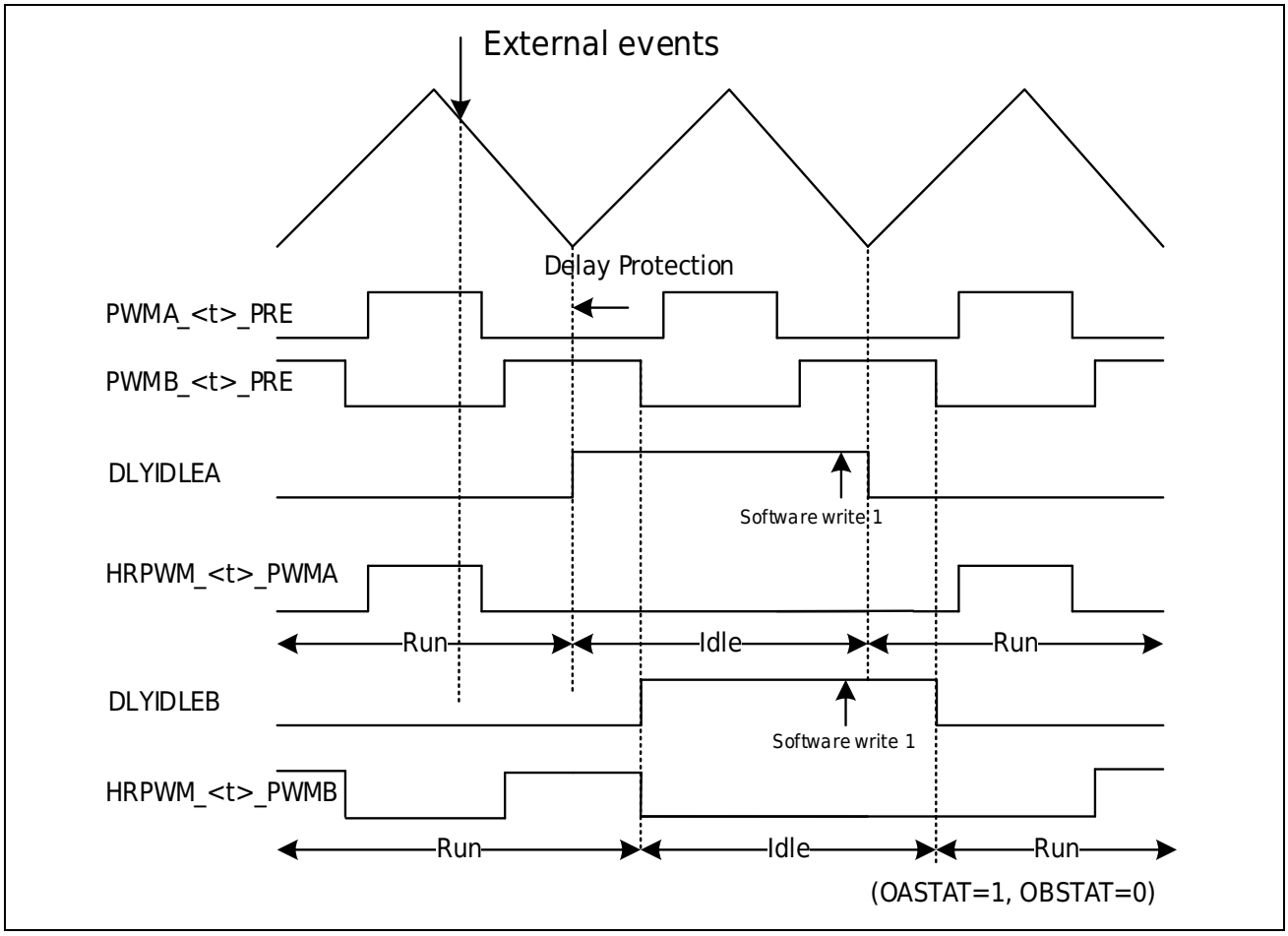


Figure 19-18 Delayed Idle Entry and Exit

The delayed idle mode has a higher priority than the burst mode: any burst mode exit request is discarded once the delayed idle protection has been triggered. On the contrary, If the delayed protection is exited while the burst mode is active, the burst mode is resumed normally and the output is maintained in the idle state until the burst mode exits.

The Burst Mode controller allows the HRPWM_<t>_PWMA<B> to alternately output the idle and run states through hardware control. This mode is enabled by the BMEN bit in the HRPWM_BMCR register.

In the Burst Mode idle state, the port state of HRPWM_<t>_PWMA<B> can be set to be unaffected or to the level of the output idle state (configure HRPWM<t>_IDLECR.IDLEBMA<B> bit).

Burst Mode controller includes:

- 1 counter (BM-counter)
- 1 compare register: HRPWM_BMCMAR, used to define the duration of the idle state
- 1 period register: HRPWM_BMPERAR, used to define the sum of the idle state and the running state

Note:

When HRPWM_BMCR.BMMD is set to 1, the duration of the idle state is 0 BM-counter count clock cycles when HRPWM_BMCMAR is set to 0; when HRPWM_BMCR.BMMD is set to 0, the duration of the idle state is 1 BM-counter count clock cycle when HRPWM_BMCMAR is set to 0.

BM-counter is a 16-bit up counter, which supports continuous mode and single mode. When HRPWM_BMCR.BMCTN=1, the BM counter operates in continuous mode and the Burst Mode process continues until the HRPWM_BMCR.BMOPTF bit is 0. When HRPWM_BMCR.BMCTN = 0, the BM counter operates in one-shot mode and stops counting when the counter reaches the period value. When the BM-counter reaches the period value, BMOVFF in HRPWM_BMCR is set.

During the Burst Mode operation (running, idle), the counter of each unit can be stopped and reset by setting the HRPWM_BMCR.BMTMR1~6 bits. The counter stops counting and clears only in the Burst Mode idle state, and operates normally in the running state. When BMTMR1~6 in the Burst Mode control register is 0, the counter in the corresponding unit is not affected and counts normally; when BMTMR1~6 in the Burst Mode control register is 1, the counter in the corresponding unit stops in the Burst Mode idle state and is cleared.

Note:

Do not set the corresponding unit BMTMR<t> to 1 when the PWM is enabled in the Burst Mode delay mode.

The counting clock of BM-counter can select multiple clock sources, which can be selected by setting HRPWM_BMCR.BMCLKS[3:0]. There are three main types of clock sources: the Full period point of units 1 to 6, the comparison event of TMR0, or the division of PCLK0. When the count clock is selected as the division of PCLK0, the division coefficient is set by the HRPWM_BMCR.BMPSC[3:0] bits. The duration of the Burst Mode idle state is defined by HRPWM_BMPCMAR and the period of the Burst Mode is defined by HRPWM_BMPERAR.

The action mode of the Burst Mode can be set by configuring the HRPWM_BMCR.BMMD bit. When HRPWM_BMCR.BMMD is 0, the Burst Mode works in mode 1, and the duration of the idle time of the Burst Mode is HRPWM_BMPCMAR+1 Burst Mode count clock cycle. When the trigger signal occurs, the Burst Mode first enters the idle state, and enters the running state when BM-counter is equal to HRPWM_BMCMAR; when HRPWM_BMCR.BMMD is 1, the Burst Mode works in mode 2, and the duration of the idle time of the Burst Mode is HRPWM_BMPCMAR Burst Mode count clock cycles. When the trigger signal occurs, the Burst Mode first enters the running state of 1 Burst Mode count clock cycle, and then enters the idle state. The Burst Mode operation status in the two modes is shown in Figure 19-19 and Figure 19-20.

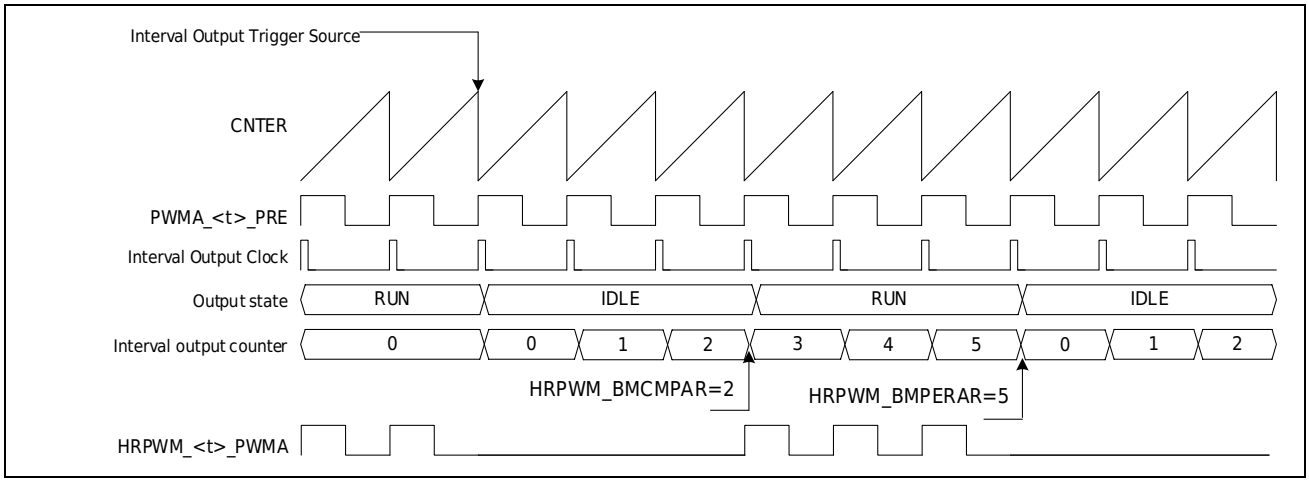


Figure 19-19 Timing Diagram of Burst Mode Mode

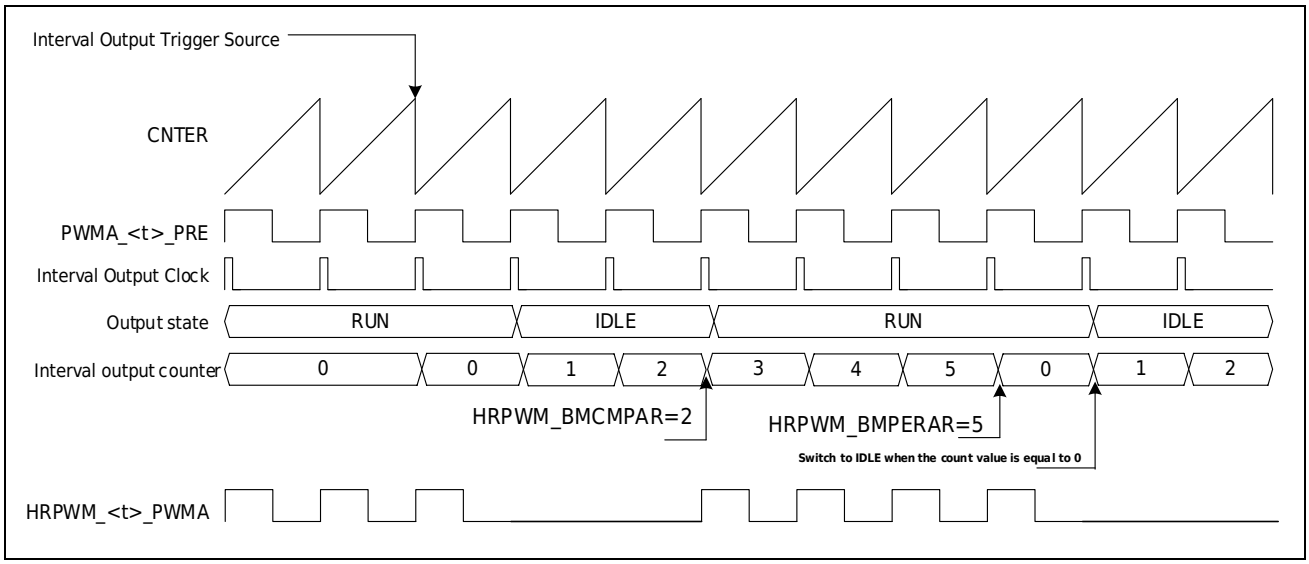


Figure 19-20 Timing Diagram of Burst Mode Mode 2

Burst Mode Trigger Source

There are several types of Burst Mode trigger sources, which are selected through register HRPWM_BMSTRG1<2>:

- Software Triggering
- 6×4 events from each unit: count value equal to 0, count value equal to period value or hardware clear in sawtooth counter mode, count value equal to High-resolution general compare HRGCMAR and count value equal to High-resolution general compare HRGCMBR
- TMR0 comparison event
- External event 7 (filtered by unit 1) and external event 8 (filtered by unit 4)
- The Unit1's Period event(or hardware clear in sawtooth counter mode) or ZERO event, after External event 7;
- The Unit4's Period event(or hardware clear in sawtooth counter mode) or ZERO event, after External event 8;

Multiple trigger sources can be selected and enabled at the same time. During the Burst Mode operation (idle and running), the Burst Mode cannot be triggered again.

Burst Mode Entry and State Switching

When entering the Burst Mode, switching from run to idle, or from idle to run during the interval mode, there are two modes: Regular Mode and delay mode.

Regular Mode

When the HRPWM<t>_IDLECR.DIDLA<B> bit is 0, it is the regular entry mode of Burst Mode. When a trigger event arrives, the port of HRPWM_<t>_PWMA<B> will immediately enter the preset idle state; during the Burst Mode period, when switching from idle to running or running back to idle, the port of HRPWM_<t>_PWMA<B> will immediately enter idle or run according to the state of the Burst Mode controller. The following figure is an example of entering Burst Mode triggered by an external event or a Full period point after an external event (defined by HRPWM<t>_GCONR1.PRDSEL) and state switching during Burst Mode.

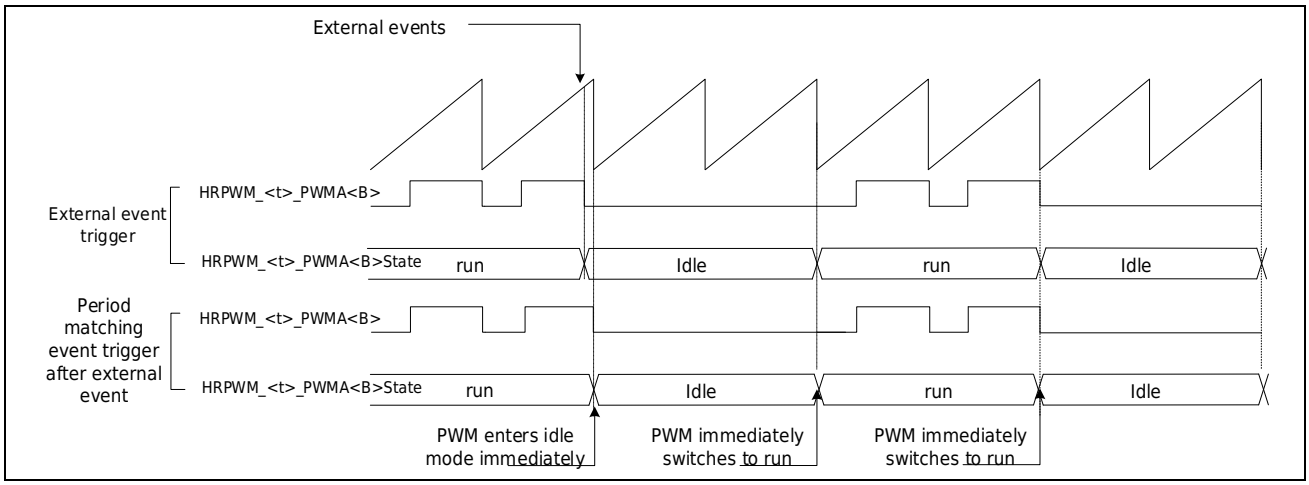


Figure 19-21 Mode for Plot Burst Mode

Delayed Mode

When the HRPWM<t>_IDLECR.DIDLA<B> bit is 1, it is in delay mode for Burst Mode. When entering Burst Mode and switching states during Burst Mode, the PWM switches between idle and run when it detects that the set idle state is consistent with the internal PWM (PWMA<B>_<t>_PRE) output state after the Full period point(defined by HRPWM<t>_GCONR1.PRDSEL) occurs. In delayed mode, the Burst Mode idle or running time must be greater than or equal to one PWM period.

Figure 19-22 is a diagram of entering Burst Mode in delay mode, and Figure 19-23 is a diagram of state switching during Burst Mode in delay mode.

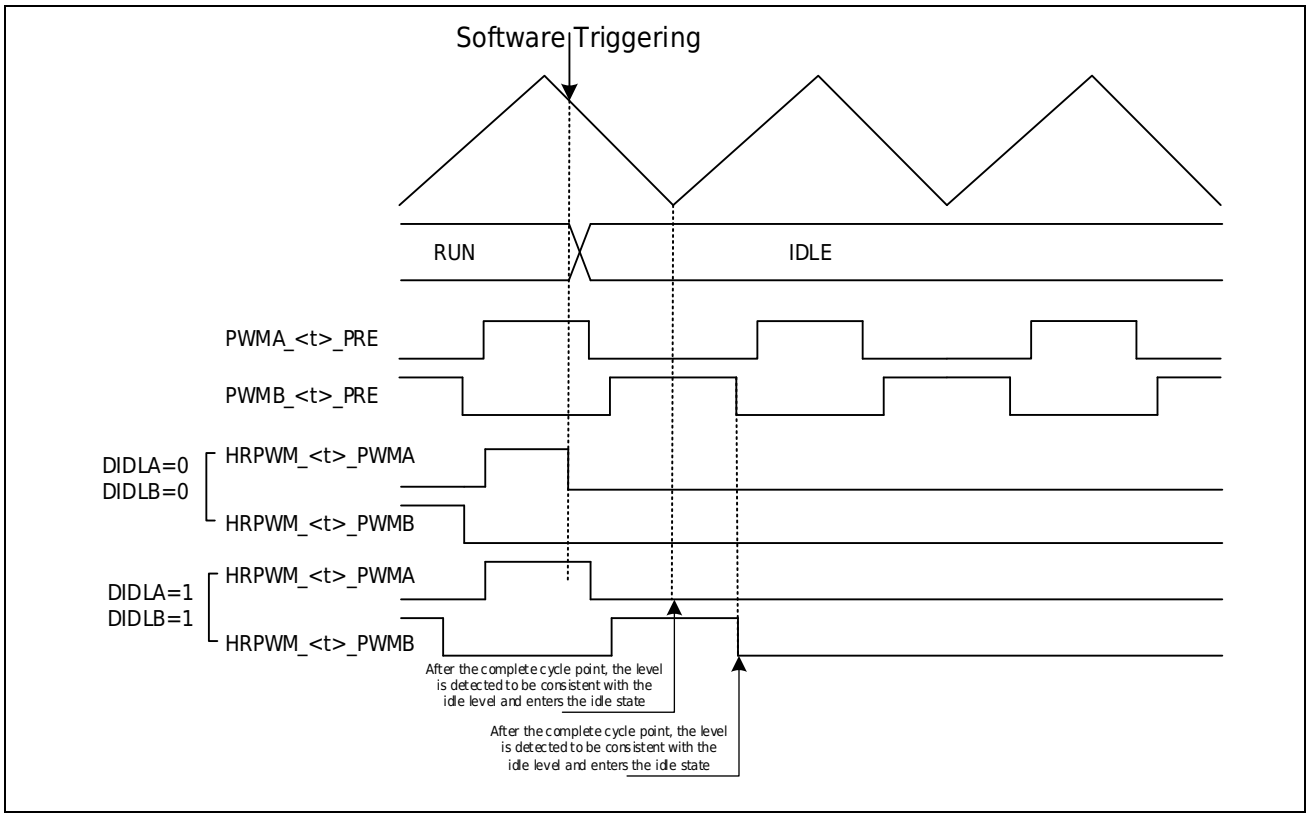


Figure 19-22 Delay Mode Valid Burst Mode Enter Legend

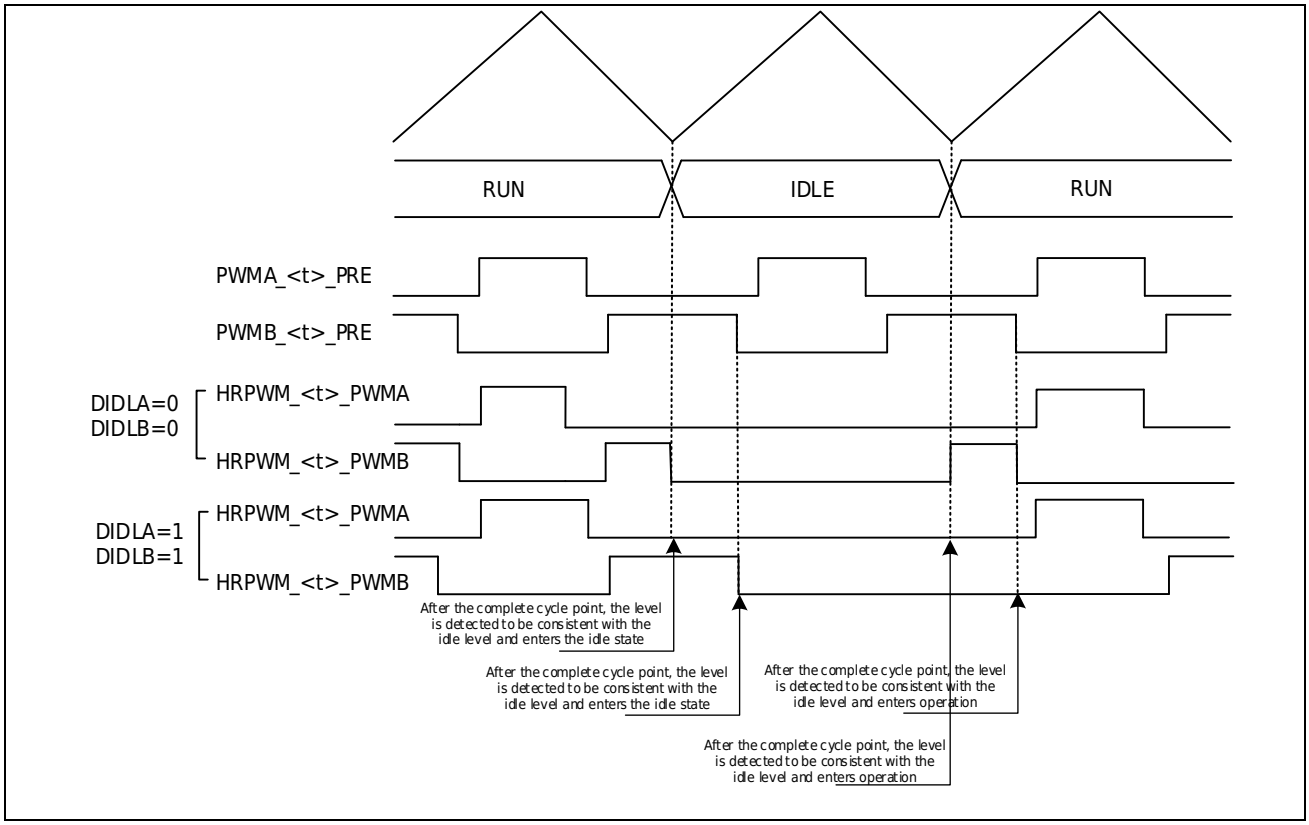


Figure 19-23 Legend of State Switching During Delay Mode Valid Burst Mode

Exit of Burst Mode

In continuous mode, the Burst Mode is forced out by software; in single-shot mode, the Burst Mode is exited once the idle period has passed. In continuous mode, write 0 to HRPWM_BMCR.BMOPTF or BMEN bit to exit the Burst Mode. When exiting, the idle state will be detected after the complete cycle point occurs and the set idle state is consistent with the internal PWM (PWMA<B>_<t>_PRE) output state, and then exit the Burst Mode idle state, As shown in Figure 19-24.

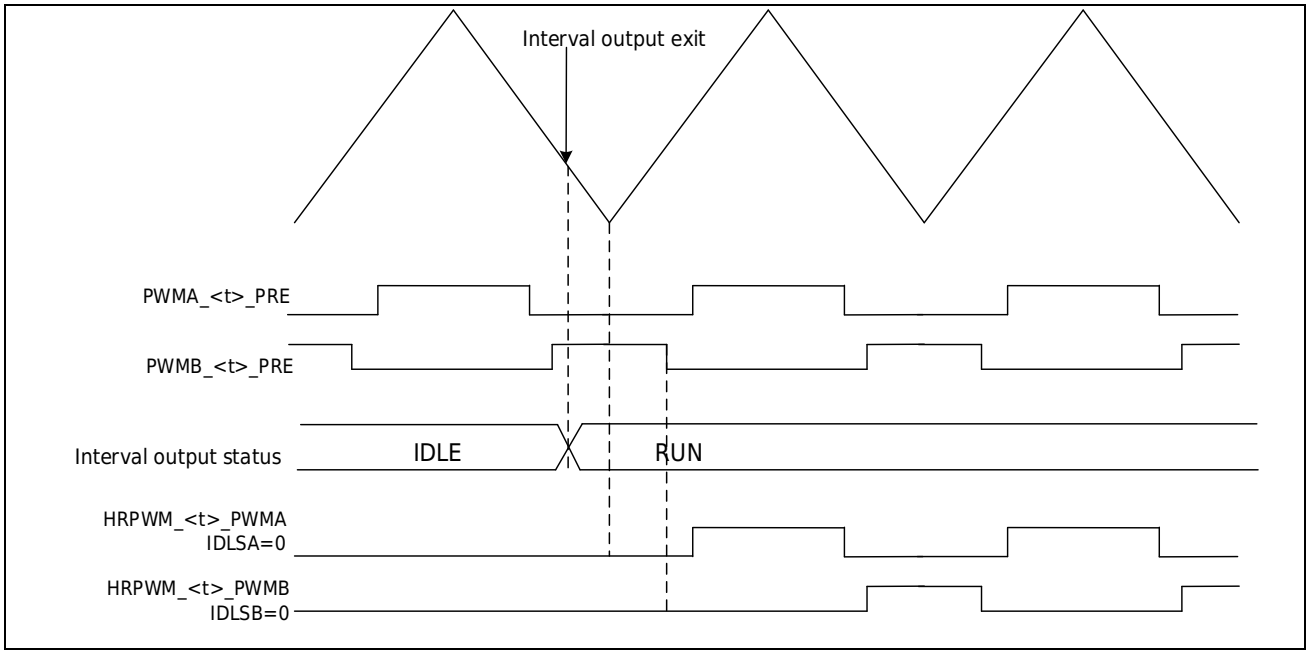


Figure 19-24 Exit of Burst Mode

Note:

After writing 0 to HRPWM_BMCR.BMOPTF or BMEN bit to exit Burst Mode, the Burst Mode cannot be triggered again before all channels with Burst Mode Function Enable exit. HRPWM_BMCR.BMOPTF will be cleared by hardware only after all channels exit Burst Mode.

Burst Mode Follow Function

The ports of HRPWM_<t>_PWMB can follow PWM<A>_<t>_PRE in and out of idle when entering, exiting, and switching states during Burst Mode. This function valid only when the idle level settings of PWMA and PWMB are the same (HRPWM<t>_IDLECR.IDLESA = HRPWM<t>_IDLECR.IDLESB), and the delay mode of both channel A and channel B is enabled (HRPWM<t>_IDLECR.DIDLA<B> = 1). In this mode, when channel A enters or exits Burst Mode or switches between idle and run during Burst Mode, it still enters idle or exits idle when the level consistency is detected after the complete cycle point. The moment when channel B enters idle occurs when channel A detects PWM<B>_<t>_PRE flips after entering, and the moment when channel A exits idle occurs when channel A detects PWM<B>_<t>_PRE flips after exiting. For specific exit time, refer to Table 19-4. Figure 19-25 is a diagram showing how channel B enters and exits the idle state when the idle levels of channels A and B are both set to low and the follow function is enabled.

Table 19-4 Channel B Follow Function Exits and Enters Idle Time

HRPWM<t>_IDLECR.IDLESA	HRPWM<t>_IDLECR.IDLESB	Channel B Exit Time
0	0	PWM<B>_<t>_PRE falling edge
1	1	PWM<B>_<t>_PRE rising edge
Other		Level consistent entry and exit detected after Full period point

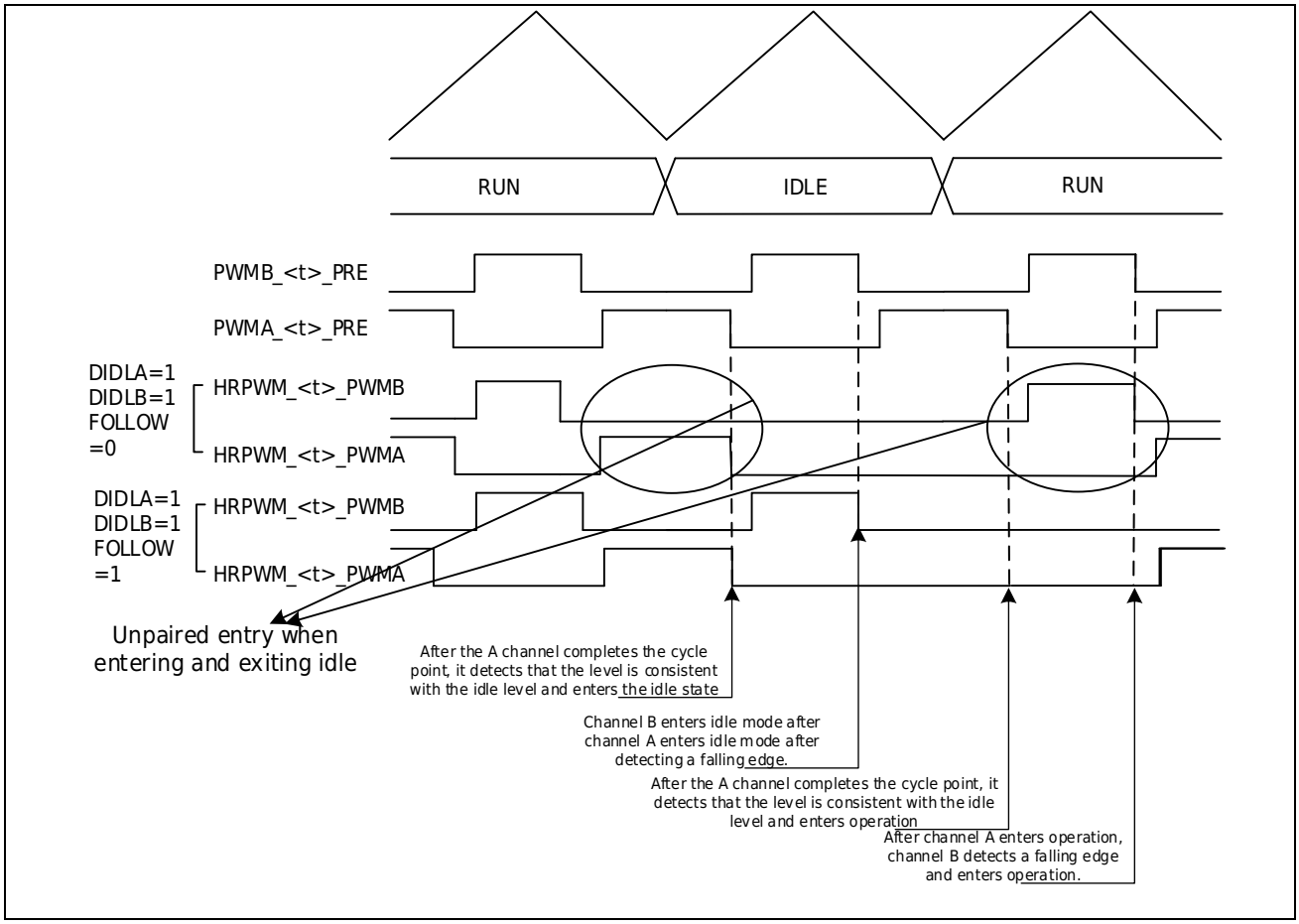


Figure 19-25 Channel B Follow Function Exit and Enter Idle Legend

19.3.10 Phase Function

The phase function of HRPWM can be used to achieve multi-phase interleaved output. HRPWM includes 6 counting units, among which unit 1 can be used as the master unit to synchronize units 2~6.

Unit 1 includes five high-resolution phase Compare Value Registers HRPWM<t>_PHSCMP1~5A, which can generate a phase match event when the count compare matches.

The 5 phase Compare Registers all have a Shadow function. The Shadow function can be enabled by setting the BENPHS bit in the unit 1 phase control register (HRPWM<t>_PHSCTL), Shadow Function Section.

Units 2 to 6 act as slave units. When PHSEN in the phase register control register (HRPWM<t>_PHSCTL) is enabled, one of the five phase match events of unit 1 is selected as the phase match event by setting the HRPWM<t>_PHSCTL.PPHCMPSEL bit. The slave unit is enabled and cleared on a phase match event. Figure 19-26 is an example of starting and clearing the counter of the multi-phase PWM master and slave units in triangle count mode. In this example, the slave unit is configured in single counting mode. Figure 19-27 is an example of starting and clearing the

counter of the multi-phase PWM master and slave units in sawtooth wave mode. In this example, the slave unit period value is set to the maximum value.

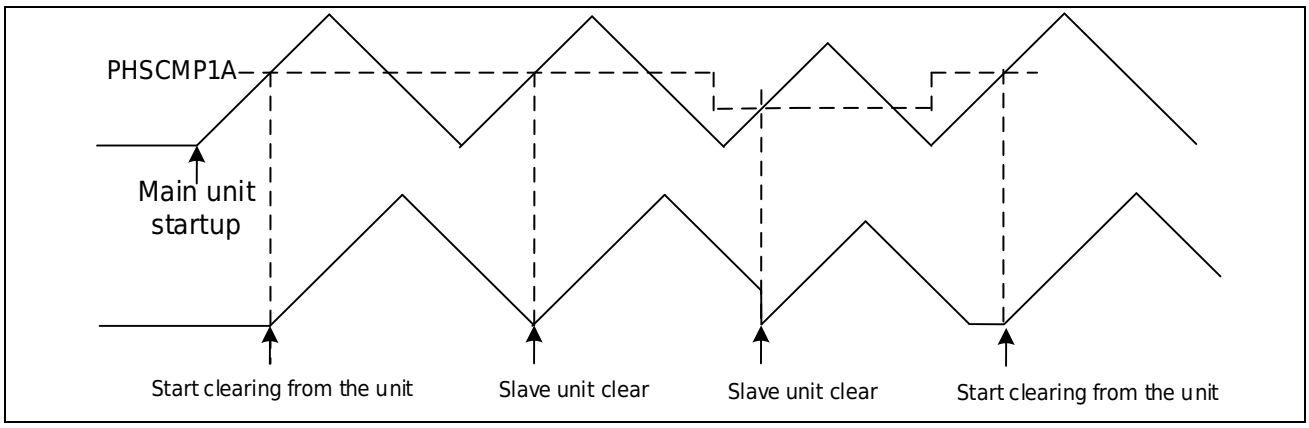


Figure 19-26 Triangle count mode Multi-phase PWM Output Slave Unit Startup and Clearing Action

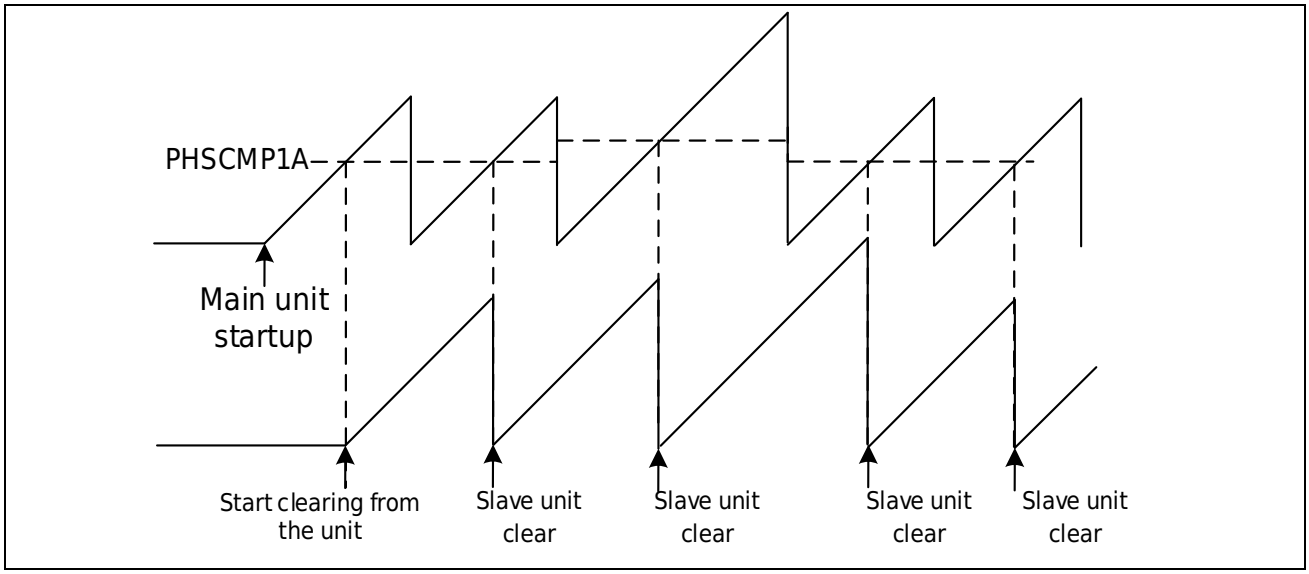


Figure 19-27 Sawtooth Wave Mode Multi-phase PWMOutput Slave Unit Startup and Clearing Action

During multi-phase PWM output, dynamic adjustment of period, duty cycle, dead zone, etc. can be achieved through the Shadow functions of high-resolution period value register, high-resolution general compare register and high-resolution phase comparison reference value. For details, please refer to [Multi-Phase Interleaved PWM Output Between Units]. The slave unit can configure the sawtooth wave and triangle count modes bysetting the PHSFORCA<B> bit of HRPWM<t>_PHSCTL, and force the PWM to be low to avoid continuous waves when dynamic adjustments are made during multi-phase PWM output. In triangle count mode, enabling the corresponding bit can configure PWM to a low level when a phase match event occurs; in sawtooth wave mode, enabling the corresponding bit can configure PWM to a low level when the current count value is greater than or equal to the value in the Shadow Comparison Reference Value Register at the unit 1 Single Shadow Transfer Time. The above functions are only valid when the phase function is enabled

(HRPWM<t>_PHSCTL.PHSEN=1). For details, please refer to the[Multi-Phase Interleaved PWM Output Between Units] section.

There is a 1 PCLK0 delay between the master phase match event and the enabling and clearing of the slave. Set the high-resolution phase Comparison Reference Value Register PHSCMP1~5A according to the following formula:

-
-
- N = number of phases
- M=PWM unit number (M=2~6)

Period link

Configure HRPWM<t>_GCONR1.PRDLK to 1 to enable the period link function (only units 2 to 6). In this mode, the value of the unit period value register HRPWM<t>_HRPERBR (t=2~6) is consistent with the period value of unit 1.

19.3.11 External Events

HRPWM has 10 groups of external events that can be used for all units. Configure external events through HRPWM_EECR1 and HRPWM_EECR2 registers:

- Select the source of the external event by configuring EEySRC (y=1~10) (4 sources can be selected for each external event), see Table 19-5.
- By configuring EEySNS[1:0], select the effective edge of the external event, which can be set to level sensitive or edge sensitive (rising, falling, or both edges)
- When level sensitivity is selected, the external event level polarity is selected by the EEyPOL bit.

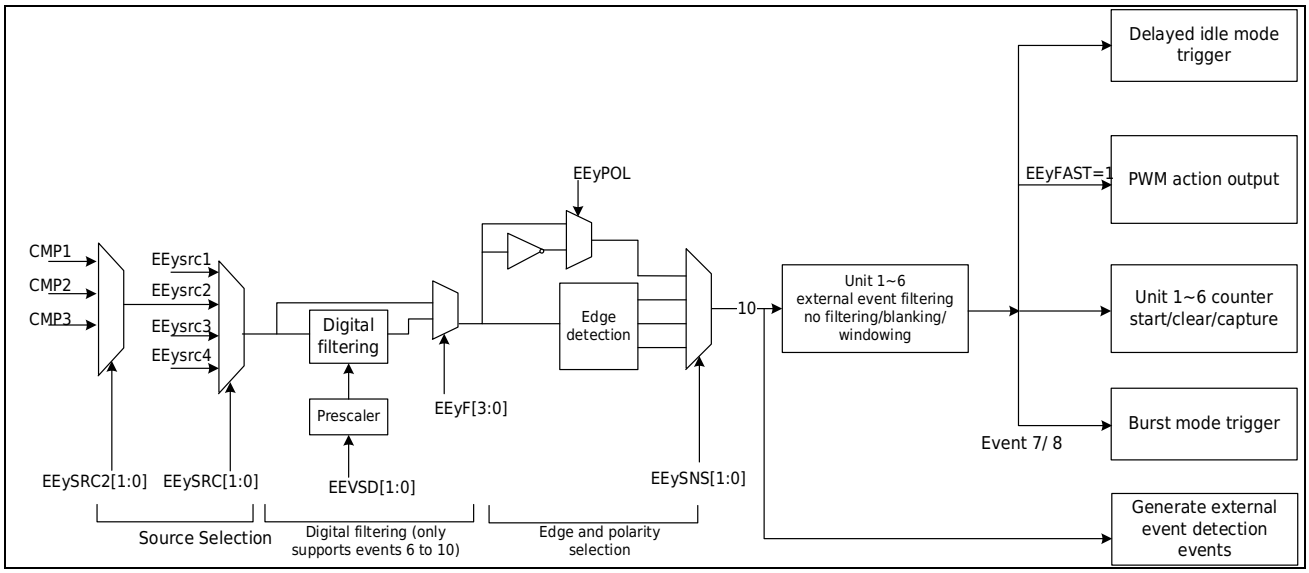


Figure 19-28 External Event Processing Block Diagram

These external events can be used to trigger hardware start, clear, update counters, Burst Mode, delayed idle, and change port state(according AQ Module), and can also generate external event detection events.

Note:

External events are used to trigger hardware start, clear, update counter, capture, Burst Mode, delayed idle, and generate external event detection events. They are only edge sensitive, even if the EEySNS bit is set to 0b00 (level sensitive). When EEyPOL=0, a trigger signal is generated when a rising edge is detected, and when it is 1, a trigger signal is generated when a falling edge is detected. When HRPWM_GCTLR.BKDLY is set to 1 and the level valid, the **Blanking DeLaY** function will be enabled. When the blanking time in the unit is completed and the external event is detected as a valid level, a trigger signal will be generated.

External events 1 ~ 5 have fast asynchronous mode, which is configured by the HRPWM_EECR1.EEVyFAST register bit. The fast asynchronous mode only supports level validity. When fast asynchronous is enabled, if the corresponding external event is configured as edge validity, the external event will have no fast asynchronous function.

The external event fast asynchronous mode only supports the control of PWM setting Low and setting High, does not support the toggle function. Do not set HRPWM<t>_PCNA<B>R2.EXEVyCA<B> to 11 when the fast asynchronous mode is enabled. When using fast asynchronous mode in triangle count mode, the external event counting up and down actions must be set to be consistent.

The external event fast asynchronous mode supports the Filtering function. When the width of the external event after Filtering is less than 6 clock cycles, the PWM cannot enter the level state configured by the external event for a maximum of 3 clock cycles. External events 6~10 have digital filtering function, which is configured through the HRPWM_EECR3 register.

The sampling clock can be configured as the divided frequency of PCLK0, or fEEVS divided from PCLK0. fEEVS is set by the EEVSD[1:0] bits of HRPWM_EECR3.

The EEyFM[3:0] bits can select the filter sampling clock and filter length N. When the filter sampling reference clock samples the same level on the port N times, the level is regarded as a valid level; a level less than N times consistent will be filtered out as external interference.

The selectable sources of the 10 external events are as follows:

Table 19-5 External Event Mapping Table

External events	EEysrc1	EEysrc2	EEysrc3	EEysrc4
External Event 1	HRPWM_EEV1	EE1src2	TMR0_1CMPA	ADC1_CMP0
External Event 2	HRPWM_EEV2	EE2src2	TMR0_1CMPB	ADC1_CMP1
External Event 3	HRPWM_EEV3	EE3src2	TMR0_1CMPA	-

External events	EEysrc1	EEysrc2	EEysrc3	EEysrc4
External Event 4	HRPWM_EEV4	EE4src2	-	ADC2_CMP0
External Event 5	HRPWM_EEV5	EE5src2	-	ADC2_CMP1
External Event 6	HRPWM_EEV6	EE6src2	TMR0_2CMPA	-
External Event 7	HRPWM_EEV7	EE7src2	TMR0_2CMPB	ADC3_CMP0
External Event 8	HRPWM_EEV8	EE8src2	-	ADC3_CMP1
External Event 9	HRPWM_EEV9	EE9src2	TMR0_1CMPB	-
External Event 10	HRPWM_EEV10	EE10src2	-	-

The source of EEysrc2 in Table 19-5 can be further selected by the HRPWM_GCTLR.EEySRC2[1:0] register bits, refer to the table below.

Table 19-6 External Event EEysrc2 Selection

EEySRC2[1:0]	EEysrc2
00	CMP1
01	CMP2
10	CMP3
11	No Event Selected

19.3.11.1 External Event Filtering Function of Units

10 external events are available to all units. They can be used directly by each unit or within a limited time. In each unit, 10 external events can be selected through the external event filter control register HRPWM<t>_EEFLTCR1<2>.EEyFM (y=0~9) to select the following two event filter modes:

Blanking mode: External events occurring at a specified time are ignored

Windowing mode: External events that occur at a specified time are effective

The blanking signal in blanking mode and the windowing signal in windowing mode can be generated by the **External Event Filter OFFSET Register**(HRPWM<t>_EEFOFFSETR) of this unit, the **External Event Filter WINdow Register** (HRPWM<t>_EEFWINR) and its corresponding **External Event Filter Initial level Register** (HRPWM<t>_EEFLTCR1.EEINTPOL) (defining the initial level of the blanking signal and windowing signal when the counter starts counting), or the blanking or windowing signal generated by other units can be selected. In blanking mode, the blanking signal can also be selected as PWMA<B>_<t>_PRE1 (PWM after idle Module) of other units.

The offset value HRPWM<t>_EEFOFFSETR.OFFSET and the window value HRPWM<t>_EEFWINR.WIN are both defined as the count time starting from counter 0, and the count value is (OFFSET+64) or (WIN+64) to take effect. HRPWM<t>_EEFOFFSETR.OFFSETDIR is

defined as the offset value counting direction, and HRPWM<t>_EEFWINR.WINDIR is defined as the window value counting direction.

In blanking mode, when the counter counts in the offset value counting direction and the count value reaches the offset value, the blanking signal is reset; when the counter counts in the window value counting direction and the count value reaches the window value, the blanking signal is set.

In windowing mode, when the counter counts in the offset value count direction and the count value reaches the offset value, the window signal is set; when the counter counts in the window value count direction and the count value reaches the window value, the window signal is reset.

When the window value and offset value occur at the same time, the offset value has a higher priority. The initial level when the counter starts counting is set by HRPWM<t>_EEFLTCR1.EEINTPOL. After the counter stops, no filtering is performed.

Note:

When triangle Count Mode, when OFFSET or WIN is set to the period value, please set the corresponding OFFSETDIR or WINDIR to 0; when OFFSET or WIN is set to 0, please set the corresponding OFFSETDIR or WINDIR to 1; when sawtooth Count Mode, please set both the corresponding OFFSETDIR and WINDIR to 1; the external event filter window value and offset value do not support high precision.

Blanking Mode

In blanking mode, external events occurring within the specified time will be ignored, and external events occurring at other times will be valid. The specified time of the blanking mode refers to the time it takes for the counter to count from OFFSET to WIN, As shown in the Figure 19-30.

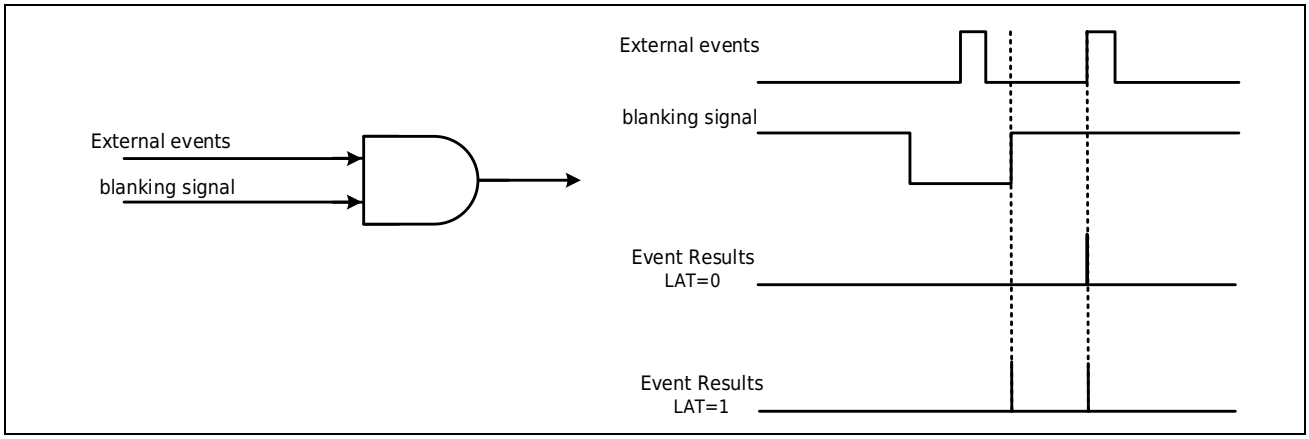


Figure 19-29 Blanking Mode

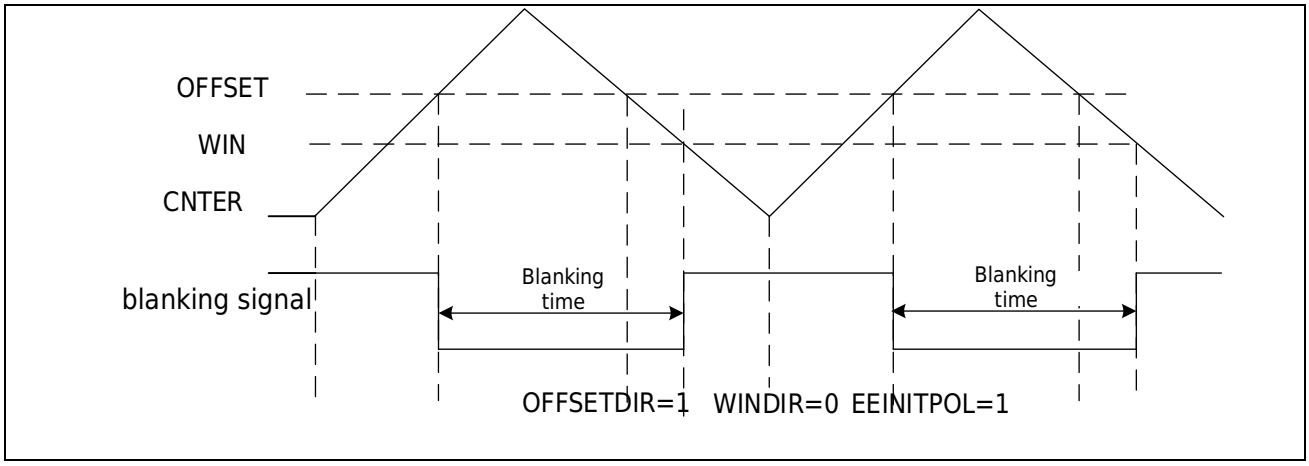


Figure 19-30 Blanking Time

In blanking mode, when HRPWM<t>_EEFLTCR1<2>.EVE1yLAT is set to 1, the **event postpone mode** will be enabled ,a the external event is not taken into account immediately but is

memorized (latched) and generated as soon as the blanking period is completed, the event postpone Mode as shown in the Figure 19-29.

In blanking mode, the external events used to trigger hardware start, clear, update counter, capture, Burst Mode, and delayed idle have a **Blanking DeLaY** function. When HRPWM_GCTLR.BKDLY<t> (t=1~6) is set to 1, the **Blanking DeLaY** function will be enabled. When the external event selection level valid, a trigger signal will be generated when the external event is at a valid level when the specified time is completed. This function is only valid when the **event postpone mode** invalid. As shown in Figure 19-31, if the external event valid level is set to high level, the external event is detected as high level at the end of the blanking window, and a trigger signal is generated at this time.

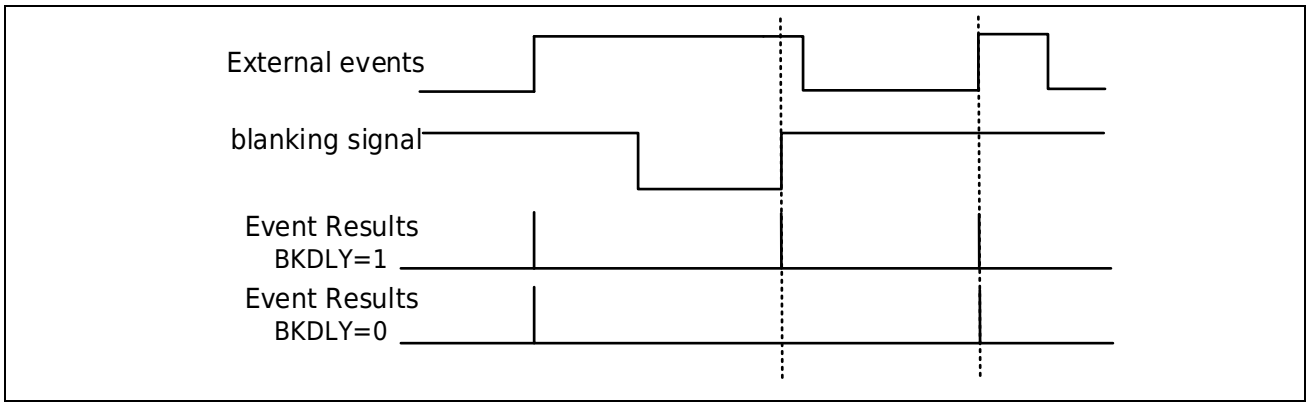


Figure 19-31 External Event Delay Function in Blanking Mode

Blanking signals include the following sources:

Local unit: (HRPWM<t>_EEFLTCR1<2>.EEyFM[3:0]=0001) The blanking time is the time from the offset value of this unit to the window value, please refer to it for details Figure 19-30.

From other units: (HRPWM<t>_EEFLTCR1<2>.EEyFM[3:0]=0010~1101) The blanking time refers to the time from the offset to the window in other Units; it can also be selected as the PWMA<B>_<t>_PRE1 signal of other units (in this case, the event will be ignored as long as the PWM signal is low).

In addition, the blanking signal selected by HRPWM<t>_EEFLTCR1<2>.EEyFM[3:0] = 0010~0011 can be replaced by configuring the HRPWM<t>_GCONR1.EEFM bit. The blanking time of the replaced blanking signal can be set to the time from the reference point to the offset value or the window value; by configuring the HRPWM<t>_GCONR1.EEREF bit, the reference point can be configured as the count value of this unit equals 0 and the count value equals the period value (or hardware clear in sawtooth count mode). For details, refer to the table below. When the window value or offset value is set to coincide with the reference value, the blanking time is the entire counting interval.

Table 19-7 HRPWM<t>_EEFLTCR1<2>.EEyFM Corresponding Blanking Time Setting

EEFLTCR1<2>.EEyFM[3:0]	GCONR1.EEFM=0 GCONR1.EEREF=x	GCONR1.EEFM=1 GCONR1.EEREF=0	GCONR1.EEREF=1 GCONR1.EEFM=1
0b0010	Blanking time refers to the time from unit 1 offset value to window value.	Blanking time refers to the time when the unit count value is equal to the period value (or hardware clear in the sawtooth count Mode) to the offset value.	Blanking time refers to the time from when the count value of this unit is equal to 0 to when the offset value is reached.
0b0011	PWMA<B>_1_PRE1	Blanking time refers to the time when the unit count value is equal to the period value o or hardware clear in the sawtooth count Mode) to the window value.	Blanking time refers to the time when the count value of this unit is equal to 0 to the window value

Windowing Mode

In windowing mode, external events that occur within the windowing time are valid, and external events that occur at other times are ignored. The windowing time for windowing mode is the time it takes for the counter to count from OFFSENT to WINDOW. In windowing mode, when HRPWM<t>_EEFLTCR1<2>.EEyLAT is set to 0 and HRPWM<t>_EEFLTCR1<2>.EEyTMO is set to 0, the events within the windowing time are valid. If HRPWM<t>_EEFLTCR1<2>.EEyTMO is set to 1, if no external event occurs within the specified time, a timeout event will be generated at the end of the windowing time; when HRPWM<t>_EEFLTCR1<2>.EEyLAT is set to 1, the **event postpone mode** of the event will be enabled. If an external event occurs within the windowing time, the external event will not take effect immediately. Once the windowing time is completed, the external event will be stored and generated, such as Figure 19-32. The windowing mode is configured by HRPWM<t>_EEFLTCR1<2>.EEyFM[3:0] bits, ranging from 1110 to 1111.

Note: When HRPWM<t>_EEFLTCR1<2>.EEyLAT is set to 1, the HRPWM<t>_EEFLTCR1<2>.EEyTMO setting invalid.

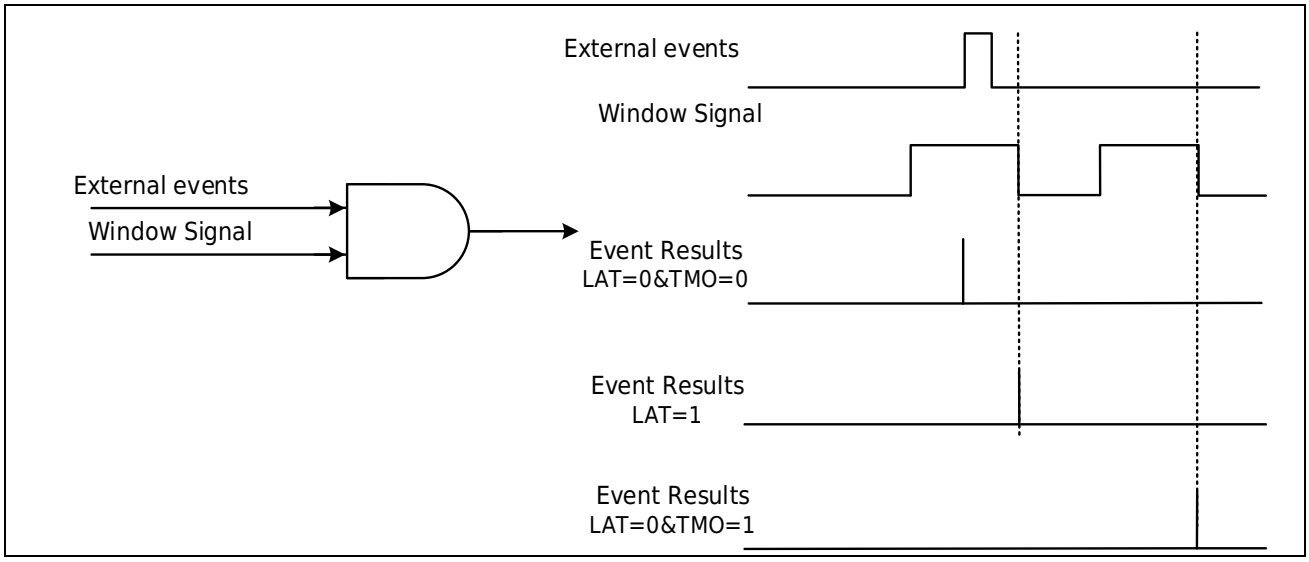


Figure 19-32 Windowing Mode

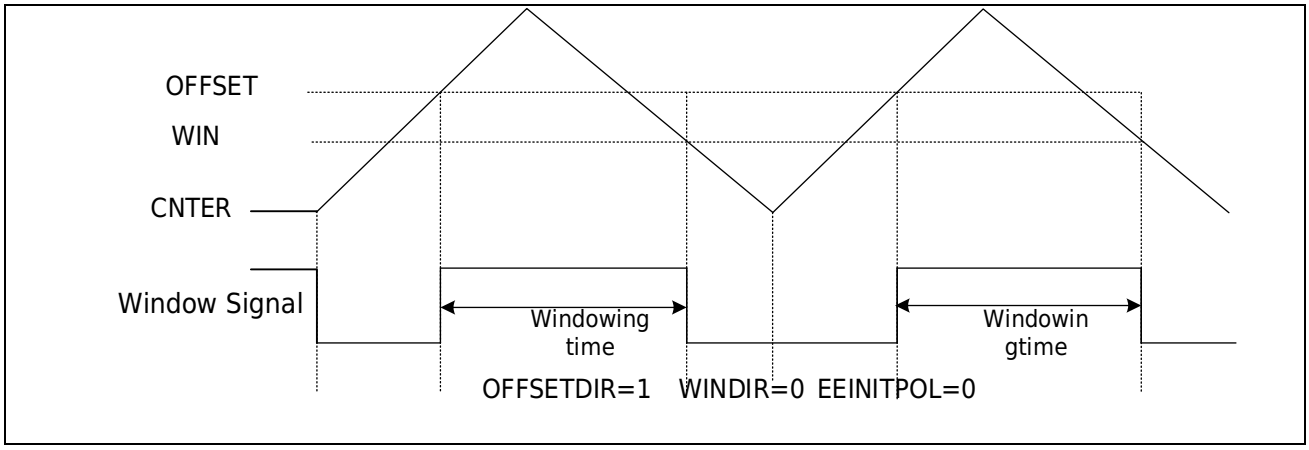


Figure 19-33 Windowing Time

The windowing signal sources include the following two types:

This unit: (HRPWM<t>_EEFLTCR1<2>.EEyFM[3:0]=0b1110) The windowing time is the time from the offset value to the window value of this unit, Please refer to the specific Figure 19-33.

From other units: (HRPWM<t>_EEFLTCR1<2>.EEyFM[3:0]=0b1111) windowing time is the time from the offset value of other units to the window value, See Table 19-8 for details.

Table 19-8 Filter Signal Mapping in Window Mode

Target	Unit 1	Unit 2	Unit 3	Unit 4	Unit 5	Unit 6
TMRWIN	Unit 2	Unit 1	Unit 4	Unit 3	Unit 6	Unit 5

In addition, the windowing signal corresponding to HRPWM<t>_EEFLTCR1<2>.EEyFM[3:0]= 1110~1111 can be replaced by configuring the HRPWM<t>_GCONR1.EEFM bit. The windowing time of the replaced window signal is the time from the reference point to the offset value or the window value. By configuring the HRPWM<t>_GCONR1.EEREF bit, the reference point can be configured as

the count value of this unit equal to 0 and the count value equal to the period value (or hardware clea in sawtooth count Mode). For details, refer to the table below. When the window value or offset value is set to coincide with the reference value, the window time is the entire counting interval.

Table 19-9 HRPWM<t>_EEFLTCR1<2>.EEyFM Corresponding Window Time Setting

EEFLTCR1<2>.EEyFM[3:0]	GCONR1.EEFM=0 GCONR1.EEREF=x	GCONR1.EEFM=1 GCONR1.EEREF=0	GCONR1.EEREF=1 GCONR1.EEFM=1
0b1110	Window time refers to the time from the offset value of this unit to the window value	The window time refers to the time when the unit count value is equal to the period value (or hardware clea in sawtooth count Mode) to the offset value.	The window time refers to the time from when the count value of this unit is equal to 0 to when the offset value is reached.
0b1111	Window signal comes from other units, TMRWIN source	Window time refers to the time when the unit count value is equal to the period value (or hardware clea in sawtooth count Mode) to the window value.	Window time refers to the time from the unit count value equal to 0 to the window value.

19.3.12 Software Synchronization

19.3.12.1 Software Synchronization start

Each unit can achieve synchronous start of the target unit counter (CNTER) by setting the relevant bits of the software synchronous start control register (SSTAR).

19.3.12.2 Software Synchronization stop

Each unit can achieve the synchronous stop of the target unit counter (CNTER) by setting the relevant bits of the software synchronous stop control register (SSTPR).

19.3.12.3 Software Synchronous Clear

Each unit can achieve synchronous clearing of the target unit counter (CNTER) by setting the relevant bits of the software synchronous clear control register (SCLRR).

19.3.12.4 Software Synchronization Update

Each unit can achieve synchronous update of the target unit counter (CNTER) by setting the relevant bits of the software synchronous update control register (SUPDR). Software synchronous update can update the counter value to any value during the counter counting process, which may cause the loss of comparison match events. Please pay attention when using it.

As shown in Figure 19-34, if SSTAR.SSTA1=SSTAR.SSTA2=SSTAR.SSTA3=1 is set, the software synchronization start of units 1~3 can be achieved, and if SSTPR.SSTP1=SSTPR.SSTP2=SSTPR.SSTP3=1 is set, the software synchronization stop of units 1~3 can be achieved.

Software synchronous action related registers (SSTAR, SSTPR, SCLRR, SUPDR) are a group of registers that are independent of the unit and shared between each unit. Each bit of this group of

registers is only valid when writing 1, and writing 0 is invalid. When the SSTAR register is read, the timer status of each unit (counting stopped or counting) is read, and when the SSTPR, SCLRR, or SUPDR is read, zero is read.

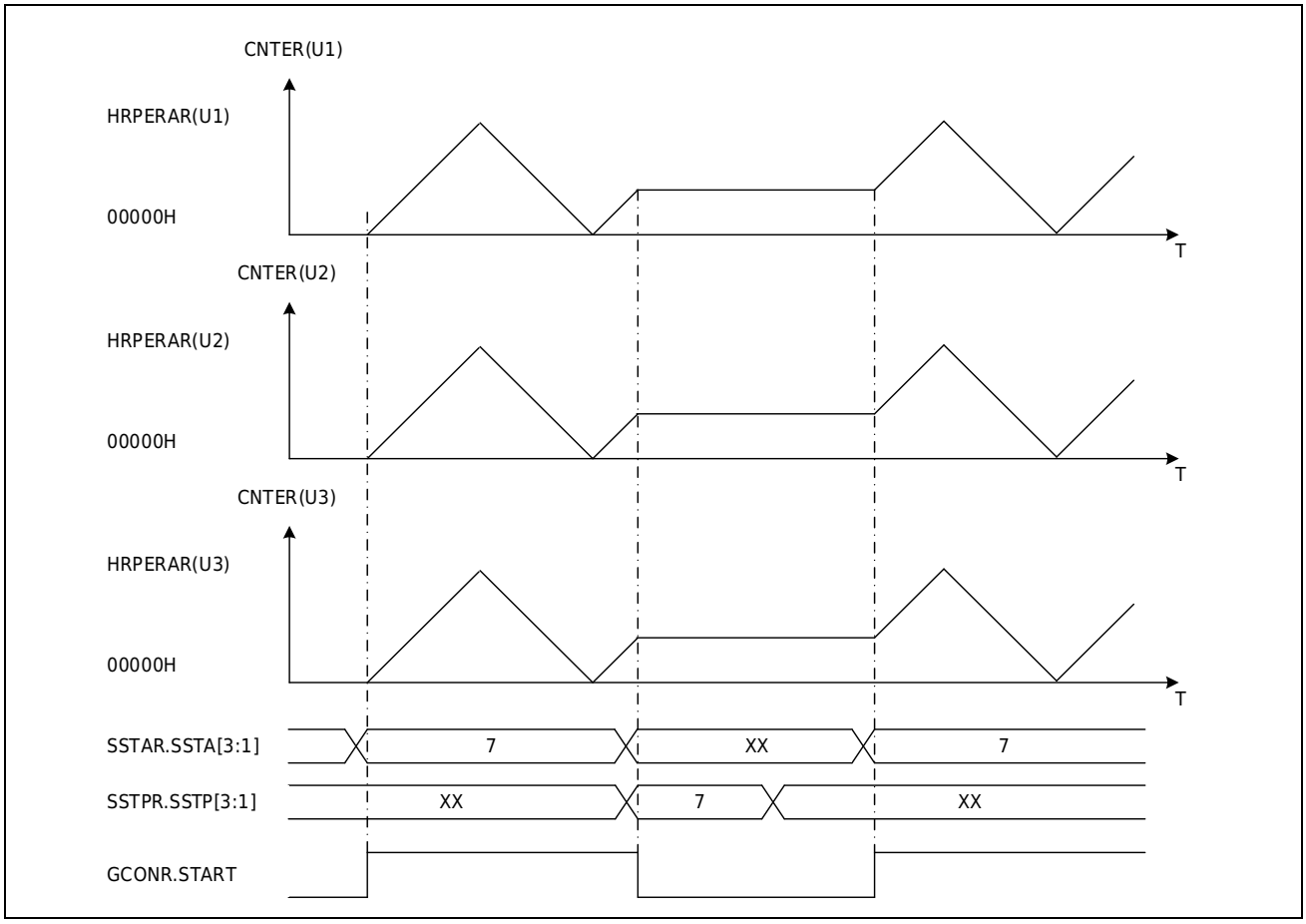


Figure 19-34 Software Synchronization Action

19.3.13 Hardware Synchronization

Each unit has 4 Universal external trigger input ports (HRPWM_TRIGA, HRPWM_TRIGB, HRPWM_TRIGC, HRPWM_TRIGD), 4 internal trigger events, 10 external events and compare match events of each unit, which can realize hardware synchronization action between units.

The source of the internal trigger events can be selected by setting the corresponding number in the **HRPWM Hardware Trigger Event Selection Register** (HRPWM_TRGSEL0~3). For specific event correspondence, please refer to the AOS chapter. When using the internal hardware trigger function, the function clock control register (PWC_FCG0) peripheral circuit trigger function is required to enable.

19.3.13.1 Hardware Sync Start

Each unit can choose to start the timer by hardware, and the units with the same hardware start condition can realize synchronous start when the start condition is valid. The specific hardware

startup conditions are determined by the settings of the **Hardware STARtup Registers** (HSTAR1, HSTAR2).

19.3.13.2 Hardware synchronous clear

Each unit can choose to clear the timer by hardware, and select the units with the same hardware clearing condition to realize synchronous clearing when the clearing condition is valid. The specific hardware clear condition is determined by the setting of the **Hardware CLeaR Register** (HCLRR1, HCLRR2).

19.3.13.3 Hardware Synchronous Capture

Each unit can choose to implement the capture function in hardware. Units that select the same capture input function conditions can achieve synchronous capture when the capture function conditions are valid. The specific hardware capture function conditions are determined by the settings of the **Hardware CAPture Registers** (HRPWM<t>_HCPAR1<2>, HRPWM<t>_HCPBR1<2>).

19.3.14 Synchronous Input Port Digital Filtering

HRPWM_TRIGA~D port input has digital filtering function. The HRPWM_TRIGA~D ports are ports shared between by all timing units. The filtering function of the corresponding port is enabled by setting the relevant enable bit of the **Filter Couter Register** (FCNTR). The filter reference clock when the filtering is effective can also be set through the **Filter Couter Register** (FCNTR).

When the filtered sampling reference clock is sampled to the same level three times on the port, the level is as an effective level. A level less than three times consistent will be filtered out as external interference. Its action example is shown in Figure 19-35.

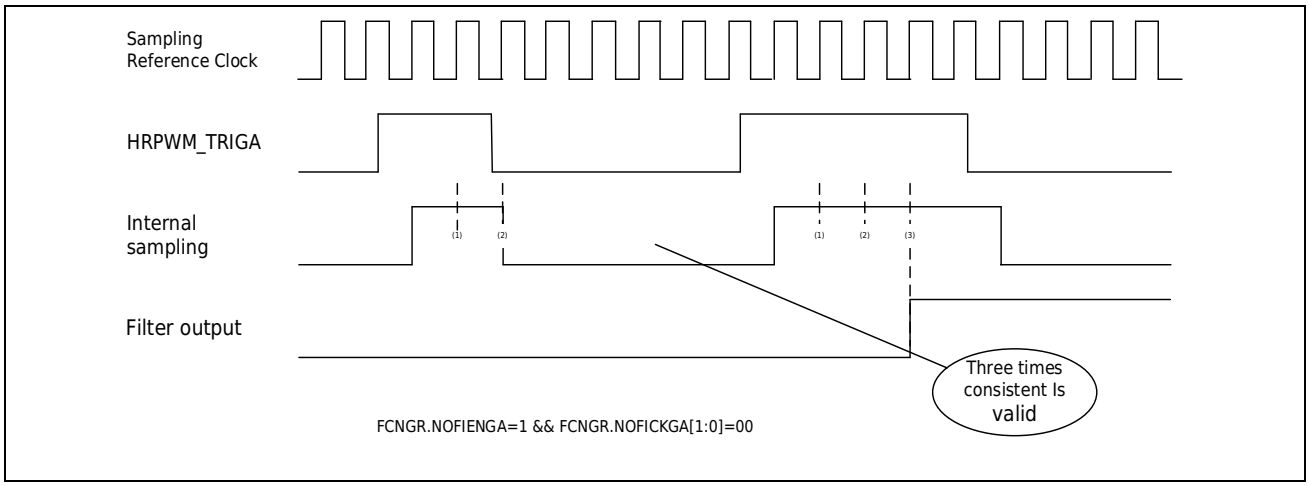


Figure 19-35 HRPWM_TRIGA Filtering Function

19.3.15 Sync Output

The HRPWM can be used as a master to synchronize external MCUs or timers. By configuring the SYNCSR[3:0] bits in the HRPWM_SYNOCR register, you can select the source sent to the synchronization output:

- When the counters of units 1 to 6 are counting and the count value is equal to 0
- When the counters of units 1 to 6 are counting and the count value is equal to SCMBR

When the synchronization source is selected as SCMBR, the SRCDIR bit of the HRPWM_SYNOCR register is configured as the effective direction of the counter when the count value is equal to SCMBR. In the triangle count mode, it can be set to the up counting or the down counting. In the sawtooth wave mode, please set SRCDIR to 1.

The polarity of the synchronous output signal can be configured by the SYNOPLS[1:0] bits of the HRPWM_SYNOCR register:

0b0x: Pulse Generation Disabled
0b10: Generate positive pulse on the synchronization output pin HRPWM_SCOUT. The width of the positive pulse is HRPWM_SYNOCR.SYNCCMP[7:0] PCLK0, followed by an inactive low level with a width of HRPWM_SYNOCR.SYNCCMP[7:0] PCLK0.

0b11: Generate a negative pulse on the synchronization output pin HRPWM_SCOUT. The width of the negative pulse is HRPWM_SYNOCR.SYNCCMP[7:0] PCLK0, followed by an inactive high level with a width of HRPWM_SYNOCR.SYNCCMP[7:0] PCLK0.

During the entire synchronization signal generation period, (HRPWM_SYNOCR.SYNCCMP[7:0] PCLK0 pulse width + HRPWM_SYNOCR.SYNCCMP[7:0] PCLK0 invalid level), any synchronization request will be ignored.

19.3.16 Shadow Function

For unit 1 to unit 6, the HRPWM high-resolution period value, High-resolution general compare , special compare value , high-resolution dead-time value, external event filter offset value, external event filter window value, interrupt control register, port control register, etc. all have a Shadow function, which can realize period changes, duty cycle changes, dead-time changes, etc. in hardware during counting. In addition, the Burst Mode cycle reference value and the Burst Mode compare value also have a Shadow function.

What is different from other units is that unit 1 has a high-resolution phase register and also has a Shadow function.

The list of all registers with Shadow function and their corresponding Shadow transfer enable bits are shown in Table 19-10. When the corresponding Shadow Enable bit of the register with Shadow function valid and a Shadow transfer point occurs, the value of the Shadow register is transferred

to the corresponding active register. It should be noted that when the Shadow function is enabled for registers that support Shadow, the activeregisters still support register writing. Do not write to the activeregisters that have the Shadow Function Enable to avoid unexpected waveforms.

Table 19-10 Supports Shadow Registers

Counting Unit	Active Registers Supporting Shadow	Shadow Register	Shadow Enable Bit
Unit 1	HRPWM1_HRPERAR	HRPWM1_HRPERBR	HRPWM1_BCONR1.BENP
	HRPWM1_HRGCMAR	HRPWM1_HRGCMCR	HRPWM1_BCONR1.BENAE
	HRPWM1_HRGCMBR	HRPWM1_HRGCMDR	HRPWM1_BCONR1.BENBF
	HRPWM1_HRGCMER	HRPWM1_HRGCMGR	HRPWM1_BCONR1.BENAE
	HRPWM1_HRGCMFR	HRPWM1_HRGCMHR	HRPWM1_BCONR1.BENBF
	HRPWM1_HRDTUAR	HRPWM1_HRDTUBR	HRPWM1_DCONR1.DTBENU
	HRPWM1_HRDTDAR	HRPWM1_HRDTDBR	HRPWM1_DCONR1.DTBEND
	HRPWM1_PHSCMP1A	HRPWM1_PHSCMP1B	HRPWM1_PHSCTL.BENPHS
	HRPWM1_PHSCMP2A	HRPWM1_PHSCMP2B	
	HRPWM1_PHSCMP3A	HRPWM1_PHSCMP3B	
	HRPWM1_PHSCMP4A	HRPWM1_PHSCMP4B	
	HRPWM1_PHSCMP5A	HRPWM1_PHSCMP5B	
	HRPWM1_SCMAR	HRPWM1_SCMCR	HRPWM1_BCONR1.BENSPA
	HRPWM1_SCMBR	HRPWM1_SCMDR	HRPWM1_BCONR1.BENSPB
	HRPWM1_ICONR	HRPWM1_BICONR	HRPWM1_BCONR2.BENCTL
	HRPWM1_PCNAR1~3	HRPWM1_BPCNR1~3	
	HRPWM1_PCNBR1~3	HRPWM1_BPCNBR1~3	
	HRPWM1_GCONR1	HRPWM1_BGCONR1	
	HRPWM1_EEFOFFSETAR	HRPWM1_EEFOFFSETBR	HRPWM1_BCONR2.BENEEFOFF
	HRPWM1_EEFWINAR	HRPWM1_EEFWINBR	HRPWM1_BCONR2.BENEEFWIN
Units 2~6 (t=2~6)	HRPWM<t>_HRPERAR	HRPWM<t>_HRPERBR	HRPWM<t>_BCONR1.BENP
	HRPWM<t>_HRGCMAR	HRPWM<t>_HRGCMCR	HRPWM<t>_BCONR1.BENAE
	HRPWM<t>_HRGCMBR	HRPWM<t>_HRGCMDR	HRPWM<t>_BCONR1.BENBF
	HRPWM<t>_HRGCMER	HRPWM<t>_HRGCMGR	HRPWM<t>_BCONR1.BENAE
	HRPWM<t>_HRGCMFR	HRPWM<t>_HRGCMHR	HRPWM<t>_BCONR1.BENBF
	HRPWM<t>_HRDTUAR	HRPWM<t>_HRDTUBR	HRPWM<t>_DCONR1.DTBENU
	HRPWM<t>_HRDTDAR	HRPWM<t>_HRDTDBR	HRPWM<t>_DCONR1.DTBEND
	HRPWM<t>_SCMAR	HRPWM<t>_SCMCR	HRPWM<t>_BCONR1.BENSPA
	HRPWM<t>_SCMBR	HRPWM<t>_SCMDR	HRPWM1<t>_BCONR1. BENSPB
	HRPWM<t>_ICONR	HRPWM<t>_BICONR	HRPWM<t>_BCONR2.BENCTL
	HRPWM<t>_PCNAR1~3	HRPWM<t>_BPCNR1~3	
	HRPWM<t>_PCNBR1~3	HRPWM<t>_BPCNBR1~3	
	HRPWM<t>_BGCONR1	HRPWM<t>_BGCONR1	

Counting Unit	Active Registers Supporting Shadow	Shadow Register	Shadow Enable Bit
	HRPWM<t>_EEFOFFSETAR	HRPWM<t>_EEFOFFSETBR	HRPWM<t>_BCONR2.BENEEFOFF
	HRPWM<t>_EEFWINAR	HRPWM<t>_EEFWINBR	HRPWM<t>_BCONR2.BENEEFWIN
Other	HRPWM_BMPERAR	HRPWM_BMPERBR	HRPWM_BMCR.BENBMP
	HRPWM_BMCMAR	HRPWM_BMCMBR	HRPWM_BMCR.BENBMCMP

19.3.16.1 Shadow Transfer time

Shadow Transfer time in Units

In the triangle count and sawtooth count mode, all registers with Shadow functions can be set to undergo Shadow transfer at different Shadow transfer points when the Shadow function is enabled. The Shadow registers and Shadow transfer points supported by each unit are as shown in the Table 19-11.

Table 19-11 Shadow Transfer Points Supported by Each Unit

Counting Unit	Registers Support Shadow	Shadow time
Unit 1	HRPWM1_HRPERAR	Counter = Zero; Counter = Period; Software trigger (when the counter is stopped) Single Shadow transfer (unit 1 single Shadow transfer enable valid, other Shadow points are invalid)
	HRPWM1_HRGCMAR	
	HRPWM1_HRGCMBR	
	HRPWM1_HRGCMER	
	HRPWM1_HRGCMFR	
	HRPWM1_HRDTUAR	
	HRPWM1_HRDTDAR	
	HRPWM1_PHSCMP1A	
	HRPWM1_PHSCMP2A	
	HRPWM1_PHSCMP3A	
	HRPWM1_PHSCMP4A	
	HRPWM1_PHSCMP5A	
	HRPWM1_SCMAR	
	HRPWM1_SCMBR	
	HRPWM1_ICONR	
	HRPWM1_PCNAR1~3	
	HRPWM1_PCNBR1~3	
	HRPWM1_GCONR1	
	HRPWM1_EEFOFFSETAR	
	HRPWM1_EEFWINAR	
Units 2~6 (t=2~6)	HRPWM<t>_HRPERAR	Counter = Zero; Counter = Period; Software trigger (when the counter is stopped) Unit 1 Single Shadow Transfer; counter = period in The sawtooth or counter
	HRPWM<t>_HRGCMAR	
	HRPWM<t>_HRGCMBR	
	HRPWM<t>_HRGCMER	

Counting Unit	Registers Support Shadow	Shadow time
	HRPWM<t>_HRGCMFR	= zero in triangle counter wave after the Single Shadow Transfer Time of unit 1 (valid only when the single Shadow transfer of unit 1 is enabled)
	HRPWM<t>_HRDTUAR	
	HRPWM<t>_HRDTDAR	
	HRPWM<t>_SCMAR	
	HRPWM<t>_SCMBR	
	HRPWM<t>_ICONR	
	HRPWM<t>_PCNAR1~3	
	HRPWM<t>_PCNBR1~3	
	HRPWM<t>_BGCONR1	
	HRPWM<t>_EEFOFFSETAR	
	HRPWM<t>_EEFWINAR	
Other	HRPWM_BMPERAR	Overflow point of BM-counter
	HRPWM_BMCMAR	

Note:

In the above table, in the triangle count mode, the counter = zero is the moment when the down-count value is equal to 64 or the moment when the hardware clear, and the counter = period is the moment when the up-count value is equal to the period value - 64; in the sawtooth count mode, the counter = zero is the moment when the count value is equal to 0, and counter = period is the moment when the count value is equal to the period value or the hardware clear.

When the counter counts, unit 1 can configure the **Shadow transfer time** as *counter = zero* or *counter = period*, and enable different **Shadow transfer time** by configuring HRPWM1_BCONR1, HRPWM1_BCONR2, HRPWM1_DCONR, and HRPWM1_PHSCTL. When counting stops, writing 1 to HRPWM_SBUFR.SBU1TRG can trigger a software Shadow.

In addition to the above Shadow time, unit 1 also has a **Single Shadow Transfer Time** during the counter counting process. The **Single Shadow Transfer Time** of unit 1 can be enabled by setting the OSTENU1 bit of HRPWM_GBCONR to 1.

A Single Shadow Transfer Time refers to: in triangle count mode, after writing 1 to the OSTBTRU1 bit, the first counter = zero; in sawtooth wave mode, after writing 1 to the OSTBTRU1 bit, the first counter = period. When the **Single Shadow Transfer Time** is enabled (HRPWM_GBCONR. OSTENU1 = 1), other **Shadow Transfer Time** set in HRPWM1_BCONR1 and HRPWM1_BCONR2 are invalid.

Units 2 to 6 can configure t **Shadow transfer time** as *counter = zero* or *counter = period*, and enable different **Shadow transfer time** by configuring HRPWM<t>_BCONR1, HRPWM<t>_BCONR2, HRPWM<t>_DCONR, and HRPWM<t>_PHSCTL. When counting stops, writing 1 to SBUFR.SBU<t>TRG. In addition to the above Shadow points, units 2 to 6 can also Shadow at the **unit 1 Single Shadow Transfer Time** during the counter counting process, which can be

enabled by configuring the BTRU0xx bits in HRPWM<t>_BCONR2 and HRPWM<t>_PHSCTL. By configuring the BTRU0Pxx bits in HRPWM<t>_BCONR2, HRPWM<t>_DCONR, and HRPWM<t>_PHSCTL. Shadow transfer time can also be configured as counter = period in The sawtooth counter Mode or counter = zero in triangle counter Mode after the **Single Shadow Transfer Time of unit 1** (v HRPWM_GBCONR. OSTENU1=1).

Note:

When the period value update time is set at the counter = period in the triangle count mode, the following abnormal situation will occur. When the period value changes from small to large, a lost period event will occur. When the period changes from large to small, the counter will immediately update to the new period value and start counting down. Avoid update period value at p counter = period in triangle count mode. Figure 19-36 illustration of the unusual situation of shadow the period value at the counter = period in the triangle count mode.

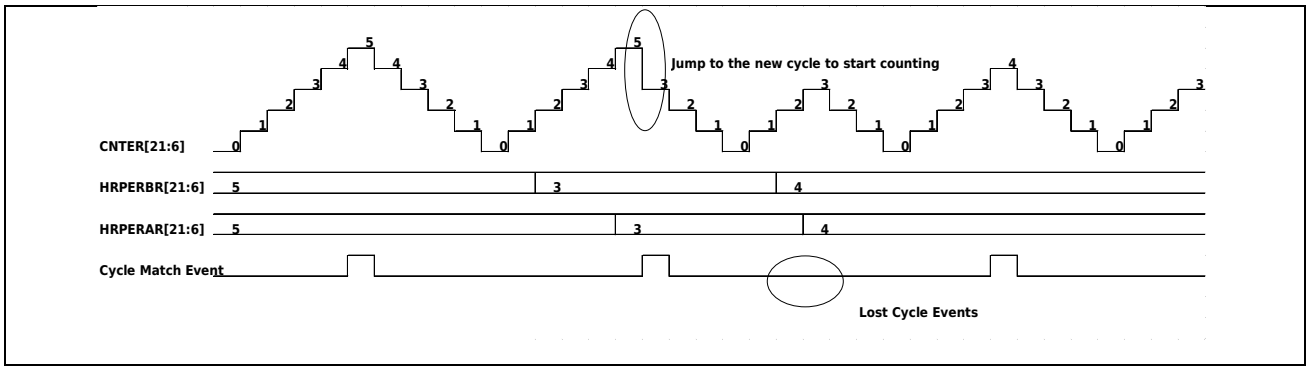


Figure 19-36 Abnormal Situation of Shadow Period Value of Triangle Wave Period Shadow Transmission time

Note:

1. *In the triangle count mode, when the counter value is set to 0 in the stop state, if the **Shadow Transfer Time** is set to the counter = zero, no Shadow will occur when the first count value is equal to 0 after the count is started. Please pay attention when using it. If you need to Shadow a new reference value before starting the counter, you can write 1 to HRPWM_SBUFR.SBU<t>TRG when the counter is stopped to trigger a software Shadow.*
2. *In triangular count mode, when **Shadow Transfer Time of** phase value is set to be counter = period, the phase value in the down-counting interval is the updated phase value, and the phase value in the up-counting interval before the Shadow is the phase value before the update. In triangular count mode, the phase value is avoided from being Shadowd at the counter = period.*

Figure 19-37 *Shadow Transfer Time is counter = period*.

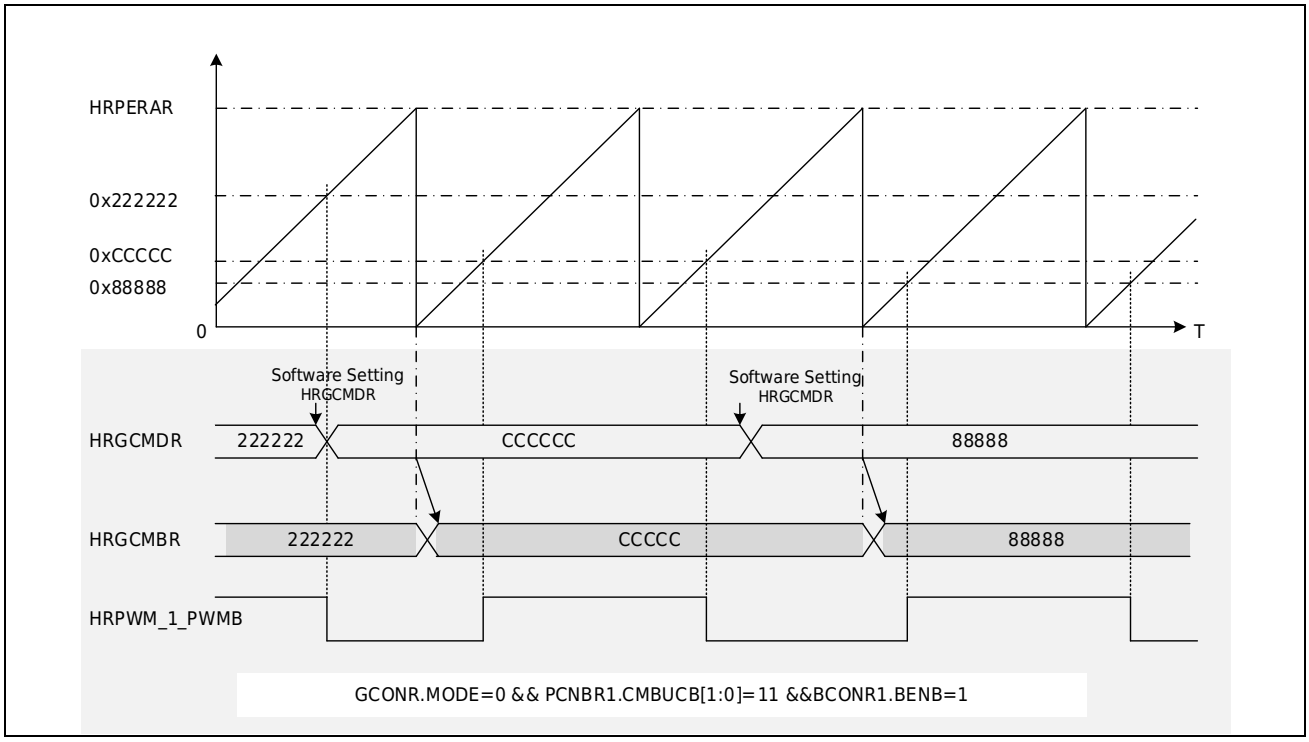


Figure 19-37 Period Shadow Transfer Time

Figure 19-38 which is a schematic diagram showing the shadow action of the triangle Mode at the zero **Shadow Transfer Time**, and Figure 19-39, which is a schematic diagram shadow the shadow action occurring at both the zero and period **Shadow Transfer Time** of the triangle wave.

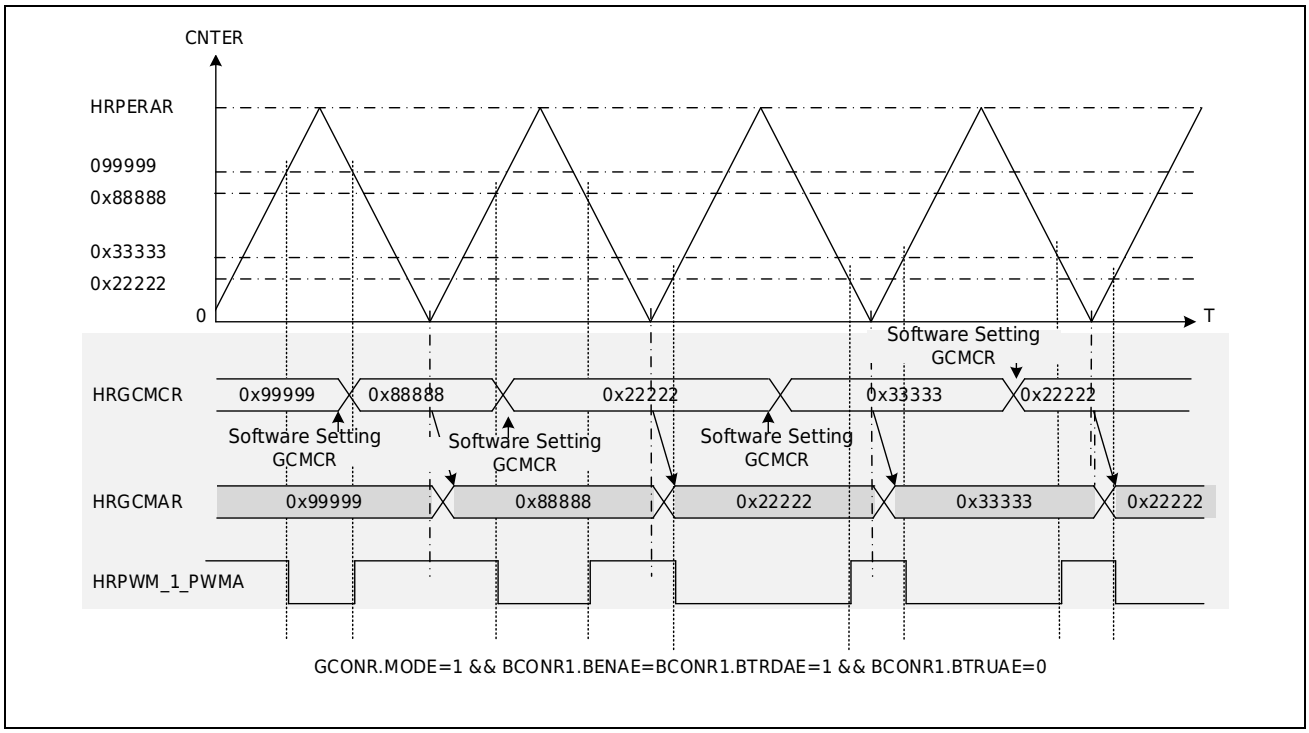


Figure 19-38 Shadow Action 1 in Triangle count mode

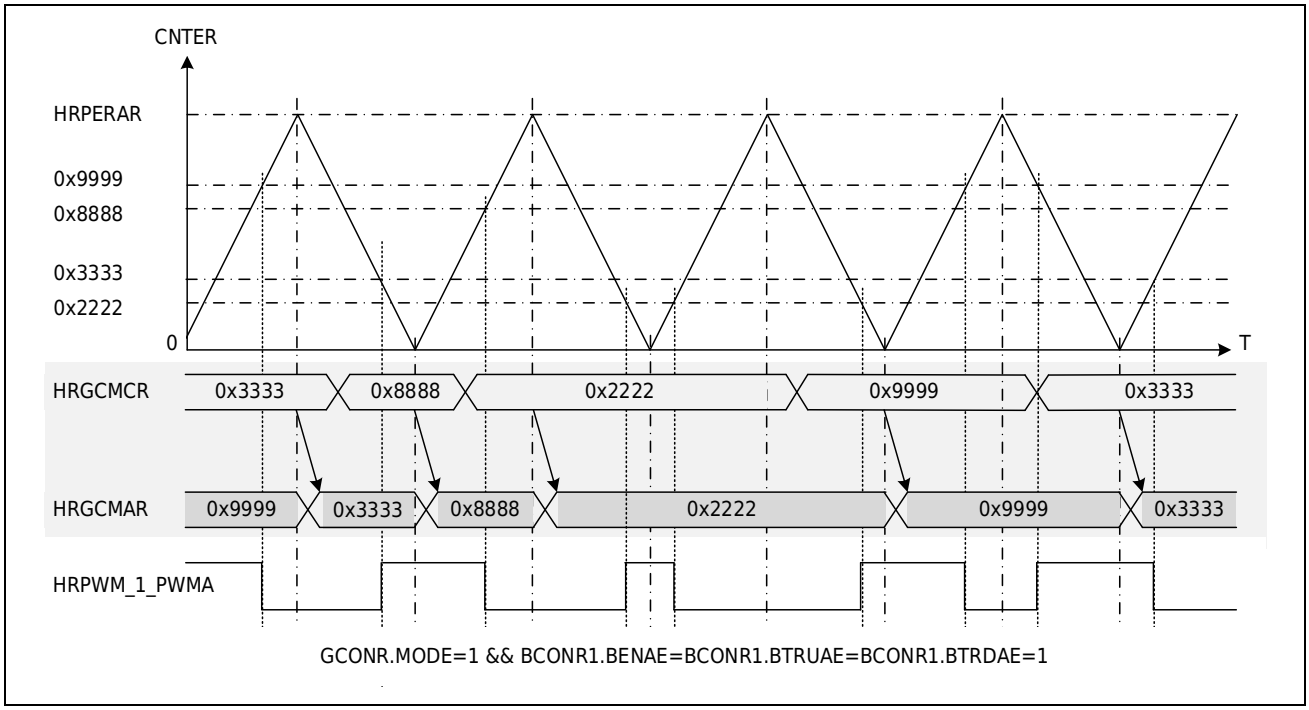


Figure 19-39 Shadow Action 2 in Triangle count mode

Burst Mode Shadow Time

The Shadow transfer time of the Burst Mode Period value and the Burst Mode compare value occurs at the overflow point of the BM-counter. Figure 19-40 shows the Burst Mode Shadow action.

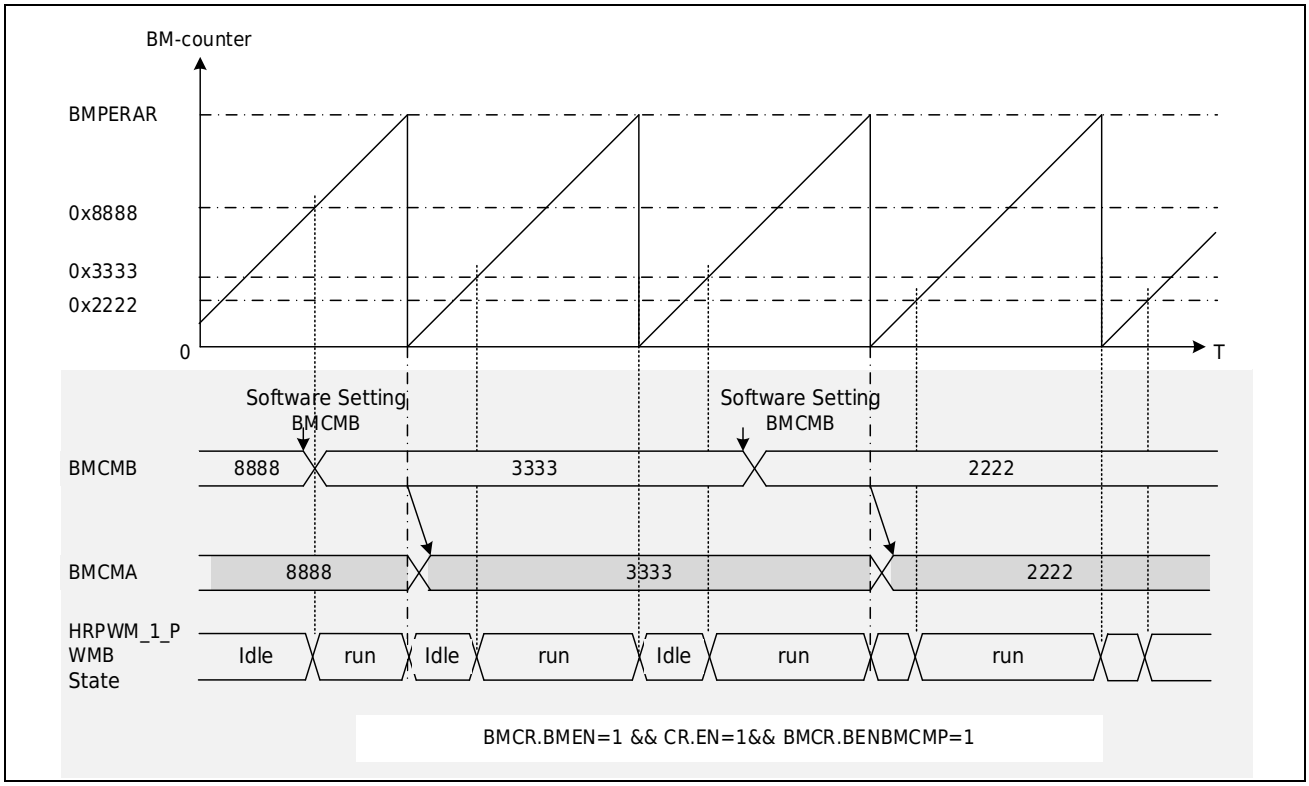


Figure 19-40 Burst Mode Shadow Action

Note:

If you need to ensure that the period value and the compare value are Shadowd at the same time, please write the period value first and then the compare value; if you only modify the period value, you need to write the compare value again after writing the cycle value; if you only modify the compare value, just write the compare value.

19.3.16.2 Global Shadow Shielding Function Per Unit

By setting the corresponding unit control bit BTRU<t>EN of HRPWM_GBCONR to 0, the Shadow enable bits of each unit in the Table 19-10 can be masked to 0 at the same time.When the corresponding unit control bit is 1, Table 19-10 the Shadow Enable of each unit restores the state set by its own register.

19.3.16.3 Global Shadow Completion Flag Per Unit

The global buffer status flag register bit HRPWM_GBSFLR.U<t>BTRENF can be set high when Active Register is updated.The set time of the unit HRPWM_GBSFLR. U<t>BTRENF is as shown in the following table. Each unit HRPWM_GBSFLR. U<t>BTRENF occurs at the following set time and is set when the corresponding HRPWM_GBCONR.BTRUxEN is 1.

Table 19-12 Global ShadowStatus Flag Setting Conditions for Each Unit

unit cell	Counter Status	HRPWM_GBSFLR. U<t>BTRENF set time	Set conditional enable bit
Unit 1	Stop	Software Shadow trigger, write 1 to HRPWM_GBCONR.BTRU1SFT	-
	Counting	Period **Shadow Transfer Time**	HRPWM1_GCONR1. BFSEL[0]
		Zero **Shadow Transfer Time**	HRPWM1_GCONR1. BFSEL[1]
		Single **Shadow Transfer Time**	HRPWM1_GCONR1. BFSEL[2]
Unit<t> (t=2-6)	Stop	Software trigger, write 1 to HRPWM_GBCONR.BTRU<t>SFT	-
	Counting	Period **Shadow Transfer Time**	HRPWM<t>_GCONR1. BFSEL[0]
		Zero **Shadow Transfer Time**	HRPWM<t>_GCONR1. BFSEL[1]
		Unit 1 Single Shadow Transfer Time	HRPWM<t>_GCONR1. BFSEL[2]
		This unit Sawtooth Mode period **Shadow Transfer Time** Triangle Mode Zero **Shadow Transfer Time after** Unit 1 Single Shadow Transfer Time	HRPWM<t>_GCONR1. BFSEL[3]

Please follow the following process to clear the global Shadow completion status flag:

1) After reading the global Shadow completion status flag bit 1, set the HRPWM_GBCONR. BTRU<t>EN of the corresponding unit to 0;
2) Writing 0 to HRPWM_GBSFLR. U<t>BTRENF clears the global buffer completion status flag.

19.3.17 PWM Output

The resolution of HRPWM output PWM waveform is 1/ (fPCLK0*64), and it mainly supports the following PWM output waveforms.

19.3.17.1 Single-edge Aligned Independent PWM Output

In sawtooth counting mode (HRPWM<t>_GCONR.MODE=0), the two ports HRPWM_<t>_PWMA and HRPWM_<t>_PWMB of each unit can output PWM waves independently. If the same level change is set when the count value is equal to the period value, single-side aligned independent high-resolution PWM output can be achieved. As shown in Figure 19-41 below.

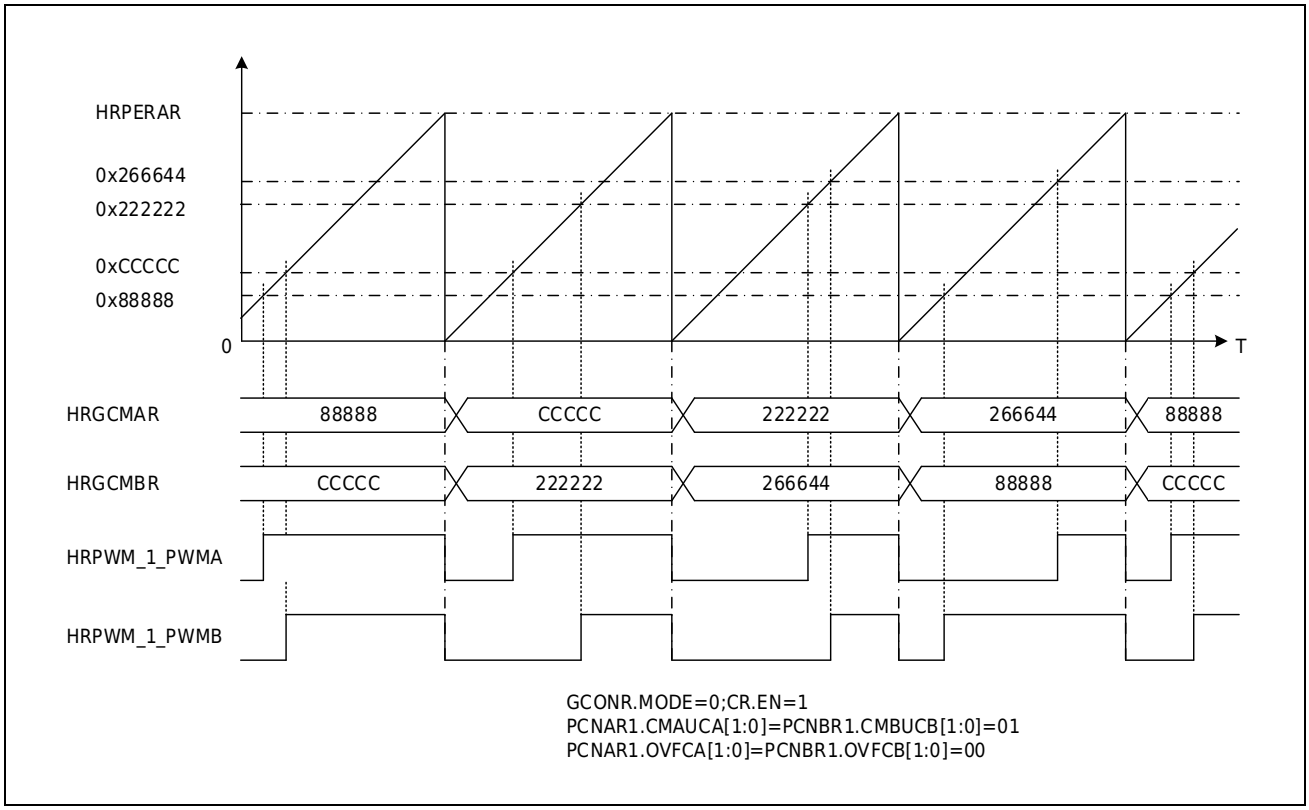


Figure 19-41 Single Edge Aligned Independent High Resolution PWM

19.3.17.2 Symmetrical Independent PWM Output on Both Sides

In the triangle counting mode (HRPWM<t>_GCONR.MODE=1), the two ports HRPWM_<t>_PWMA and HRPWM_<t>_PWMB of each unit can output PWM waves independently. If the set level changes during count comparison matching, and the set level does not change when the count value is equal to 0 or the count value is equal to the period value, bilateral symmetrical independent PWM output can be achieved. As shown in the Figure 19-42, the HRPWM_<t>_PWMA port and the HRPWM_<t>_PWMB port implement bilateral symmetrical independent output PWM.

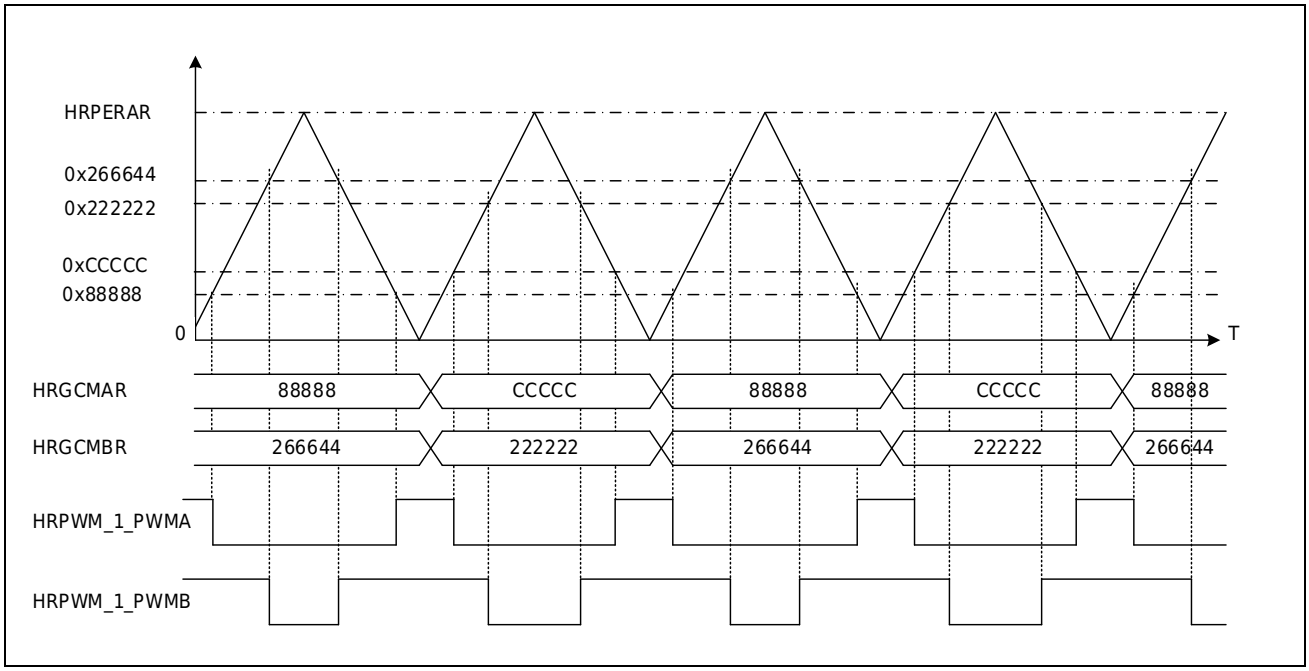


Figure 19-42 Bilateral Symmetrical Independent PWM

19.3.17.3 Complementary PWM Output

Software Setting to Realize Complementary PWM Output

When the dead zone is not enabled (HRPWM<t>_DCONR.DTCEN=0), set the port level state when counting starts, compare matches, count value equals period value, and count value equals 0, so that a pair of complementary PWM waveforms can be output on the HRPWM_<t>_PWMA and HRPWM_<t>_PWMB ports.

Figure 19-43 is an example of the output of bilateral symmetrical complementary PWM waves.

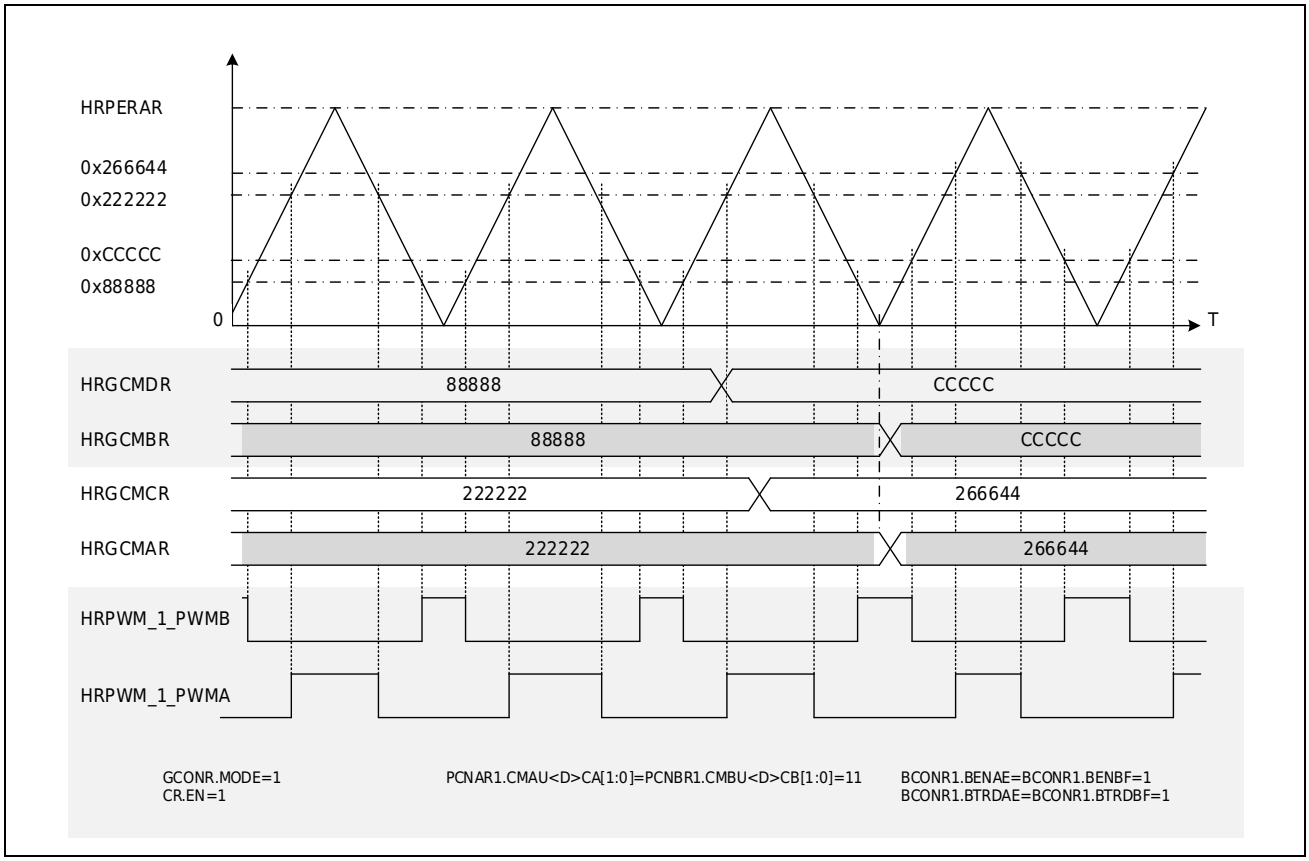


Figure 19-43 Software Realizes Complementary PWM Wave Output

Realization of Complementary PWM Output by Hardware Setting

When the dead zone is enabled (HRPWM<t>_DCONR.DTCEN=1), set the port level state when counting starts, compare matches, count value equals period value and count value equals 0, generate channel A internal output signal (HRPWM_<t>_PWMA_ORG), and offset the set value of the high-precision dead time reference value register (HRPWM<t>_DTU<D>AR) through timing, and output a pair of complementary PWM waveforms on the HRPWM_<t>_PWMA and HRPWM_<t>_PWMB ports in hardware. For details, refer to [Dead Zone Module] for details.

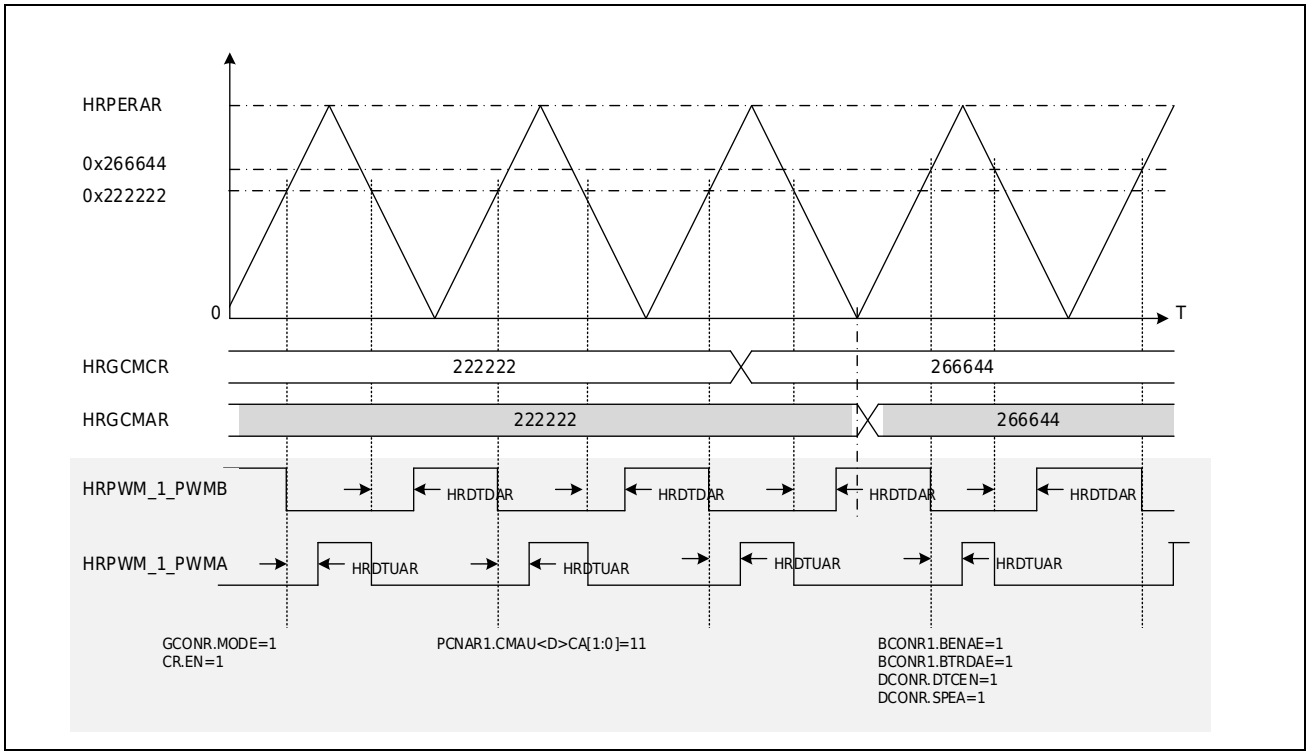


Figure 19-44 Hardware Realizes Complementary PWM Wave Output

19.3.17.4 Double-sided Asymmetric PWM Output

In the triangle wave counting mode (HRPWM<t>_GCONR.MODE=1), the two ports of each unit, HRPWM_<t>_PWMA and HRPWM_<t>_PWMB, can independently output high-resolution PWM waves, and each port can make corresponding level output changes according to the reference of the High-resolution general compare Register (HRPWM<t>_HRGCMAR, HRPWM<t>_HRGCMBR). If the port level change of HRPWM_<t>_PWMA is controlled by the comparison results of the reference values of HRPWM<t>_HRGCMAR and HRPWM<t>_HRGCMBR during the up-counting and down-counting periods of the counter, then bilateral asymmetric high-resolution PWM output can be achieved on the HRPWM_<t>_PWMA port. As shown in the Figure 19-45, it is the double-sided asymmetric high-resolution PWM output of the HRPWM_<t>_PWMA port.

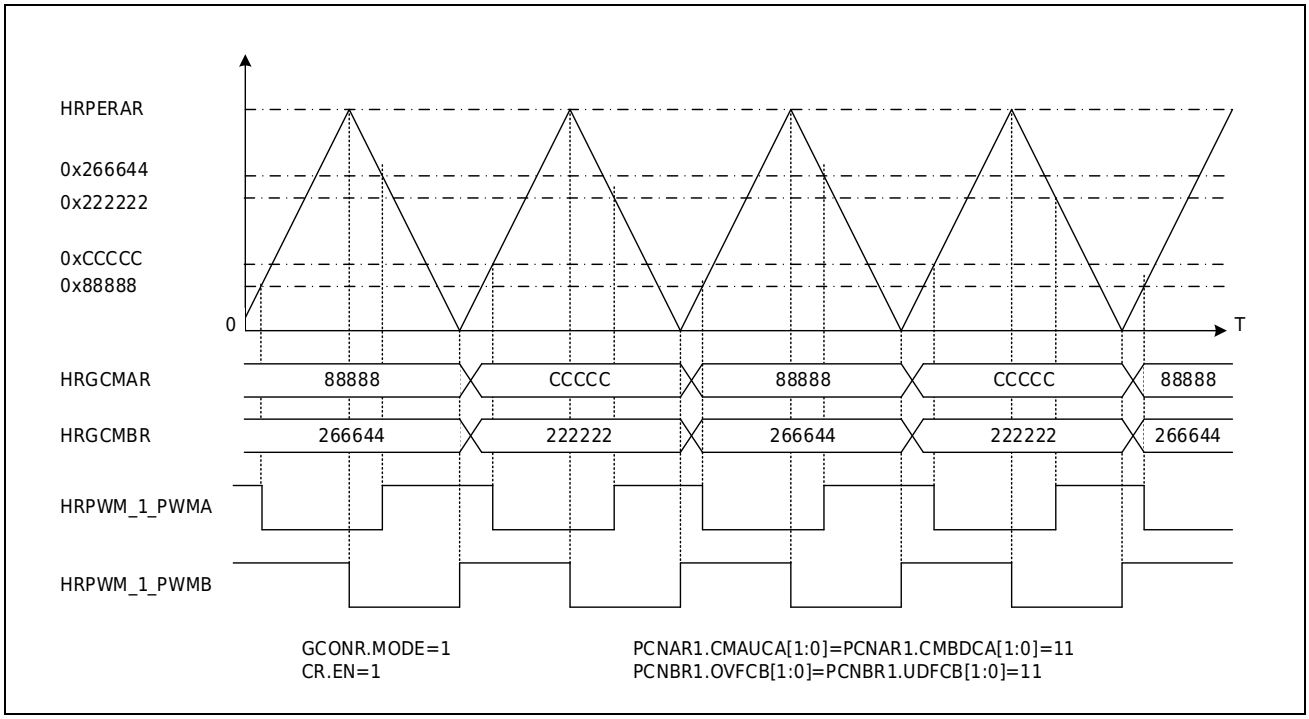


Figure 19-45 Double-sided Asymmetric PWM Output

19.3.17.5 Multi-phase PWM Output between Units

The HRPWM_<t>_PWMA and HRPWM_<t>_PWMB ports of each unit can output 2-phase PWM waves. By combining multiple units and combining software and hardware synchronization actions, multi-phase high-resolution PWM wave output can be achieved. As shown in the Figure 19-46, unit 1, unit 2, and unit 3 are combined to output 6-phase single-side aligned independent high-resolution PWM.

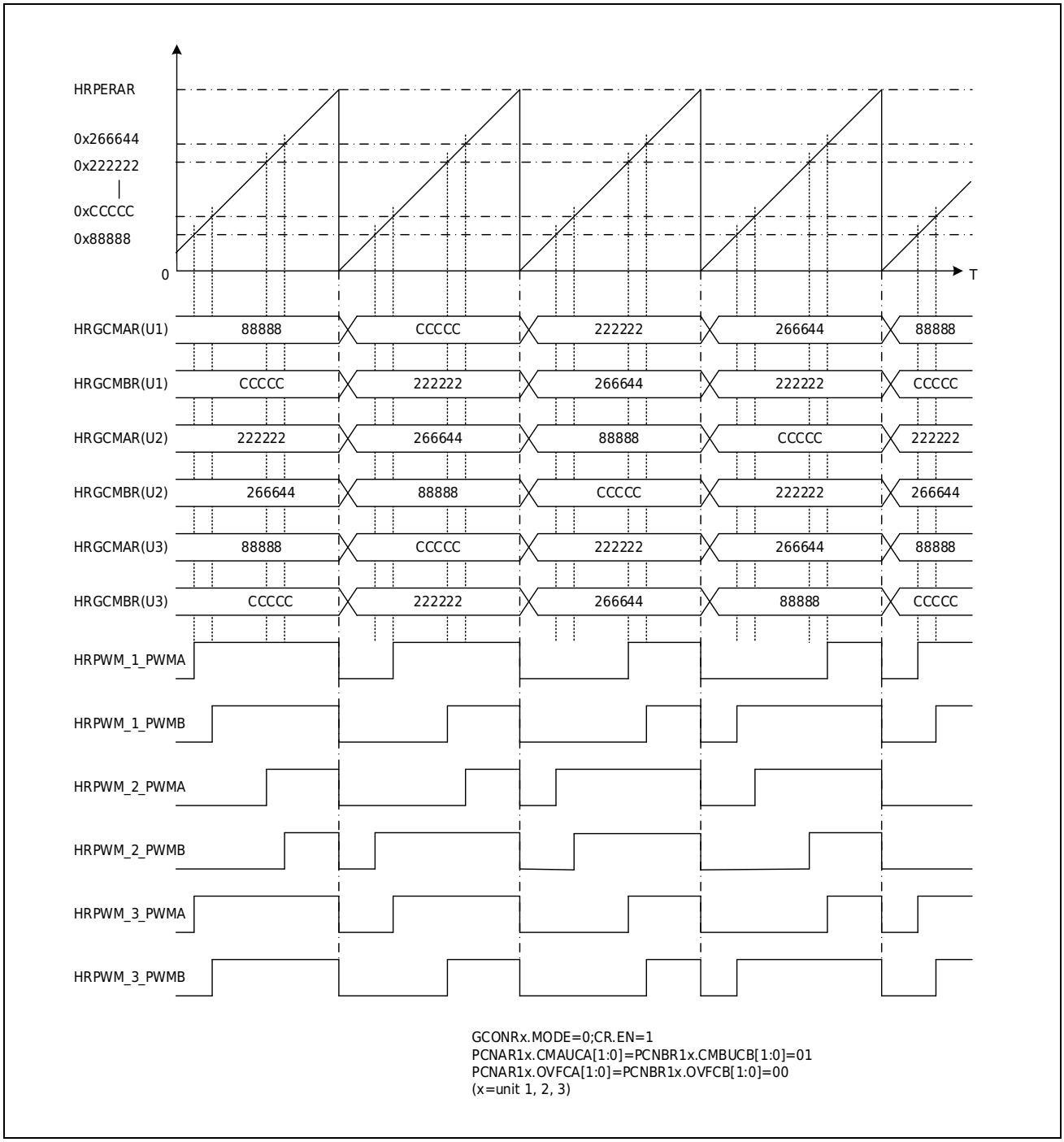


Figure 19-46 6-Phase Single-side Aligned Independent PWM

19.3.17.6 Multi-Phase Interleaved PWM Output Between Units

Multi-Phase Interleaved PWM Output

The HRPWM_<t>_PWMA and HRPWM_<t>_PWMB ports of each unit can output PWM. By using the phase reference value register of unit 1 and the phase function of units 2 to 6, multi-phase interleaved PWM output of up to 6 units can be achieved. As shown in the Figure 19-47, unit 1, unit 2, and unit 3 are combined to output three-phase interleaved complementary high-resolution PWM.

In this example, unit 1 acts as the master unit and generates phase comparison matching events at the corresponding phase comparison points. In this example, the counting mode is the triangle count mode, the period reference value is 4500, and three phases need to be implemented. The high-resolution phase comparison reference values of unit 2 and unit 3 are calculated using the following formula.

- (Triangle wave pattern)
-
- N = number of phases
- M=PWM module number

After calculation, the phase matching value of unit 2 is set to 2936, and the phase matching value of unit 3 is set to 5936.

After setting the registers of each unit according to the diagram, start unit 1. When the count value of unit 1 reaches the corresponding phase matching value, the corresponding slave unit starts counting.

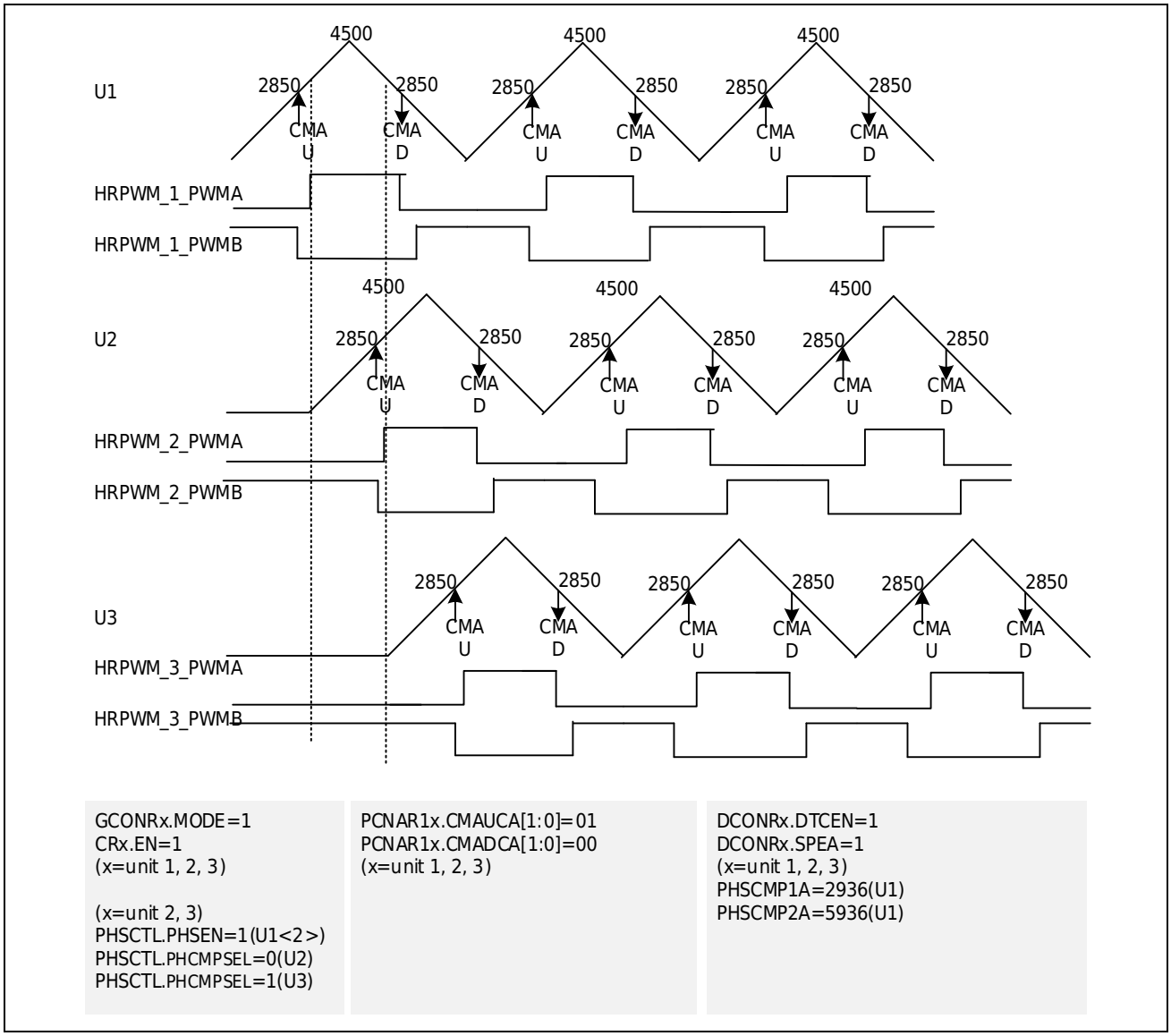


Figure 19-47 3-phase Interleaved Complementary High-resolution PWM

Multi-phase Interleaved PWM Dynamic Adjustment

During multi-phase PWM output, the period, duty cycle, dead zone, etc. can be adjusted by hardware in combination with the Shadow function of each reference value register.

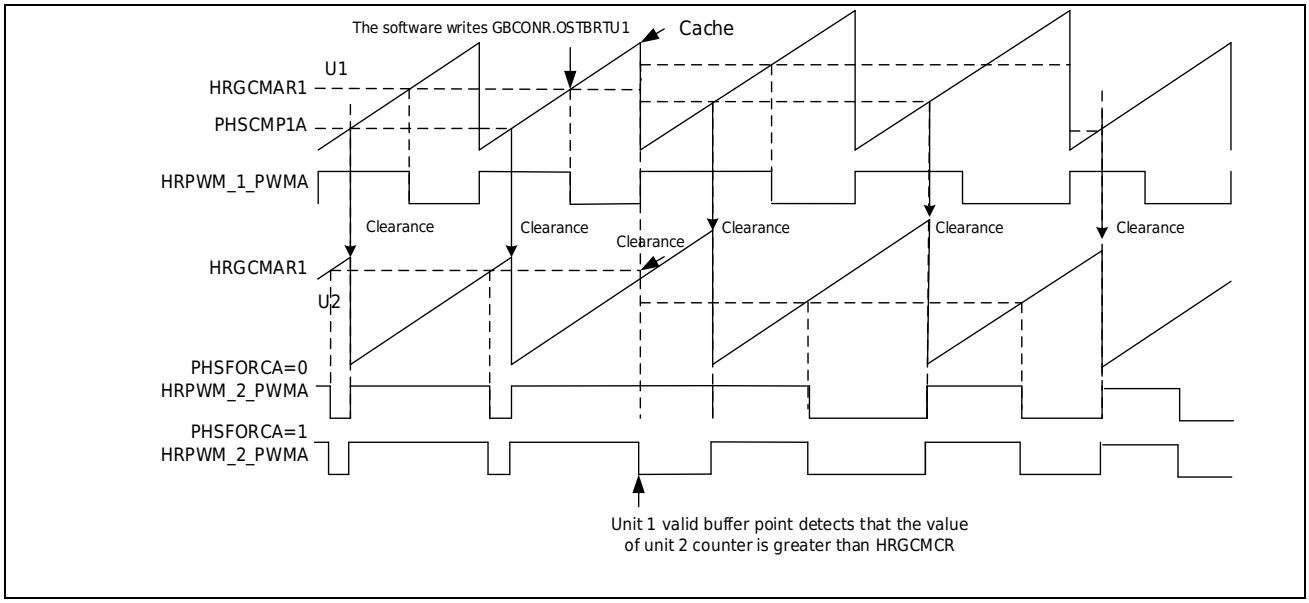


Figure 19-48 Sawtooth Wave Mode Multi-phase Interleaved PWM Variable Frequency Output

Figure 19-48 is an example diagram of multi-phase interleaved PWM frequency conversion output in sawtooth mode. The specific configuration of the above figure is as follows:

1. Master unit(Unit1):

   1) PWM output configuration: set high when the count value is equal to 0, and set low when the count value is equal to HRGCMAR;
   2) Shadow configuration: configure the Shadow control register HRPWM1_BCONR1.BENP<AE><BF>=1 to enable the Shadow function of the high-resolution period value register, high-resolution general compare register, etc.; configure the phase control register HRPWM1_PHSCTL. BENPHS=1 to enable the Shadow function of the unit 1 high-resolution phase comparison reference value register;
   3) Configure the global Shadow control register HRPWM_GBCONR. OSTENU1 enables unit 1 Single Shadow Transfer Time;
   4) Others: Configure comparison value, period value, and phase matching value.

2. From the unit:

   1) PWM output configuration: set high when the count value is equal to 0, and set low when the count value is equal to HRGCMAR;
   2) Phase function configuration: configure the phase control register HRPWM<t>_PHSCTL.PHSEN=1 to enable the phase function, HRPWM<t>_PHSCTL.PHCMPSEL=1 to select PHSCMP1A as the phase matching point, HRPWM<t>_PHSCTL.PHSFORCA=1 to enable the output A to be forced to low function when the current count value is detected to be greater than or equal to the value set by the comparison value register HRPWM<t>_HRGCMCR at the unit 1 single buffer transfer point;
   3) Shadow configuration: configure the Shadow control register HRPWM<t>_BCONR2.BENAE=1

to enable the Shadow function of the high-resolution period value register and the high-resolution general compare register respectively; configure the Shadow control register HRPWM<t>_BCONR2. BTRU0PAE<BF> =1 to set the Shadow point of the above registers as the Single Shadow Transfer Time of unit 1;

4) Others: Configure comparison value and phase matching value. The comparison value is set to the maximum period value.

When making dynamic adjustments, HRPWM_GBSFLR.U1BTRENF is read to confirm that the reference values of each unit have been Shadowd. After clearing the flag bit, the new period value and comparison value are written to the Shadow register, and OSTBTRU1 is written as 1 to trigger the single Shadow of unit 1.

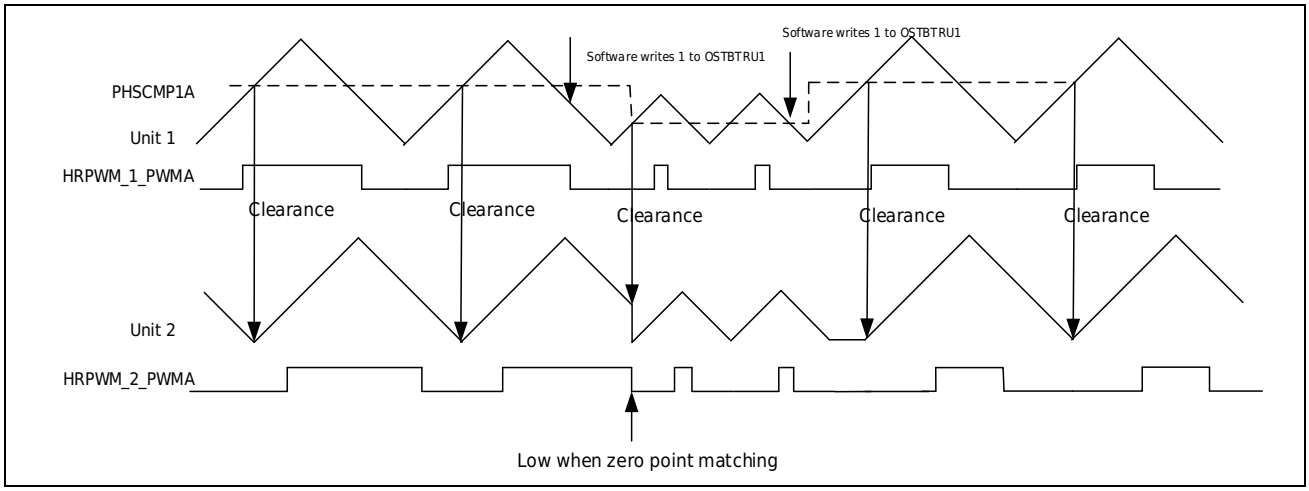


Figure 19-49 Triangle count mode Multi-phase Interleaved PWM Variable Frequency Output

Figure 19-49 is an example diagram of multi-phase interleaved PWM frequency conversion output in triangular wave mode. The specific configuration of the above figure is as follows:

1. Mater unit(Unit1):

1) PWM output configuration: when the count value is equal to HRGCMAR, it is set high; when the count value is equal to HRGCMAR, it is set low; when the count value is equal to 0, it is set low; when the counter stops, it is set low;
2) Shadow configuration: configure the Shadow control register HRPWM1_BCONR1.BENP<AE>=1 to enable the Shadow function of the high-resolution period register, high-resolution general compare register, etc.; configure the phase control register HRPWM1_PHSCTL. BENPHS=1 to enable the Shadow function of the unit 1 high-resolution phase Comparison Reference Value Register;
3) Configure the global Shadow control register HRPWM_GBCONR. OSTENU1 enables unit 1 Single Shadow Transfer Time;
4) Others: Configure comparison value, period value, and phase matching value.

2. Slave Unit

1) PWM output configuration: when the count value is equal to HRGCMAR, it is set high; when the count value is equal to HRGCMAR, it is set low; when the count value is equal to 0, it is set low; when the counter stops, it is set low;
2) Phase function configuration: configure the phase control register HRPWM<t>_PHSCTL.PHSEN = 1 to enable the phase function, HRPWM<t>_PHSCTL. PHCMPSEL = 1 to select PHSCMP1A as the phase matching point;
3) Shadow configuration: configure the Shadow control register HRPWM<t>_BCONR2.BENP<AE><BF>=1 to enable the Shadow function of the high-resolution period register and the high-resolution general compare register respectively; configure the Shadow control register HRPWM<t>_BCONR2. BTRU0PP<AE> =1 to set the Shadow point of the above registers to the sawtooth wave period **Shadow Transfer Time** or the triangle wave zero point **Shadow Transfer Time** after the unit 1 single Shadow point;
4) Others: Configure the comparison value and period value, and configure HRPWM<t>_GCONR.OVSTP to 1 to enable the single count function.

When making dynamic adjustments, read HRPWM_GBSFLR.U<t>BTRENF to confirm that the reference values of each unit have been Shadowd. After clearing the flag bit, write the new period value and comparison value to the Shadow register, and write 1 to HRPWM_GBCONR.OSTBTRU1 to trigger a single Shadow of unit 1.

19.3.18 Periodic Interval Response

The 2 special-compare registers of HRPWM (HRPWM<t>_SCMAR, HRPWM<t>_SCMBR) can respectively output the special-compare match interrupt A signal and the special-compare match interrupt B signal to INTC; at the same time, they can respectively output the special-compare match event A signal and the special-compare match event B signal, which are used to associate actions with other modules, mostly used to start ADC, etc.

The interrupt and event request signal can generate an effective request signal every several periods, that is, realize periodic interval response. This function is enabled by setting the HRPWM<t>_VPERR.PCNTE[1:0] bits and the HRPWM<t>_VPERR.SPPERIA/B bits in the Active Period Register (HRPWM<t>_VPERR). Set the HRPWM<t>_VPERR.PCNTS[4:0] bits to specify how many cycles the request signal valid. In other cycles, even if the count value is equal to the value of the special-compare Register HRPWM<t>_SCMAR or HRPWM<t>_SCMBR, no valid request signal will be output.

When this function is enabled, the period/zero match interrupt and period/zero match event in each waveform mode are also output only in the valid period of the dedicated compare match interrupt and event output (the period when HRPWM<t>_STFLR.VPERNUM=0 in the figure below). Figure 18-24 shown is an example of the operation of the periodic interval valid request signal.

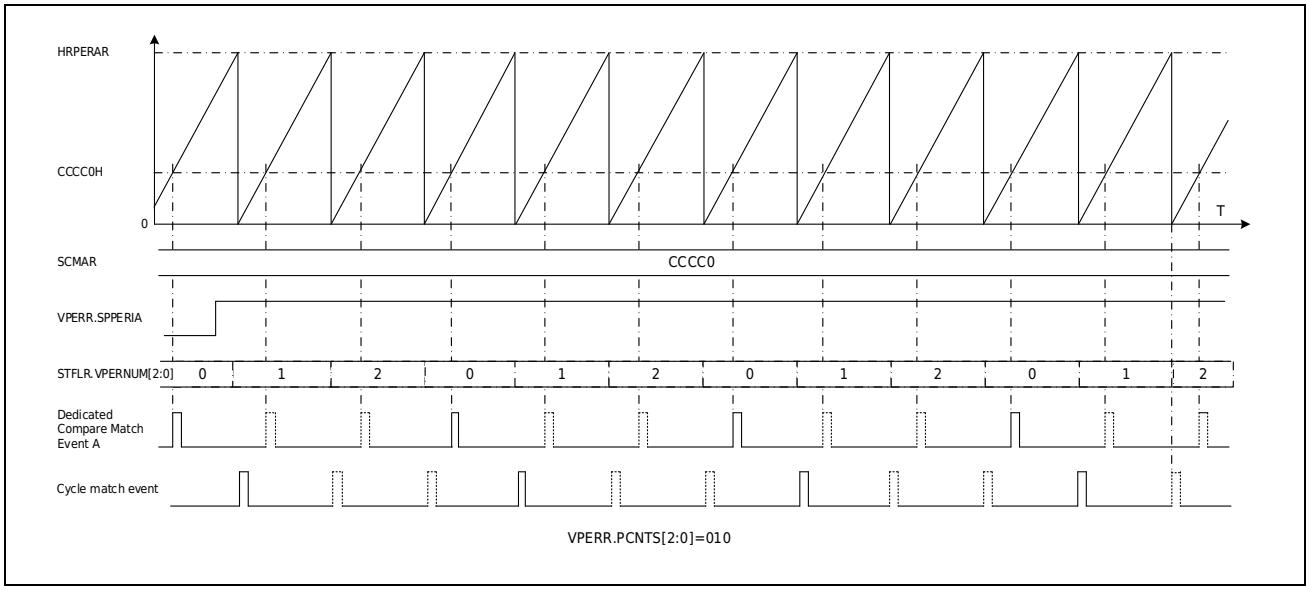


Figure 19-50 Cycle Interval Valid Request Signal Action

Note:

When the HRPWM<t>_VPERR.PCNTS[4:0] bit is not 0 and the HRPWM<t>_VPERR.SPPERIB bit is 1, the hardware clear, start and the special-compare match event B used as source of special-compare match event A source of each unit is an event after the periodic interval response processing. When the HRPWM<t>_VPERR.PCNTS[4:0] bits are not 0, the period/zero match events sent to the peripheral modules and used as the special-compare match event A source are the events after the interval response processing.

19.3.19 EMB(Fault) Control

The HRPWM has a versatile EMB(fault) protection circuitry to disable the outputs in case of an abnormal operation. Once a fault has been triggered, the outputs take a predefined safe state. This state is maintained until the output is re-enabled by software. In case of a permanent fault request, the output remains in its fault state, even if the software attempts to re-enable them, until the fault source disappears.

All units have 6 EMB(Faut) interfaces. Each unit selects the EMB event connection interface to be used through the setting of the port control register (HRPWM<t>_PCNA<B>R1.EMBSA<B>). This interface connects to a group of EMB events output by the EMB module. At the same time, the abnormal condition events gated on the interface can be set from the EMB side (see the EMB chapter). When abnormal conditions are detected on these interfaces, the control of the universal PWM output can be realized.

During the normal output period of the port, if the EMB event from the EMB is detected, the output state of the port can be changed to a predefined safe state. When an EMB abnormal event occurs, the PWM output port can change its state to output high impedance, output low level or output

high level (determined by the setting of HRPWM<t>_PCNA<B>R1.EMBCA<B>). For example, if HRPWM_PCNAR1.EMBCA=01 is set, then during the normal output of the HRPWM_PWMA port, if an EMB event occurs, the output on the HRPWM_PWMA port becomes a high-impedance state.

After the EMB event invalid (the abnormal event connected from the EMB module to the HRPWM disappears and the signal becomes a normal level), the output of the PWM port can automatically return to normal output. At this time, you can choose to restore the normal PWM output immediately (HRPWM<t>_PCNA<B>R1.EMBRA<B>=00; this method is called One Shot release) or restore the normal PWM output after period match or zero match (HRPWM<t>_PCNA<B>R1.EMBRA<B>=01, 10, 11; this method is called Cycle By Cycle release) through the setting of the port control register.

Note:

When the event release mode of EMB is configured as state release (configured by EMB_RLSSEL register), and the event source of EMB is configured as external port or the comparison result of internal voltage comparator valid, it must be ensured that the level width of the comparison result of external port and internal voltage comparator is greater than 3 count clock cycles, otherwise a glitch of up to 2 count clock cycles will appear on PWM output. When the event release mode of EMB is configured as software release (configured through the EMB_RLSSEL register), and the event source of EMB is configured as PWM output port in phase, a glitch of up to 2 count clock cycles will occur before PWM in-phase output occurs and PWM enters the safe state.

19.3.20 Output Management

The final stage of the output portis polarity selection and swap output selection, As shown in Figure 19-51.The polarity of the HRPWM_<t>_PWMA or HRPWM_<t>_PWMB port level can be inverted by configuring the INVCA<B>EN bit in the Universal control register 1 (HRPWM<t>_GCONR1).

By configuring the SWAPEN bit of HRPWM<t>_GCONR1, the outputs of the HRPWM_<t>_PWMA and HRPWM_<t>_PWMB ports can be swapped. The exchange supports two modes, immediate exchange or exchange at a Full period point (defined by HRPWM<t>_GCONR1.PRDSEL), and the exchange mode is selected by SWAPMD of HRPWM<t>_GCONR1.

When SWAPMD of HRPWM<t>_GCONR1 is 1, when a change in the SWAPEN bit of HRPWM<t>_GCONR1 is detected, the PWM will enter the idle cycle at a Full period point (defined by HRPWM<t>_GCONR1.PRDSEL) and swap outputs or output normally at the next Full period point, as shown in Figure 19-53.The level of the idle cycle is set by HRPWM<t>_IDLECR.IDLESA<B>. If the normal output state is switched to the exchange output state, the level of the idle cycle is set by each HRPWM<t>_IDLECR.IDLESA<B>; if the exchange output state is switched to the normal output state, the level of the idle cycle is set by the other party's HRPWM<t>_IDLECR.IDLESA<B>.

When SWAPMD of HRPWM<t>_GCONR1 is 0, PWMA and PWMB are swapped or not swapped immediately according to the SWAPEN bit of HRPWM<t>_GCONR1, seeFigure 19-52.

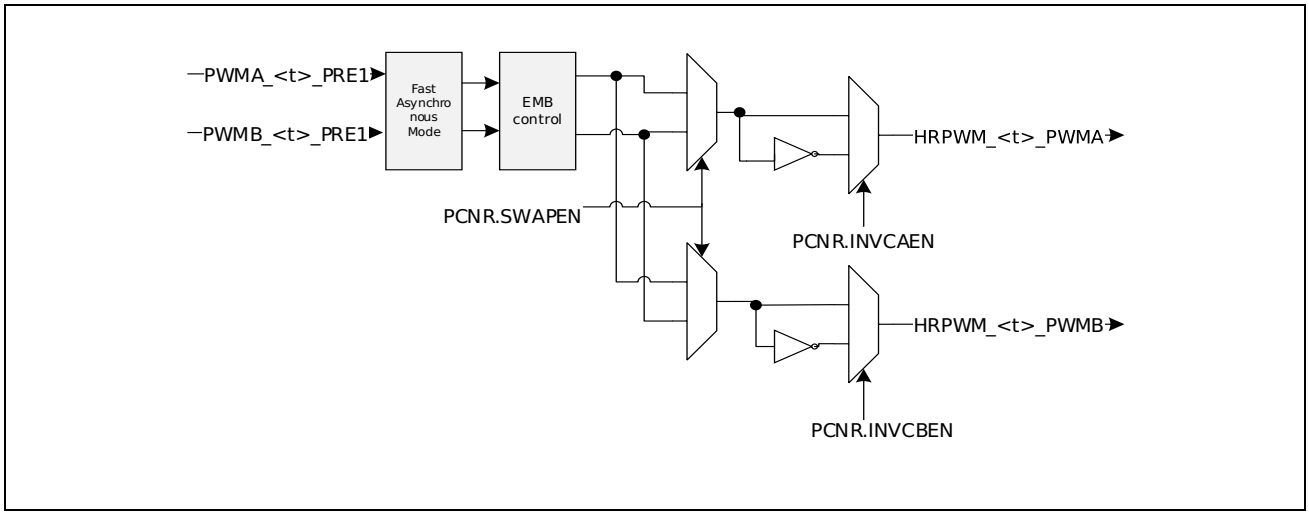


Figure 19-51 Output Management

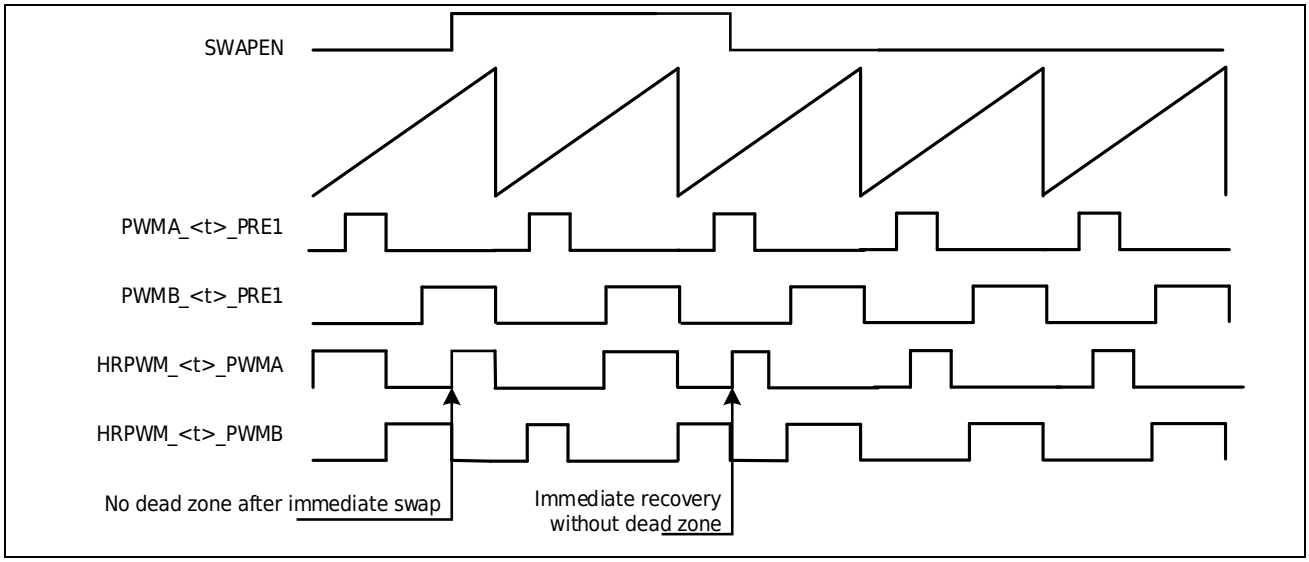


Figure 19-52 Immediately Swap Output Waveform

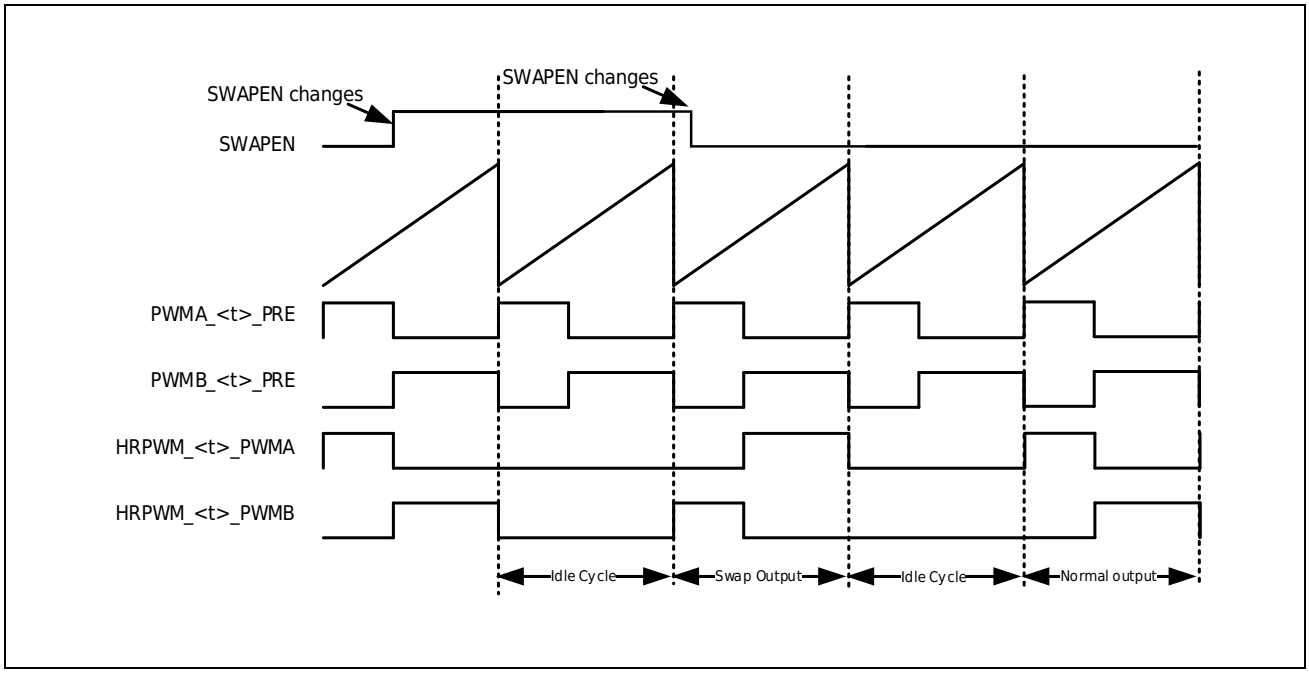


Figure 19-53 Full period point Switching Output Waveform

19.3.21 DAC Trigger Source

HRPWM includes three DAC synchronization trigger signals, which can trigger the data of two DACs on the chip to be updated synchronously with HRPWM.

By configuring HRPWM<t>_CR.DACSRC1<2> inside each unit, the source of the DAC synchronous trigger signal can be selected. The count value can be configured as 0, the count value can be configured as the period value(or hardware cleared in the sawtooth count mode), the count value can be configured as SCMAR, or the count value can be configured as SCMBR.

The above source is configured to one of the two DAC channel 1 and channel 2 synchronous trigger signals by selecting the internal HRPWM<t>_CR.DACSYNC1<2> of the configuration unit. All units are configured to source or post-send the synchronous trigger signal of the same DAC channel, see the figure below.

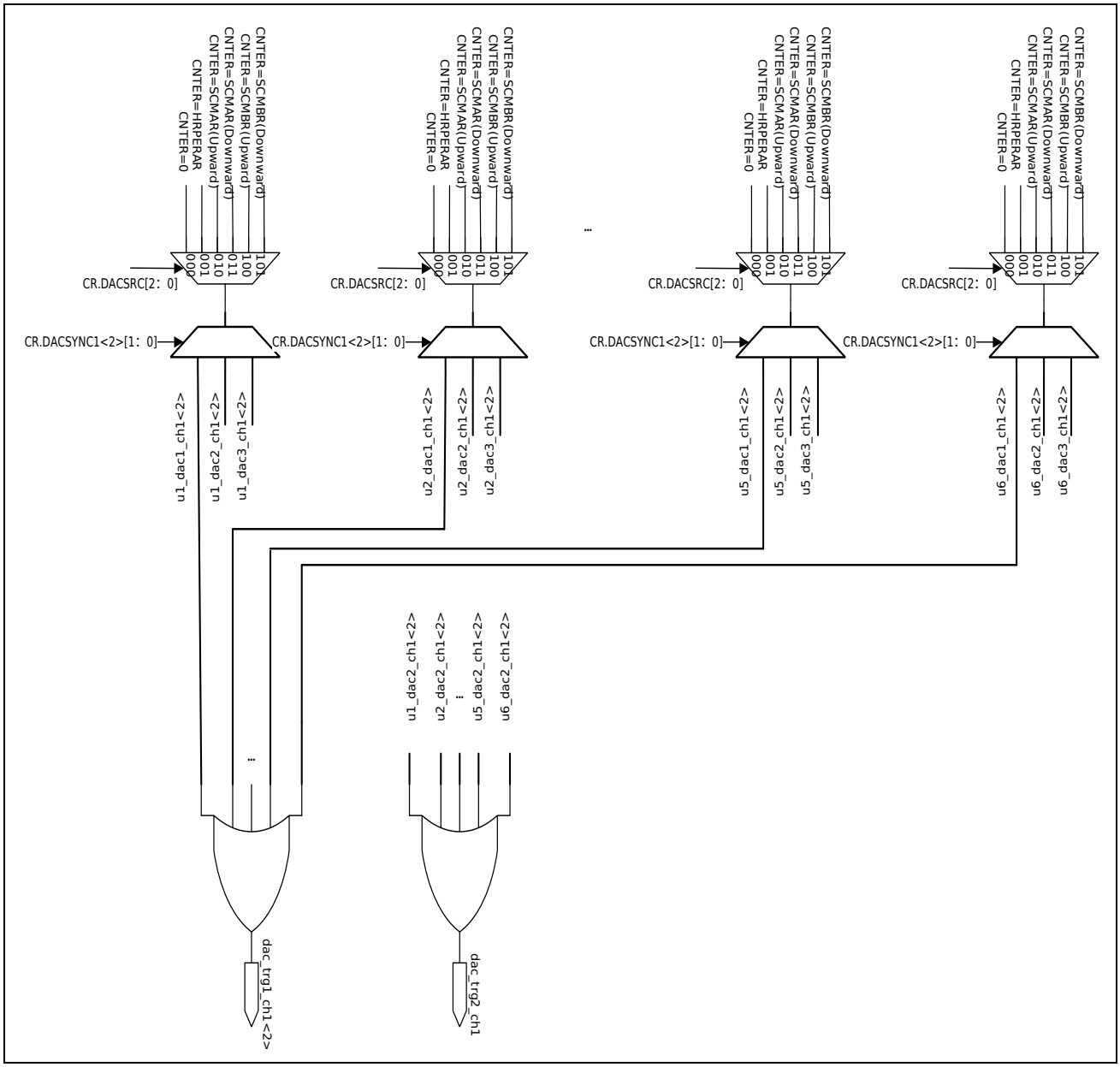


Figure 19-54 DAC Synchronous Trigger Source

19.3.22 Function Summary Table

The summary table of the main functions in the sawtooth wave mode and triangle count mode of HRPWM is shown in Table 19-13.

Table 19-13 Comparison of Functions in Different Modes

PWM output function			Sawtooth wave	Triangle wave	Related main registers
Port State Control	At startup		Support	Support	HRPWM<t>_PCNA<B>R1.STACA<B>
	stop		Support	Support	HRPWM<t>_PCNA<B>R1.STPCA<B>
	period matching		Support	Support	HRPWM<t>_PCNA<B>R1.OVFCA<B>
	Zero point matching		Support	Support	HRPWM<t>_PCNA<B>R1.UDFCA<B>
	External events occur (Up Counting)		Support	Support	HRPWM<t>_PCNA<B>R2.EXEVyUCA<B>
	External events occur (Down Counting)		Support	Support	HRPWM<t>_PCNA<B>R3.EXEVyDCA<B>
	When counting matches (Up Counting)		Support	Support	HRPWM<t>_PCNA<B>R1.CMA<B>UCA<B> HRPWM<t>_PCNA<B>R2.CME<F>UCA<B> HRPWM<t>_PCNA<B>R2.SCMA<B>UCA<B>
	When counting matches (Down Counting)		Support	Support	HRPWM<t>_PCNA<B>R1.CMA<B>DCA<B> HRPWM<t>_PCNA<B>R3.CME<F>DCA<B> HRPWM<t>_PCNA<B>R3.SCMA<B>DCA<B>
Shadow transfer	Burst Mode period value	Single Shadow	Support	Support	HRPWM_BMPERBR-> HRPWM_BMPERAR
	Burst Mode compare value	Single Shadow	Support	Support	HRPWM_BMCMBR-> HRPWM_BMCMAR
	External event filter offset value	Single Shadow	Support	Support	HRPWM<t>_EEFOFFSETBR-> HRPWM<t>_EEFOFFSETAR
	External event filter window	Single Shadow	Support	Support	HRPWM<t>_EEFWINBR-> HRPWM<t>_EEFWINAR
	High-resolution period	Single Shadow	Support	Support	HRPWM<t>_HRPERBR-> HRPWM<t>_HRPERAR
	High-resolution general compare	Single Shadow	Support	Support	HRPWM<t>_HRGCMDR-> HRPWM<t>_HRGCMBR HRPWM<t>_HRGCMCR-> HRPWM<t>_HRGCMAR HRPWM<t>_HRGCMGR-> HRPWM<t>_HRGCMER HRPWM<t>_HRGCMHR-> HRPWM<t>_HRGCMFR
	Port control register	Single Shadow	Support	Support	HRPWM<t>_BPCNA<B>R1-> HRPWM<t>_PCNA<B>R1 HRPWM<t>_BPCNA<B>R2-> HRPWM<t>_PCNA<B>R2 HRPWM<t>_BPCNA<B>R3-> HRPWM<t>_PCNA<B>R3
	High resolution dead zone	Single Shadow	Support	Support	HRDTUBR->HRDTUAR HRDTDBR->HRDTDAR
	Interrupt control register	Single Shadow	Support	Support	HRPWM<t>_BICONR-> HRPWM<t>_ICONR

PWM output function			Sawtooth wave	Triangle wave	Related main registers
	Universal control register	Single Shadow	Support	Support	HRPWM<t>_BGCONR1-> HRPWM<t>_ GCONR1
PWM output without dead zone			Support	Support	HRPWM<t>_DCONR.DTCEN=0
PWM output with dead zone			Support	Support	HRPWM<t>_DCONR.DTCEN=1
Burst Mode			Support	Support	HRPWM<t>_IDLECR.IDLEBMA<B>
Delayed Idle Mode			Support	Support	HRPWM<t>_IDLECR.DLYPRT
Idle now			Support	Support	HRPWM_SSTAIDLR. SSTAIDL<t>A<B>
EMB			Support	Support	HRPWM<t>PCNA<B>R1.EMBCA<B>

19.4Interrupt

19.4.1 Interrupt Output

HRPWM contains 6 high-resolution general count compare match interrupts, 2 special-compare count compare match interrupts, 1 period match interrupt, 1 zero match interrupt, 2 capture interrupts, 1 Burst Mode period match interrupt, 1 calibration completion interrupt and 1 delayed idle trigger interrupt.

19.4.1.1 Count Comparison Match Interrupt

There are 6 **High-Resolution General Compare Registers** (HRPWM<t>_HRGCMAR~HRPWM<t>_HRGCMFR), which can be compared with the count value to generate comparison match. When the count compare matches, the HRPWM<t>_STFLR1.CMAF~STFLR1.CMFF bits in the status flag register (HRPWM<t>_STFLR1) will be set to 1 respectively. At this time, if the corresponding bit in INTENA~INTENF of the interrupt control register (HRPWM<t>_ICONR) is set to 1 to enable interrupt, the corresponding interrupt request (HRPWM_<t>_GCMA~F) will also be triggered.

The two Special-Compare Registers (HRPWM<t>_SCMAR~HRPWM<t>_SCMBR) can also be compared with the count value to generate a compare match.When the count compare matches, the HRPWM<t>_STFLR1.CMSPAF~CMSPBF bits in the status flag register (HRPWM<t>_STFLR1) will be set to 1 respectively. At this time, if the corresponding bit in INTENSAU<D> or INTENSBU<D> of the interrupt control register (HRPWM<t>_ICONR) is set to 1 to enable interrupt, the corresponding interrupt request (HRPWM_<t>_SCMA~B) will also be triggered.

19.4.1.2 Period/Zero Match Interrupt

When the count value of the counter is equal to the period value or hardware clear occurs int sawtooth count mode, the HRPWM<t>_STFLR1.OVFF of the status flag register (HRPWM<t>_STFLR1) will be set to 1. At this time, if the HRPWM<t>_ICONR.INTENOVF bit of the interrupt control register (HRPWM<t>_ICONR) is set to enable the interrupt, a period match interrupt (HRPWM_<t>_GOVF) will be generated.

When the counter value is equal to 0, the HRPWM<t>_STFLR1.UDFF bit will be set to 1. At this time, if the interrupt control register (HRPWM<t>_ICONR) HRPWM<t>_ICONR.INTENUDF bit is set to enable interrupt, the count zero match interrupt (HRPWM_<t>_GUDF) can be triggered at the corresponding time point.

19.4.1.3 Capture Interrupt

When the hardware capture is enabled (HRPWM<t>_HCPA<B>R2. HCPA<B>EN) and the capture input valid condition selected by the hardware capture event selection register (HRPWM<t>_HCPAR1<2>, HRPWM<t>_HCPBR1<2>) occurs, or when the software synchronization capture is triggered, the capture action occurs and the HRPWM<t>_STFLR1. CAPA<B>F bit in the status flag register (HRPWM<t>_STFLR1) will be set to 1. At this time, if the INTENCAPA or INTENCAPB bit of the interrupt control register (HRPWM<t>_ICONR) is set to 1 to enable interrupt, the corresponding interrupt request (HRPWM_<t>_GCAPA<B>) is triggered.

19.4.1.4 Burst Mode Period Match Interrupt

When the BM-counter counts up to the period value, the BMOVFF bit in the Burst Mode control register (BMCR) will be set to 1. At this time, if the INTENBMOVF enable interrupt of the Burst Mode control register (BMCR) is set, the Burst Mode period match interrupt (HRPWM_BMOVF) can be triggered at the corresponding time point.

19.4.1.5 Calibration Complete Interrupt

When the calibration function is used and the calibration is completed, CALENF in the High Resolution Calibration Control Register (CALCR) will be set to 1. If the CALIE interrupt of the High Resolution Calibration Control Register (CALCR) is enabled, a calibration complete interrupt (HRPWM_CALENF) can occur when calibration is complete.

19.4.1.6 Delayed Idle Trigger Interrupt

When delayed idle is enabled, once a trigger signal is detected, the DLYPRT bit in the status flag register (HRPWM<t>_STFLR2) will be set to 1. At this time, if the INTENDLYPRT enable interrupt of the output control register (HRPWM<t>_IDLECR) is set, a delayed idle trigger interrupt (HRPWM_DLYPRT) can be generated. All units share one interrupt, and the unit in which delayed idle occurs is confirmed by querying the DLYPRT bit in the HRPWM<t>_STFLR2 register of each unit.

19.4.2 Event Output

During the clock counting process, if a period match event, a zero match event (general count compare match event, a special-compare count compare match event, a capture event, or an external event 1 to 10 detection event occurs, a corresponding event output signal will be generated to select and trigger other modules, such as ADC, DMA, etc.

External event 1~10 detection output events are selected by setting the EEVy register bit of the External Event Eetection Eelection Register (HRPWM_EEDSELR). When the selected external event occurs, an external event detection output event (HRPWM_EEDET) is generated.

The special-compare match event A and special-compare match event B of each unit can be generated in the counter up-counting or down-counting, and different effective action directions can be enabled by configuring HRPWM<t>_GCONR1.CMSCA<B>U<D>EN.

Each unit's special-compare count match event A can not only enable the unit's special-compare compare match event A as an event source, but also select multiple event sources as the source of special-compare match event A by configuring the Special-Compare Match Selection Register A (HRPWM<t>_SCMASELR), including special-compare match events A~B of units 1~6, period match events of units 1~6, zero match events of units 1~6, and external event detection events. When the value of the register HRPWM_<t>_SCMASELR is 0, the special-compare match event A of this unit is selected as the event source by default.

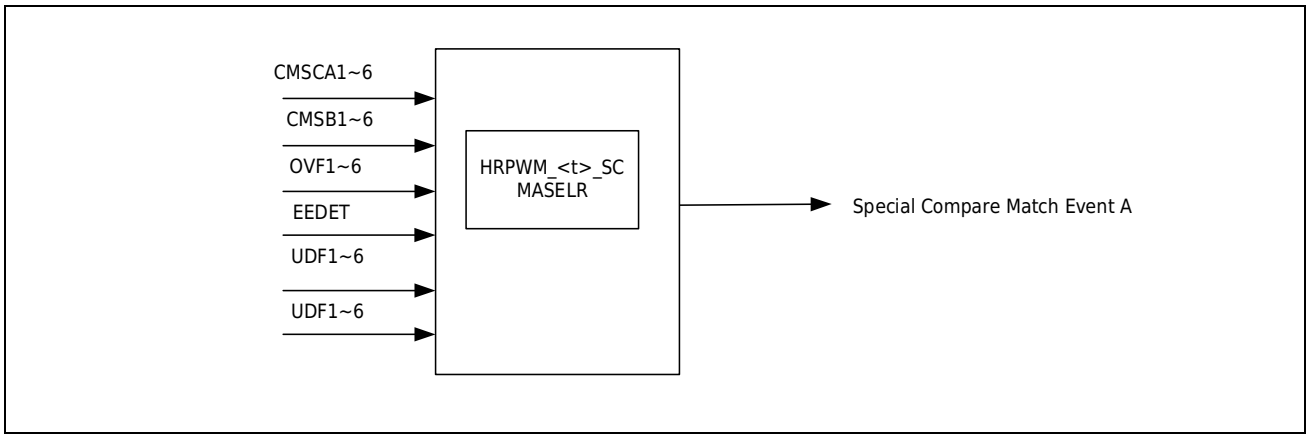
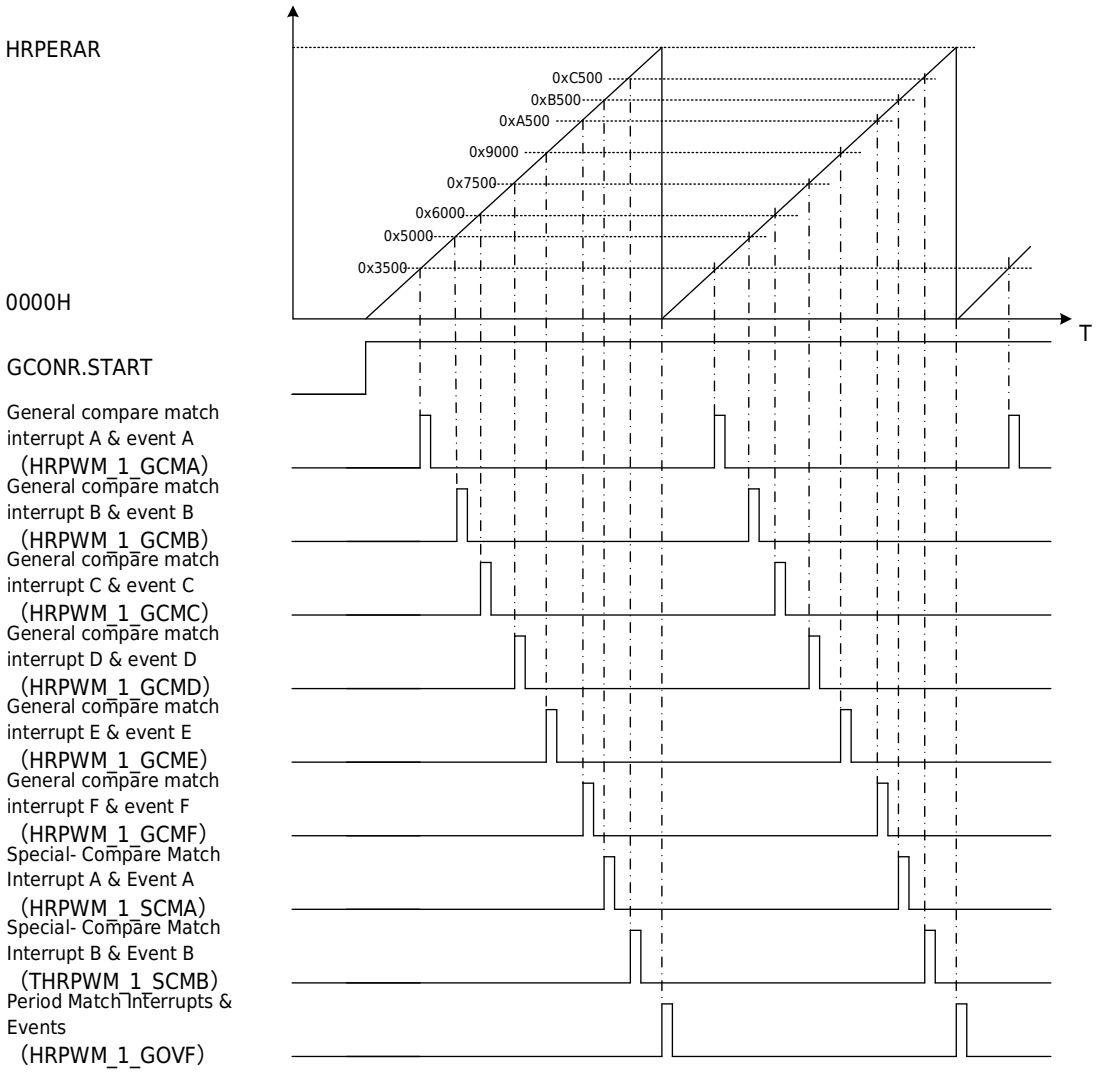


Figure 19-55 Unit 1~3 Special-compare Count Match Event

The following figure shows the operation example of the universal compare match interrupt A~F && events A~F, the special-compare compare match interrupt A~B && events A~B, and the period match interrupt && events of unit 1.




```
HRPERAR=0xFFFF
HRGCMAR=0x3500, HRGCMBR=0x5000, HRGCMCR=0x6000, HRGCMDR=0x7500
HRGCMER=0x9000, HRGCMFR=0xA500
SCMAR=0xB500, SCMBR=0xC500
ICONR.INTENA~F=0b1
ICONR.INTENSAU=ICONR.INTENSUB=0b1
ICONR.INTENOVF=0b1
```

Figure 19-56 Example of Interrupt & Event Output in Sawtooth Wave Mode

19.5Typical Application

The following describes the basic settings of HRPWM related registers in several typical application situations for user reference.

19.5.1 Calibration Function

Before enabling HRPWM, a calibration is required according to the following steps.

Single Calibration

a) Configure the enable clock PCLK0
b) Configure HRPWM_CALCR.CAL = 1 to enable single calibration
c) Wait for HRPWM_CALCR.CALENF = 1
d) Confirm HRPWM_CALCR.ERRF=0

Periodic Calibration

a) Configure the enable clock PCLK0
b) Configure HRPWM_CALCR.CALEN=1 to enable period calibration
c) Wait for HRPWM_CALCR.CALENF = 1
d) Confirm HRPWM_CALCR.ERRF=0

19.5.2 Basic Counting and Interrupt Actions

a) Follow 19.5.1 the setup steps to perform a calibration
b) Enable HRPWM (HRPWM<t>_CR.EN=1)
c) Set the high-resolution period value (HRPWM<t>_HRPERAR)
d) Set the general compare values, including high-resolution general compare (HRPWM<t>_HRGCMAR~HRPWM<t>_HRGCMFR), special-compare (HRPWM<t>_SCMAR~HRPWM<t>_SCMBR), etc.
e) Set the required Interrupt Enable Bits, including count cycle interrupt (HRPWM<t>_ICONR.INTENOVF), count zero interrupt (HRPWM<t>_ICONR.INTENUDF), count match interrupt (HRPWM<t>_ICONR.INTENA~F, HRPWM<t>_ICONR.INTENSAU, HRPWM<t>_ICONR.INTENSAD, HRPWM<t>_ICONR.INTENSBU, HRPWM<t>_ICONR.INTENSBD), etc.
f) Set counter mode (HRPWM<t>_GCONR.MODE)
g) Start the counter (HRPWM<t>_GCONR.START=1)

19.5.3 Compare Output and Interrupt Action

a) Perform a calibration according to the calibration function setup steps
b) Enable HRPWM (HRPWM<t>_CR.EN=1)
c) Set the high-resolution period A Register (HRPWM<t>_HRPERAR)
d) Set the compare value of each channel, including high-resolution general compare A, E (HRPWM<t>_HRGCMA<E>R), high-resolution general compare B, F (HRPWM<t>_HRGCMB<F>R)
e) Set the required Interrupt Enable Bits, including count cycle interrupt (HRPWM<t>_ICONR.INTENOVF), count zero interrupt (HRPWM<t>_ICONR.INTENUDF), count match interrupt (HRPWM<t>_ICONR.INTENA~B), etc.
f) Set the port output status of each channel in different counting states (refer to the related control of HRPWM<t>_PCNAR1<2><3>, HRPWM<t>_PCNBR1<2><3>, etc.)
g) Set waveform mode (HRPWM<t>_GCONR.MODE)

h) Set each channel output enable (HRPWM<t>_PCNAR1.OUTENA=1, HRPWM<t>_PCNBR1.OUTENB=1)
i) Set exit immediate idle (HRPWM_SSTARUNR. SSTARUN<t>A<B>=1)
j) Start the counter (HRPWM<t>_GCONR.START=1)

19.5.4 Burst Mode And Interrupt Action

a) Refer to 19.5.3 steps a~i in the chapter to set up each unit that needs to be started.
b) Set the required Interrupt Enable Bit, BM-counter overflow interrupt (HRPWM_BMCR.INTENBMOVF=1)
c) Enable Burst Mode (HRPWM<t>_IDLECR.IDLEBMA, HRPWM<t>_IDLECR.IDLEBMB)
d) Configure the state of the PWM idle state (HRPWM<t>_IDLECR.IDLESA, HRPWM<t>_IDLECR.IDLESB)
e) Set the Burst Mode Period Rregister (HRPWM_BMPERAR)
f) Set the Burst Mode Compare Rregister (HRPWM_BMCMAR)
g) Set Burst Mode operation mode (HRPWM_BMCR.BMCTN)
h) Set the Burst Mode count clock (HRPWM_BMCR.BMCLKS[1:0], HRPWM_BMCR.BMPSC[3:0])
i) Set the counter operation status in the unit under Burst Mode (HRPWM_BMCR.BMTMR)
j) Enable Burst Mode (HRPWM_BMCR.BMEN=1)
k) Start the counter (HRPWM<t>_GCONR.START=1)
l) Start Burst Mode (HRPWM_BMSTRG.SSTRG=1)

19.5.5 Shadow Transfer Action (Period Reference Value)

a) Perform a calibration according to the calibration function setup steps
b) Enable HRPWM (HRPWM<t>_CR.EN=1)
c) Set the required high-resolution period value (HRPWM<t>_HRPERAR, HRPWM<t>_HRPERBR)
d) Set the shadow transfer time point (HRPWM<t>_BCONR1.BTRUP, HRPWM<t>_BCONR1.BTRDP, HRPWM<t>_BCONR2.BTRU0P, HRPWM<t>_BCONR2.BTRU0PP (valid only when the phase function is enabled))
e) Set Counter mode (HRPWM<t>_GCONR.MODE)
f) Set the shadow function to be valid (HRPWM<t>_BCONR1.BENP=1)
g) Start the counter (HRPWM<t>_GCONR.START=1)
h) Waiting for the corresponding Shadow transfer time point, Shadow action occurs (HRPWM<t>_HRPERBR->H HRPWM<t>_RPERAR)

19.5.6 Shadow Transfer Action (general compare)

a) Perform a calibration according to the calibration function setup steps
b) Enable HRPWM (HRPWM<t>_CR.EN=1)
c) Set the required High-Resolution General Compare Rregister

(HRPWM<t>_HRGCMAR~HR)

d) Set the shadow transfer time point of each channel (HRPWM<t>_BCONR1.BTRUAE, HRPWM<t>_BCONR1.BTRDAE, HRPWM<t>_BCONR1.BTRUBF, HRPWM<t>_BCONR1.BTRDBF, HRPWM<t>_BCONR2.BTRUP0AE, HRPWM<t>_BCONR2.BTRU0PBF, HRPWM<t>_BCONR2.BTRU0AE, HRPWM<t>_BCONR2.BTRU0BF (valid only when the phase function is enabled))

e) Set counter mode (HRPWM<t>_GCONR.MODE)

f) Set each channel shadow function to be valid (HRPWM<t>_BCONR1.BENAE=1, HRPWM<t>_BCONR1.BENBF=1)

g) Start the counter (HRPWM<t>_GCONR.START=1)

h) Waiting for the corresponding buffer transfer time point set for each channel, the shadow action occurs (HRPWM<t>_HRGCMCR-> HRPWM<t>_HRGCMAR, HRPWM<t>_HRGCMDR->HRGCMBR, HRPWM<t>_HRGCMGR-> HRPWM<t>_HRGCMER, HRPWM<t>_HRGCMHR-> HRPWM<t>_HRGCMFR)

19.5.7 Shadow Transfer Action (Special-compare)

a) Perform a calibration according to the calibration function setup steps

b) Enable HRPWM (HRPWM<t>_CR.EN=1)

c) Set the required special-compare Register (HRPWM<t>_SCMAR, HRPWM<t>_SCMCR, HRPWM<t>_SCMBR, HRPWM<t>_SCMDR)

d) Set the shadow transfer time point of each channel (HRPWM<t>_BCONR.BTRUSPA, HRPWM<t>_BCONR1.BTRDSPA, HRPWM<t>_BCONR1.BTRUSPB, HRPWM<t>_BCONR1.BTRDSPB, HRPWM<t>_BCONR2.BTRU0PSPA, HRPWM<t>_BCONR2.BTRU0PSPB, HRPWM<t>_BCONR2.BTRU0SPA, HRPWM<t>_BCONR2.BTRU0SPB (valid only when the phase function is enabled))

e) Set counter mode (HRPWM<t>_GCONR.MODE)

f) Set each channel buffer function to be valid (HRPWM<t>_BCONR1.BENSPA=1, HRPWM<t>_BCONR1.BENSPB=1)

g) Start the counter (HRPWM<t>_GCONR.START=1)

h) Waiting for the corresponding shadow transfer time point set for each channel, and the buffer action occurs (HRPWM<t>_SCMCR-> HRPWM<t>_SCMAR, HRPWM<t>_SCMDR-> HRPWM<t>_SCMBR)

19.5.8 Shadow Transfer Action (External Event Filter Offset Value)

a) Perform a calibration according to the calibration function setup steps

b) Enable HRPWM (HRPWM<t>_CR.EN=1)

c) Set the required external event filter offset value (HRPWM<t>_EEFOFFSETAR, HRPWM<t>_EEFOFFSETBR)

d) Set the buffer transfer time point of each channel (HRPWM<t>_BCONR2.BTRUEEFOFF, BCONR2.BTRDEEFOFF, HRPWM<t>_BCONR2.BTRU0PEEFOFF, HRPWM<t>_BCONR2.BTRU0EEFOFF (valid only when the phase function is enabled))
e) Set waveform mode (HRPWM<t>_GCONR.MODE)
f) Set each channel buffer function to be valid (HRPWM<t>_BCONR2.BENEEFOFF=1)
g) Start the counter (HRPWM<t>_GCONR.START=1)
h) Waiting for the corresponding buffer transfer time point set for each channel, and the buffer action occurs (HRPWM<t>_EEFOFFSETBR->HRPWM<t>_EEFOFFSETAR)

19.5.9 Shadow Transfer Action (External Event Filter Window)

a) Perform a calibration according to the calibration function setup steps
b) Enable HRPWM (HRPWM<t>_CR.EN=1)
c) Set the required external event filter window (HRPWM<t>_EEFWINAR, HRPWM<t>_EEFWINBRR)
d) Set the buffer transfer time point of each channel (HRPWM<t>_BCONR2.BTRUEEFWIN, HRPWM<t>_BCONR2.BTRDEEFWIN, HRPWM<t>_BCONR2.BTRU0PEEFWIN, HRPWM<t>_BCONR2.BTRU0EEFWIN (valid only when the phase function is enabled))
e) Set waveform mode (HRPWM<t>_GCONR.MODE)
f) Set each channel buffer function to be valid (HRPWM<t>_BCONR2.BENEEFWIN=1)
g) Start the counter (HRPWM<t>_GCONR.START=1)
h) Waiting for the corresponding buffer transfer time point set for each channel, and the buffer action occurs (HRPWM<t>_EEFWINBR->HRPWM<t>_EEFWINAR)

19.5.10 Buffer Transfer Action (Burst Mode period)

a) Refer to steps a to f in the comparison output and interrupt action section to set each unit that needs to be started.
b) Set the required Burst Mode Period Rregister (HRPWM_BMPERAR, HRPWM_BMPERBR)
c) Set the Burst Mode Compare Rregister (HRPWM_BMCMAR, HRPWM_BMCMBR) (When using the Burst Mode Period Rregister Shadow function, the comparison value needs to be written again after writing the period value)
d) Set the buffer function to be enabled (HRPWM_BMCR. BENBMP=1)
e) Set the state of the PWM port when the Burst Mode is idle (HRPWM<t>_IDLECR.IDLES<A>B, HRPWM<t>_IDLECR.IDLEBM<A>B)
f) Set Burst Mode operation mode (HRPWM_BMCR.BMCTN)
g) Set the Burst Mode count clock (HRPWM_BMCR.BMCLKS[1:0], HRPWM_BMCR.BMPSC[3:0])
h) Set the counter operation status in the unit under Burst Mode (HRPWM_BMCR. BMTMR<t>)
i) Enable Burst Mode (HRPWM_BMCR.BMEN=1)
j) Start the counter (HRPWM<t>_GCONR.START=1)

k) Start Burst Mode (HRPWM_BMSTRG.SSTRG=1)

l) Waiting for the corresponding buffer transfer time point to occur buffer action (HRPWM_BMPERBR->HRPWM_BMPERAR)

19.5.11 Buffer Transfer Operation (Burst Mode Compare Register)

a) Refer to steps a to f in the comparison output and interrupt action section to set each unit that needs to be started.

b) Set the Burst Mode Period Rregister (HRPWM_BMPERAR)

c) Set the required Burst Mode Compare Rregister (HRPWM_BMCMAR, HRPWM_BMCMBR)

d) Set the buffer function to be enabled (HRPWM_BMCR. BENBMCMP = 1)

e) Set the state of the PWM port when the Burst Mode is idle (HRPWM<t>_IDLECR.IDLES<A>B, HRPWM<t>_IDLECR.IDLEBM<A>B)

f) Set Burst Mode operation mode (HRPWM_BMCR.BMCTN)

g) Set the Burst Mode count clock (HRPWM_BMCR.BMCLKS[1:0], HRPWM_BMCR.BMPSC[3:0])

h) Set the counter operation status in the unit under Burst Mode (HRPWM_BMCR. BMTMR<t>)

i) Enable Burst Mode (HRPWM_BMCR.BMEN=1)

j) Start the counter (HRPWM<t>_GCONR.START=1)

k) Start Burst Mode (HRPWM_BMSTRG.SSTRG=1)

l) Waiting for the corresponding buffer transfer time point to occur buffer action (HRPWM_BMCMBR->HRPWM_BMCMAR)

19.5.12 Buffer Transfer Action (Deadband)

a) Perform a calibration according to the calibration function setup steps

b) Enable HRPWM (HRPWM<t>_CR.EN=1)

c) Set the required high-resolution dead-times (HRPWM<t>_HRDTUAR, HRPWM<t>_HRDTUBR, HRPWM<t>_HRDTDAR, HRPWM<t>_HRDTDBR)

d) Set the buffer transfer time point (HRPWM<t>_DCONR.DTBTRU, HRPWM<t>_DCONR.DTBTRD, HRPWM<t>_BCONR2.BTRU0PDT, HRPWM<t>_BCONR2.BTRU0DT (valid only when the phase function is enabled))

e) Set waveform mode (HRPWM<t>_GCONR.MODE)

f) Set the high-resolution buffer function to be enabled (HRPWM<t>_DCONR.DTBENU=1, HRPWM<t>_DCONR.DTBEND=1)

g) Set the hardware dead zone function to be valid (HRPWM<t>_DCONR.DTCEN=1)

h) Start the counter (HRPWM<t>_GCONR.START=1)

i) Waiting for the corresponding Shadow transfer time point, Shadow action occurs (HRPWM<t>_HRDTUBR-> HRPWM<t>_HRDTUAR, HRPWM<t>_HRDTDBR-> HRPWM<t>_HRDTDAR)

19.5.13 Synchronous Start Action (Software Method)

a) Refer to steps a~f in the [Basic Counting and Interrupt Action] chapter to set the units that need to be started synchronously
b) Synchronous start counter (set the corresponding bit of the HRPWM_SSTAR register to 1, each unit corresponds to one register bit)

19.5.14 Synchronous Start Action (Hardware Method)

a) Perform a calibration according to the calibration function setup steps
b) Enable HRPWM (HRPWM<t>_CR.EN=1)
c) Set the high-resolution period value (HRPWM<t>_HRPERAR)
d) Set the required comparison reference values, including High-Resolution General Compare Rregisters (HRPWM<t>_HRGCMAR~HRGCMFR), special compare s (HRPWM<t>_SCMAR~SCMBR), etc.
e) Set the required Interrupt Enable Bits, including count cycle interrupt (HRPWM<t>_ICONR.INTENOVF), count zero interrupt (HRPWM<t>_ICONR.INTENUDF), count match interrupt (HRPWM<t>_ICONR.INTENA~F, HRPWM<t>_ICONR.INTENSAU, HRPWM<t>_ICONR.INTENSAD, HRPWM<t>_ICONR.INTENSBU, HRPWM<t>_ICONR.INTENSBD), etc.
f) Set hardware start conditions (selected by HRPWM<t>_HSTAR1, HRPWM<t>_HSTAR2)
g) Set hardware start enable (HRPWM<t>_HSTAR1.STAS=1)
h) Repeat steps a~g above to make settings for each unit that needs to be started synchronously (in each unit that needs to be started synchronously, the setting of step d must be consistent)
i) Wait for the set trigger event to occur, and confirm that the counters of each unit start synchronously

19.5.15 Sync Output

a) Refer to steps a to f in the [Comparison Output and Interrupt Action] section to set the required comparison reference values, including High-Resolution General Compare Rregisters (HRPWM<t>_HRGCMAR to HRPWM<t>_HRGCMFR), special compare s (HRPWM<t>_SCMAR to HRPWM<t>_SCMBR), etc.
b) Set the output source of the synchronous output signal (HRPWM_SYNOCR.SYNCSRC[3:0])
c) Set the counting direction of the synchronous output signal source (HRPWM_SYNOCR.SRCDIR)
d) Set the output pulse type (HRPWM_SYNOCR.SYNOPLS[1:0])
e) Set the synchronous output signal pulse length (HRPWM_SYNOCR. SYNCCMP[7:0])
f) Start the counter (HRPWM<t>_GCONR.START=1)
g) Wait for the set output source to occur and confirm that HPRPWM_SCOUT outputs the

required pulse

19.5.16 Single PWM Output

Refer to the settings in steps a~f of the [Comparison Output and Interrupt Action] section (the output states of the two PWM channels HRPWM_<t>_PWMA and HRPWM_<t>_PWMB in each unit can be set independently to form two independent single-channel PWM outputs)

19.5.17 Complementary PWM Output (Software Dead Time)

a) Perform a calibration according to the calibration function setup steps
b) Enable high resolution (HRPWM<t>_CR.EN=1)
c) Set the common high-resolution period reference value (HRPWM<t>_HRPERAR)
d) Set the High-Resolution General Compare Rregister A and E (HRPWM<t>_HRGCMA<E>R) and High-Resolution General Compare Rregister B and F (HRPWM<t>_HRGCMB<F>R).
e) Set the required Interrupt Enable Bits, including count cycle interrupt (HRPWM<t>_ICONR.INTENOVF), count zero interrupt (HRPWM<t>_ICONR.INTENUDF), count match interrupt (HRPWM<t>_ICONR.INTENA~B), etc.
f) Set the port output status at different counting states (refer to the related control of HRPWM<t>_PCNAR1<2><3>, HRPWM<t>_PCNBR1<2><3>, set the appropriate High-Resolution General Compare Rregister to ensure the formation of complementary dead zones between the two PWM outputs)
g) Set waveform mode (HRPWM<t>_GCONR.MODE)
h) Set output enable (HRPWM<t>_PCNAR1.OUTENA=1, HRPWM<t>_PCNBR1.OUTENB=1)
i) Set exit immediate idle (HRPWM_SSTARUNR<t>. SSTARUN1A<B>=1)
j) Start counter (GCONR.START=1)

19.5.18 Complementary PWM Output (Hardware Dead Zone)

a) Perform a calibration according to the calibration function setup steps
b) Enable HRPWM (CR.EN=1)
c) Setting the High Resolution Universal Periodic Reference Value (HRPERAR)
d) Set the high-resolution common comparison reference values A and E (HRGCMA<E>R) and the high-resolution common comparison reference values B and F (HRGCMB<F>R).
e) Set the required Interrupt Enable Bits, including count cycle interrupt (ICONR.INTENOVF), count zero interrupt (ICONR.INTENUDF), count match interrupt (ICONR.INTENA~B), etc.
f) Set the output state of the internal output signal (HRPWM_<t>_PWMA_ORG) of channel A at different counting states (refer to the relevant control of PCNAR1<2><3>, combined with the setting values of HRGCMAR, HRDTUAR and HRDTDAR, it is necessary to ensure that the two PWM outputs form a complementary dead zone)
g) Set waveform mode (GCONR.MODE)
h) Set each channel output enable (PCNAR.OUTENA=1, PCNBR.OUTENB=1)

i) Set exit immediate idle (HRPWM_SSTARUNR. SSTARUN<t>A<B>=1)

j) Set the hardware dead zone function to be valid (HRPWM<t>_DCONR.DTCEN=1)

k) Start the counter (HRPWM<t>_GCONR.START=1)

19.5.19 EMB Monitoring and Interrupt Action

a) Refer to steps a~i in chapter [Complementary PWM output (software dead zone)] or steps a~j in chapter [Complementary PWM output (hardware dead zone)] to set the complementary PWM output action

b) Set the state of the PWM port when an EMB event occurs (HRPWM<t>_PCNAR1.EMBCA, HRPWM<t>_PCNBR1.EMBCB) (Select the corresponding protection state according to different system applications)

c) Set the time point when the PWM port resumes normal output when the EMB event becomes invalid (HRPWM<t>_PCNAR1.EMBRA, HRPWM<t>_PCNBR1.EMBRB)

d) Set and select the EMB event source input from the EMB module (HRPWM<t>_PCNAR1.EMBSA, HRPWM<t>_PCNBR1.EMBSB)

e) Set the relevant registers of the EMB module (including EMB interrupt permission register (EMB_INTEN0~5), EMB control register 1/2 (EMB_CTL1/2_0~5), EMB control PWM output release mode selection register (EMB_RLSSEL0~5), etc.)

f) Start the counter (HRPWM<t>_GCONR.START=1), and the EMB module monitors the system status in real time

19.6Register Description

Table 19-14 HRPWN Base Address

Name	Base Address	Description
HRPWM1	0x4003 C000	HRPWM1 base address
HRPWM2	0x4003 C400	HRPWM2 base address
HRPWM3	0x4003 C800	HRPWM3 base address
HRPWM4	0x4003 CC00	HRPWM4 base address
HRPWM5	0x4003 D000	HRPWM5 base address
HRPWM6	0x4003 D400	HRPWM6 base address
HRPWM_COMMON	0x4003 C2F0	HRPWM common register base address

Table 19-15 HRPWM Register List

Register	Description	Offset Address	Bit Width	Reset Value
HRPWM<t>_CNTER	count register	0x0000	32	0x0000 0000
HRPWM<t>_UPDAR	Update Register	0x0004	32	0x0000 0000
HRPWM<t>_HRPERAR	High-Resolution Period Register A	0x0040	32	0x003F FFC0
HRPWM<t>_HRPERBR	High-resolution Period Register B	0x0044	32	0x003F FFC0
HRPWM<t>_HRGCMAR	High-Resolution General Compare Register A	0x0080	32	0x003F FFC0
HRPWM<t>_HRGCMBR	High-Resolution General Compare Register B	0x0084	32	0x003F FFC0
HRPWM<t>_HRGCMCR	High-resolution general compare Register C	0x0088	32	0x003F FFC0
HRPWM<t>_HRGCMDR	High-resolution general compare Register D	0x008C	32	0x003F FFC0
HRPWM<t>_HRGCMER	High-resolution general compare Register E	0x0090	32	0x003F FFC0
HRPWM<t>_HRGCMFR	High-resolution general compare Register F	0x0094	32	0x003F FFC0
HRPWM<t>_HRGCMGR	High-resolution general compare Register G	0x0098	32	0x003F FFC0
HRPWM<t>_HRGCMHR	High-resolution general compare Register H	0x009C	32	0x003F FFC0
HRPWM<t>_SCMAR	Special-compare register A	0x00C0	32	0x003F FFC0
HRPWM<t>_SCMBR	Special-compare register B	0x00C4	32	0x003F FFC0
HRPWM<t>_SCMCR	Special-compare register C	0x00C8	32	0x003F FFC0
HRPWM<t>_SCMDR	Special-compare register D	0x00CC	32	0x003FFFC0
HRPWM<t>_CAPAR	Capture value register A	0x00D0	32	0x003FFFC0
HRPWM<t>_CAPBR	Capture value register B	0x00D4	32	0x003FFFC0
HRPWM<t>_HRDTUAR	High Resolution Dead Time Register UA	0x0100	32	0x003FFFC0
HRPWM<t>_HRDTDAR	High Resolution Dead Time Register DA	0x0104	32	0x003FFFC0
HRPWM<t>_HRDTUBR	High-Resolution Dead Time Register UB	0x0108	32	0x003FFFC0
HRPWM<t>_HRDTDBR	High Resolution Dead Time Register DB	0x010C	32	0x003FFFC0

Register	Description	Offset Address	Bit Width	Reset Value
HRPWM<t>_GCONR	General control register	0x0140	32	0x00000000
HRPWM<t>_ICONR	Interrupt control register	0x0144	32	0x00000000
HRPWM<t>_BCONR1	Shadow Control Register 1	0x0148	32	0x00000000
HRPWM<t>_DCONR	Dead Band Control Register	0x014C	32	0x00000000
HRPWM<t>_PCNAR1	Port Control Register (A1)	0x0154	32	0x00000000
HRPWM<t>_PCNBR1	Port Control Register (B1)	0x0158	32	0x00000000
HRPWM<t>_VPERR	Valid period register	0x0160	32	0x00000000
HRPWM<t>_STFLR1	Status Flag Register 1	0x0164	32	0x80000000
HRPWM<t>_STFLR2	Status Flag Register 2	0x0168	32	0x00000000
HRPWM<t>_HSTAR1	Hardware Start Register 1	0x0180	32	0x00000000
HRPWM<t>_HCLRR1	Hardware clear register 1	0x0188	32	0x00000000
HRPWM<t>_HCPAR1	Hardware Capture Register (A1)	0x0190	32	0x00000000
HRPWM<t>_HCPBR1	Hardware Capture Register (B1)	0x0194	32	0x00000000
HRPWM<t>_HCPAR2	Hardware Capture Register (A2)	0x0198	32	0x00000000
HRPWM<t>_HCPBR2	Hardware Capture Register (B2)	0x019C	32	0x00000000
HRPWM<t>_BCONR2	Shadow Control Register 2	0x0200	32	0x00000000
HRPWM<t>_PCNAR2	Port Control Register (A2)	0x0204	32	0x000AAAAA
HRPWM<t>_PCNBR2	Port Control Register (B2)	0x0208	32	0x000AAAAA
HRPWM<t>_HSTAR2	Hardware Start Register 2	0x0210	32	0x00000000
HRPWM<t>_HCLRR2	Hardware clear Register 2	0x0218	32	0x00000000
HRPWM<t>_EEFOFFSETAR	External Event Filter Offset Register A	0x0220	32	0x0040 0000
HRPWM<t>_EEFOFFSETBR	External Event Filter Offset Register B	0x0224	32	0x0040 0000
HRPWM<t>_EEFWINAR	External Event Filter Window Register A	0x0228	32	0x0040 0000
HRPWM<t>_EEFWINBR	External Event Filter Window Register B	0x022C	32	0x0040 0000
HRPWM<t>_EEFLTCR1	External Event Filter Controller 1	0x0230	32	0x0000 0000
HRPWM<t>_EEFLTCR2	External Event Filter Controller 2	0x0234	32	0x0000 0000
HRPWM<t>_PCNAR3	Port Control Register (A3)	0x0238	32	0x000A AAAA
HRPWM<t>_PCNBR3	Port Control Register (B3)	0x023C	32	0x000A AAAA
HRPWM<t>_IDLECR	Idle Control Register	0x0240	32	0x0000 0000
HRPWM<t>_GCONR1	General Control Register 1	0x0244	32	0x0000 0000
HRPWM<t>_BICONR	Interrupt Control Buffer Register	0x0248	32	0x0000 0000
HRPWM<t>_BPCNAR1	Port Control Buffer Register A1	0x024C	32	0x0000 0000
HRPWM<t>_BPCNBR1	Port Control Buffer Register B1	0x0250	32	0x0000 0000
HRPWM<t>_BPCNAR2	Port Control Buffer Register A2	0x0254	32	0x000A AAAA
HRPWM<t>_BPCNBR2	Port Control Buffer Register B2	0x0258	32	0x000A AAAA
HRPWM<t>_BPCNAR3	Port Control Buffer Register A3	0x025C	32	0x000A AAAA
HRPWM<t>_BPCNBR3	Port Control Buffer Register B3	0x0260	32	0x000A AAAA

Register	Description	Offset Address	Bit Width	Reset Value
HRPWM<t>_BGCONR1	General Control Buffer Register 1	0x0264	32	0x0000 0000
HRPWM<t>_SCMASELR	Special-Compare Match Event A Select Register	0x0268	32	0x0000 0000
HRPWM<t>_CR	High Resolution Control Registers	0x02A0	32	0x0000 0000
HRPWM<t>_PHSCTL	Phase Control Register	0x02A4	32	0x0000 0000
HRPWM1_PHSCMP1A	High-resolution phase Compare Register 1A	0x02B4	32	0x007F FFC0
HRPWM1_PHSCMP1B	High-resolution phase Compare Register 1B	0x02B8	32	0x007F FFC0
HRPWM1_PHSCMP2A	High-resolution phase Compare Register 2A	0x02BC	32	0x007F FFC0
HRPWM1_PHSCMP2B	High-resolution phase Compare Register 2B	0x02C0	32	0x007F FFC0
HRPWM1_PHSCMP3A	High-resolution phase Compare Register 3A	0x02C4	32	0x007F FFC0
HRPWM1_PHSCMP3B	High-resolution phase Compare Register 3B	0x02C8	32	0x007F FFC0
HRPWM1_PHSCMP4A	High-resolution phase Compare Register 4A	0x02CC	32	0x007F FFC0
HRPWM1_PHSCMP4B	High-resolution phase Compare Register 4B	0x02D0	32	0x007F FFC0
HRPWM1_PHSCMP5A	High-resolution phase Compare Register 5A	0x02D4	32	0x007F FFC0
HRPWM1_PHSCMP5B	High-resolution phase Compare Register 5B	0x02D8	32	0x007F FFC0

Table 19-16 HRPWM Common Register List

Register	Description	Offset Address	Bit Width	Reset Value
HRPWM_CALCR	High Resolution Calibration Control Register	0x000C	32	0x0000 0000
HRPWM_SCAPR	Software Synchronous Capture Control Register	0x00AC	32	0x0000 0000
HRPWM_SSTAIDLR	Software Synchronous Idle Control Register	0x00B0	32	0x0000 0000
HRPWM_SSTARUNR1	Software Synchronization Run Control Register 1	0x00B4	32	0x0000 0000
HRPWM_SSTARUNR2	Software Synchronization Run Control Register 2	0x00B8	32	0x0000 0000
HRPWM_SSTADIDLR	Software Synchronization Delayed Idle Control Register	0x00BC	32	0x0000 0000
HRPWM_GCTLR	Global Control Registers	0x00C0	32	0x0000 0000
HRPWM_GBCONR	Global Shadow Control Register	0x00C4	32	0x0000 003F
HRPWM_GBSFLR	Global Shadow Status Flag Register	0x00C8	32	0x0000 0000
HRPWM_BMCR	Burst Mode Control Register	0x00CC	32	0x0000 0000
HRPWM_BMSTRG1	Burst Mode Start Trigger Register 1	0x00D0	32	0x0000 0000
HRPWM_BMSTRG2	Burst Mode Start Trigger Register 2	0x00D4	32	0x0000 0000
HRPWM_BMPERAR	Burst Mode Period Register A	0x00D8	32	0x0000 0000
HRPWM_BMPERBR	Burst Mode Period Register B	0x00DC	32	0x0000 0000
HRPWM_BMCMAR	Burst Mode Compare Register A	0x00E0	32	0x0000 0000
HRPWM_BMCMBR	Burst Mode Compare Register B	0x00E4	32	0x0000 0000
HRPWM_EECR1	External Event Controller 1	0x00E8	32	0x0000 0000
HRPWM_EECR2	External Event Controller 2	0x00EC	32	0x0000 0000
HRPWM_EECR3	External Event Controller 3	0x00F0	32	0x0000 0000
HRPWM_SYNOCR	Synchronous Output Control Register	0x00F4	32	0x0000 0000
HRPWM_EEDSELR	External Event Detection Selection Register	0x00F8	32	0x0000 0000
HRPWM_FCNTR	Filter counter register	0x00FC	32	0x0000 0000
HRPWM_SSTAR	Software Synchronous Start Control Register	0x0100	32	0x0000 0000
HRPWM_SSTPR	Software Synchro Stop Control Register	0x0104	32	0x0000 0000
HRPWM_SCLRR	Software Synchronous Clear Control Register	0x0108	32	0x0000 0000
HRPWM_SUPDR	Software synchronous update control register	0x010C	32	0x0000 0000

19.6.1 HRPWM Unit 1~6 Registers

19.6.1.1 Count Value Register (HRPWM<t>_CNTER)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res										CNT[21:16]					
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
CNT[15:0]															

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b22	Res	Reserved Bit	Read as "0"	R
b21~b6	CNT[21:6]	Counter value	Current timer Value	RW
b5~b0	CNT[5:0]	Counter value	Read as "0"	R

19.6.1.2 Update Register (HRPWM<t>_UPDAR)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res										UPDA[21:16]					
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
UPDA[15:0]															

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b22	Res	Reserved Bit	Read as "0"	R
b21~b6	UPDA[21:6]	Update value	Set the update value to be updated into the timer	RW
b5~b0	UPDA[5:0]	Update value	Read as "0"	R

19.6.1.3 High-resolution period Register (HRPWM<t>_HRPERmR, m=A~B)

Reset value: 0x003F FFC0

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res										HRPERA-B[21:16]					
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
HRPERA-B[15:0]															

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b22	Res	Reserved Bit	Read 0 when reading, write 0 when writing	RW
b21~b0	HRPERA-B	High resolution period value	Set the counting period value and corresponding Shadow value of each round of counting The value of HRPWM_HRPERmR must be greater than 0xC0, and the maximum value must not be greater than 0x3FFFC0	RW

The value of the high-resolution period register is calculated as follows:

Among them, fPWM is the frequency of the output PWM waveform, and fhigh_res is the resolution of the counter (refer to the [Counter Clock] section for details)

19.6.1.4 High-resolution general compare Register (HRPWM<t>_HRGCMmR, m=A~H)

Reset value: 0x003F FFC0

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res										HRGCMA-H [21:16]					
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
HRGCMA-H [15:0]															

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b22	Res	Reserved Bit	Read 0 when reading, write 0 when writing	RW
b21~b0	HRGCMA-H	High-resolution compare A~H value	Comparison reference value setting, the signal matching the count value is valid The value of HRPWM_HRGCMmR must be greater than 0xC0, and the maximum value must not be greater than 0x3FFFC0	RW

19.6.1.5 Special-compare Register (HRPWM<t>_SCMmR, m=A~D)

Reset value: 0x003F FFC0

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res										SCMA~D[21:16]					
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
SCMA~D[15:0]															

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b22	Res	Reserved Bit	Read as "0"	R
b21~b6	SCMA~D[21:6]	Special-Compare value	Set comparison reference value and Shadow value	RW
b5~b0	SCMA~D[5:0]	Special-Compare value	Read as "0" This bit field does not support writing	R

19.6.1.6 Capture Value Register (HRPWM<t>_CAPmR, m=A~B)

Reset value: 0x003F FFC0

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res									CAPDIRA-B	CAPA-B [21:16]					
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
CAPA-B[15:6]										CAPA-B[5:0]					

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b23	Res	Reserved Bit	Read as "0"	RW
b22	CAPDIRA-B	Capture Direction	0: Capture is in the up counting range 1: Capture is in the down counting interval	R
b21~b6	CAPA-B [21:6]	Capturing values	Capturing Values	RW
b5~b0	CAPA-B [5:0]	Capturing values	Write "0" when writing, and "0" when reading	RW

19.6.1.7 High-Resolution Dead Time Register (HRPWM<t>_HRDTU/DmR, m=A~B)

Reset value: 0x003F FFC0

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res										HRDTU/DA-B [21:16]					
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
HRDTU/DA-B [15:0]															

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b22	Res	Reserved Bit	Read 0 when reading, write 0 when writing	RW
b21~b0	HRDTU/DA~B	Dead time value	Dead Time Setting and Shadow Value The value of HRPWM_HRDTUmR must be greater than 0xC0, and the maximum value must not be greater than 0x3FFFC0	RW

19.6.1.8 Interrupt Control Register (HRPWM<t>_ICONR)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24
Res							
b23	b22	b21	b20	b19	b18	b17	b16
Res				INTENSBD	INTENSBU	INTENSAD	INTENSAU
b15	b14	b13	b12	b11	b10	b9	b8
INTENCAPB	INTENCAPA	Res					
b7	b6	b5	b4	b3	b2	b1	b0
NTENUDF	INTENOVF	INTENF	INTENE	INTEND	INTENC	INTENB	INTENA

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b20	Res	Reserved Bit	Read as "0", write as "0"	RW
b19	INTENSBD	SCMBR down count interrupt enable B	0: During down counting, when the SCMBR register is equal to the count value, the interrupt is invalid 1: During down counting, when the SCMBR register is equal to the count value, the interrupt is enabled	RW
b18	INTENSBU	SCMBR up-counting interrupt enable B	0: During up-counting, when the SCMBR register is equal to the count value, the interrupt is invalid 1: During up-counting, when the SCMBR register is equal to the count value, the interrupt is enabled	RW
b17	INTENSAD	SCMAR down count interrupt enable A	0: During down counting, when the SCMAR register is equal to the count value, the interrupt is invalid 1: During down counting, when the SCMAR register is equal to the count value, the interrupt is enabled	RW
b16	INTENSAU	SCMAR up-counting interrupt enable A	0: During up-counting, when the SCMAR register is equal to the count value, the interrupt is invalid 1: During up-counting, when the SCMAR register is equal to the count value, the interrupt is enabled	RW
b15	INTENCAPB	Capture interrupt enable B	0: When capture event B occurs, the interrupt is invalid 1: When capture event B occurs, this interrupt is enabled	RW
b14	INTENCAPA	Capture interrupt enable A	0: When capture event A occurs, the interrupt is invalid 1: When capture event A occurs, this interrupt is enabled	RW
b13~b8	Res	Reserved Bit	Read as "0", write as "0"	RW
b7	INTENUDF	Zero match interrupt enable	0: When the count value is equal to 0, the interrupt is invalid 1: When the count value is equal to 0, the interrupt is enabled	RW
b6	INTENOVF	Period match interrupt enable	0: When the count value is equal to the period, the interrupt is invalid 1: When the count value is equal to the period, the interrupt is enabled	RW
b5	INTENF	Count match interrupt enable F	0: When the GCMFR register is equal to the count value, the interrupt is invalid 1: When the GCMFR register is equal to the count value, the interrupt is enabled	RW
b4	INTENE	Count match interrupt enable E	0: When the GCMER register is equal to the count value, the interrupt is invalid 1: When the GCMER register is equal to the count value, the interrupt is enabled	RW
b3	INTEND	Count match interrupt enable D	0: When the GCMDR register is equal to the count value, the interrupt is invalid 1: When the GCMDR register is equal to the count value, the interrupt is enabled	RW
b2	INTENC	Count match interrupt enable C	0: When the GCMCR register is equal to the count value, the interrupt is invalid 1: When the GCMCR register is equal to the count value, the interrupt is enabled	RW
b1	INTENB	Count match interrupt enable B	0: When the GCMBR register is equal to the count value, the interrupt is invalid 1: When the GCMBR register is equal to the count value, the interrupt is enabled	RW
b0	INTENA	Count Match interrupt enable A	0: When the GCMAR register is equal to the count value, the interrupt is invalid 1: When the GCMAR register is equal to the count value, the interrupt is enabled	RW

19.6.1.9 Interrupt Control Shadow Register (HRPWM<t>_BICONR)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24
Reserved							

b23	b22	b21	b20	b19	b18	b17	b16
Reserved				INTENSBD	INTENSBU	INTENSAD	INTENSAU

b15	b14	b13	b12	b11	b10	b9	b8
INTENCAPB	INTENCAPA	Reserved					

b7	b6	b5	b4	b3	b2	b1	b0
NTENUDF	INTENOVF	INTENF	INTENE	INTEND	INTENC	INTENB	INTENA

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b20	Res	Reserved Bit	Read as "0", write as "0"	RW
b19	INTENSBD	SCMBR down count interrupt enable B	0: During down counting, when the SCMBR register is equal to the count value, the interrupt is invalid 1: During down counting, when the SCMBR register is equal to the count value, the interrupt is enabled	RW
b18	INTENSBU	SCMBR up-counting interrupt enable B	0: During up-counting, when the SCMBR register is equal to the count value, the interrupt is invalid 1: During up-counting, when the SCMBR register is equal to the count value, the interrupt is enabled	RW
b17	INTENSAD	SCMAR down count interrupt enable A	0: During down counting, when the SCMAR register is equal to the count value, the interrupt is invalid 1: During down counting, when the SCMAR register is equal to the count value, the interrupt is enabled	RW
b16	INTENSAU	SCMAR up-counting interrupt enable A	0: During up-counting, when the SCMAR register is equal to the count value, the interrupt is invalid 1: During up-counting, when the SCMAR register is equal to the count value, the interrupt is enabled	RW
b15	INTENCAPB	Capture interrupt enable B	0: When capture event B occurs, the interrupt is invalid 1: When capture event B occurs, this interrupt is enabled	RW
b14	INTENCAPA	Capture interrupt enable A	0: When capture event A occurs, the interrupt is invalid 1: When capture event A occurs, this interrupt is enabled	RW
b13~b8	Res	Reserved Bit	Read as "0", write as "0"	RW
b7	INTENUDF	Zero match interrupt enable	0: When the counter counts to zero, the interrupt is invalid. 1: When the counter counts to zero, the interrupt is enabled	RW
b6	INTENOVF	Period match interrupt enable	0: When the counter count cycle matches, the interrupt is invalid 1: When the counter count period matches, the interrupt is enabled	RW
b5	INTENF	Count match interrupt enable F	0: When the GCMFR register is equal to the count value, the interrupt is invalid 1: When the GCMFR register is equal to the count value, the interrupt is enabled	RW
b4	INTENE	Count match interrupt enable E	0: When the GCMER register is equal to the count value, the interrupt is invalid 1: When the GCMER register is equal to the count value, the interrupt is enabled	RW
b3	INTEND	Count match interrupt enable D	0: When the GCMDR register is equal to the count value, the interrupt is invalid 1: When the GCMDR register is equal to the count value, the interrupt is enabled	RW
b2	INTENC	Count match interrupt enable C	0: When the GCMCR register is equal to the count value, the interrupt is invalid 1: When the GCMCR register is equal to the count value, the interrupt is enabled	RW
b1	INTENB	Count match interrupt enable B	0: When the GCMBR register is equal to the count value, the interrupt is invalid 1: When the GCMBR register is equal to the count value, the interrupt is enabled	RW
b0	INTENA	Count Match interrupt enable A	0: When the GCMAR register is equal to the count value, the interrupt is invalid 1: When the GCMAR register is equal to the count value, the interrupt is enabled	RW

19.6.1.10 Dead Time Control Register (HRPWM<t>_DCONR)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res								DTB TRD	DTB TRU	DTB END	DTB ENU	Res		SEPA	DTC EN

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b8	Res	Reserved Bit	Read as "0", write as "0"	RW
b7	DTBTRD	Dead time D value Shadow transfer time setting	0: When the zero point **Shadow Transfer Time** is transferred, the Shadow value is not transferred 1: When the zero point **Shadow Transfer Time** is reached, a Shadow value transfer occurs	RW
b6	DTBTRU	Dead time U value buffer transfer time setting	0: When the period **Shadow Transfer Time** is used, the Shadow value is not transferred 1: When the period **Shadow Transfer Time** is reached, a Shadow value transfer occurs	RW
b5	DTBEND	Dead Time D Value Shadow function enable selection	0: Buffer transmission disable 1: Buffer transfer enable (HRDTDBR->HRDTDAR)	RW
b4	DTBENU	Dead time U value Shadow transfer enable selection	0: Buffer transmission is disable 1: Shadow transfer enable (HRDTUBR->HRDTUAR)	RW
b3~b2	Res	Reserved Bit	Read as "0", write as "0"	RW
b1	SEPA	Separate settings	0: HRDTUAR and HRDTDAR are set separately 1: The value of HRDTUAR is automatically equal to the value of HRDTDAR	RW
B0	DTCEN	Dead zone function Enable	0: Dead zone function is invalid 1: The dead zone function is valid	RW

19.6.1.11 Effective Period Register (HRPWM<t>_VPERR)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res									PCNTS[4:0]					PCNTE[1:0]	
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res						SP PERIB	SP PERIA	Res							

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b23	Res	Reserved Bit	Read as "0", write as "0"	RW
b22~b18	PCNTS[4:0]	Valid period selection	00000: Valid cycle selection function is invalid 00001: Valid every 1 cycle 00010: Valid every 2 cycles 00011: Valid every 3 cycles 00100: Valid every 4 cycles 00101: Valid every 5 cycles 00110: Valid every 6 cycles 00111: Valid every 7 cycles 01000: Valid every 8 cycles 01001: Valid every 9 cycles 01010: Valid every 10 cycles 01011: Valid every 11 cycles 01100: Valid every 12 cycles 01101: Valid every 13 cycles 01110: Valid every 14 cycles 01111: Valid every 15 cycles ..... 11110: Valid every 30 cycles 11111: Valid every 31 cycles	RW
b17~b16	PCNTE[1:0]	Active period count event selection	00: Valid period selection function is invalid 01: The sawtooth count mode value is equal to the period value or hardware clear in sawtooth count mode or the triangle count mode value is equal to 0 as the counting condition 10: Sawtooth count mode value equals period value or hardware clear in sawtooth counter modes or triangle count mode value equals period value as counting condition 11: Sawtooth count mode value equals period value or hardware clear in sawtooth counter modes or triangle count mode value equals period value, count value equals 0 as counting condition	RW
b15~b10	Res	Reserved Bit	Read as "0", write as "0"	RW
b9	SPPERIB	Special effective period selection B	0: Valid cycle selection function is invalid 1: Valid cycle selection function is enabled	RW
b8	SPPERIA	Special signal effective period selection A	0: Valid cycle selection function is invalid 1: Valid cycle selection function is enabled	RW
b7~b0	Res	Reserved Bit	Read as "0", write as "0"	RW

19.6.1.12 Status Flag Register (HRPWM<t>_STFLR1)

Reset value: 0x8000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
DIRF	Res			CAPBF	CAPAF	VPERNUM[4:0]					Res				
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res			CMSBDF	CMSBUF	CMSADF	CMSAUF	Res	UDFF	OVFF	CMFF	CMEF	CMDF	CMCF	CMBF	CMAF

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31	DIRF	Counting Direction	0: count down 1: count up	R
b30~b28	Res	Reserved Bit	Read as "0", write as "0"	R
b27	CAPBF	Capture B Flag	0: Capture event B did not occur 1: Capture event B occurs Writing 0 to this bit clears it to 0, and writing 1 is invalid.	RW
b26	CAPAF	Capture A Flag	0: Capture event A did not occur 1: Capture event B occurs Writing 0 to this bit clears it to 0, and writing 1 is invalid.	RW
b25~b21	VPERNUM[4:0]	Number of cycles	When the effective cycle selection function is enabled, the number of cycles after counting	R
b20~b13	Res	Reserved Bit	Read as "0", write as "0"	R
b12	CMSBDF	SCMBR Count down flag	0: When counting down, the value of the SCMBR register is not equal to the count value 1: When counting down, the value of the SCMBR register is equal to the count value Writing 0 to this bit clears it to 0, and writing 1 is invalid.	RW
b11	CMSBUF	SCMBR Count up flag	0: When counting up, the value of the SCMBR register is not equal to the count value 1: When counting up, the value of the SCMBR register is equal to the count value Writing 0 to this bit clears it to 0, and writing 1 is invalid.	RW
b10	CMSADF	SCMAR Count down flag	0: When counting down, the value of the SCMAR register is not equal to the count value 1: When counting down, the value of the SCMAR register is equal to the count value Writing 0 to this bit clears it to 0, and writing 1 is invalid.	RW
b09	CMSAUF	SCMAR Count up flag	0: When counting up, the value of the SCMAR register is not equal to the count value 1: When counting up, the value of the SCMAR register is equal to the count value Writing 0 to this bit clears it to 0, and writing 1 is invalid.	RW
b8	Res	Reserved Bit	Read as "0", write as "0"	R
b7	UDFF	Zero flag	0: No zero matching occurred 1: Zero match occurs Writing 0 to this bit clears it to 0, and writing 1 is invalid.	RW
b6	OVFF	Period flag	0: No cycle match occurred 1: Cycle matching occurs Writing 0 to this bit clears it to 0, and writing 1 is invalid.	RW
b5	CMFF	Count F flag	0: The value of the GCMFR register is not equal to the count value 1: The value of the GCMFR register is equal to the count value Writing 0 to this bit clears it to 0, and writing 1 is invalid.	RW
b4	CMEF	count E flag	0: The value of the GCMER register is not equal to the count value 1: The value of the GCMER register is equal to the count value Writing 0 to this bit clears it to 0, and writing 1 is invalid.	RW
b3	CMDF	Count D flag	0: The value of the GCMDR register is not equal to the count value 1: The value of the GCMDR register is equal to the count value Writing 0 to this bit clears it to 0, and writing 1 is invalid.	RW
b2	CMCF	Count C flag	0: The value of the GCMCR register is not equal to the count value 1: The value of the GCMCR register is equal to the count value Writing 0 to this bit clears it to 0, and writing 1 is invalid.	RW

b1	CMBF	Count B flag	0: The value of the GCMBR register is not equal to the count value 1: The value of the GCMBR register is equal to the count value Writing 0 to this bit clears it to 0, and writing 1 is invalid.	RW
b0	CMAF	Count A flag	0: The value of the GCMAR register is not equal to the count value 1: The value of the GCMAR register is equal to the count value Writing 0 to this bit clears it to 0, and writing 1 is invalid.	RW

19.6.1.13 Status Flag Register (HRPWM<t>_STFLR2)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															OSTOVFF
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
PHSCMP5F	PHSCMP4F	PHSCMP3F	PHSCMP2F	PHSCMP1F	Res					OBSTAT	OASTAT	Res	DLYPRT	DLYIDLEB	DLYIDLEA

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b17	Res	Reserved Bit	Read as "0", write as "0"	R
b16	OSTOVFF	Single-shot mode period flag	0: In single counting mode, the single counting completion point does not occur 1: In single counting mode, the single counting completion point occurs The single count completion point means that the Sawtooth count mode value is equal to the period value and the triangle count modes down and the count value is equal to 1. Writing 0 to this bit clears it to 0, and writing 1 is invalid.	RW
b15	PHSCMP5F	Phase 5 flag	0: Phase matching event 5 did not occur 1: Phase matching event 5 occurs Only supports unit 1 Writing 0 to this bit clears it to 0, and writing 1 is invalid.	RW
b14	PHSCMP4F	Phase 4 flag	0: Phase matching event 4 did not occur 1: Phase matching event 4 occurs Only supports unit 1 Writing 0 to this bit clears it to 0, and writing 1 is invalid.	RW
b13	PHSCMP3F	Phase 3 flag	0: Phase matching event 3 did not occur 1: Phase matching event 3 occurs Only supports unit 1 Writing 0 to this bit clears it to 0, and writing 1 is invalid.	RW
b12	PHSCMP2F	Phase 2 flag	0: Phase matching event 2 did not occur 1: Phase matching event 2 occurs Only supports unit 1 Writing 0 to this bit clears it to 0, and writing 1 is invalid.	RW
b11	PHSCMP1F	Phase 1 flag	0: Phase matching event 1 did not occur 1: Phase matching event 1 occurs Only supports unit 1 Writing 0 to this bit clears it to 0, and writing 1 is invalid.	RW
b10~b6	Res	Reserved Bit	Read as "0", write as "0"	R
b5	OBSTAT	The state of PWMB when entering delayed idle	0: Low level when entering 1: High level when entering	R
b4	OASTAT	The state of PWMA when entering delayed idle	0: Low level when entering 1: High level when entering	R
b3	Res	Reserved Bit	Read as "0", write as "0"	R
b2	DLYPRT	Delayed idle state	0: Delayed idle entry does not occur 1: Delay idle entry occurrence After reading 1, write 0 to this bit to clear it to 0, writing 1 invalid	RW
b1	DLYIDLEB	PWMB is in delayed idle state	0: Output B is not in delayed idle state 1: Output B is in delayed idle state	R
b0	DLYIDLEA	PWMA is in delayed idle state	0: Output A is not in delayed idle state 1: Output A is in delayed idle state	R

19.6.1.14 Hardware Start Register 1 (HRPWM<t>_HSTAR1)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res								HSTA 23	HSTA 22	HSTA 21	HSTA 20	HSTA 19	HSTA 18	HSTA 17	HSTA 16

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res				HSTA 11	HSTA 10	HSTA 9	HSTA 8	STAS	Res						

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b24	Res	Reserved Bit	Read as "0", write as "0"	RW
b23	HSTA23	Hardware startup condition 23	Condition: HRPWM_TRIGD port is sampled to the falling edge 0: Hardware startup is invalid when conditions match 1: Hardware startup is valid when conditions match	RW
b22	HSTA22	Hardware startup condition 22	Condition: HRPWM_TRIGD port is sampled to the rising edge 0: Hardware startup is invalid when conditions match 1: Hardware startup is valid when conditions match	RW
b21	HSTA21	Hardware startup condition 21	Condition: HRPWM_TRIGC port is sampled to the falling edge 0: Hardware startup is invalid when conditions match 1: Hardware startup is valid when conditions match	RW
b20	HSTA20	Hardware startup condition 20	Condition: HRPWM_TRIGC port is sampled to the rising edge 0: Hardware startup is invalid when conditions match 1: Hardware startup is valid when conditions match	RW
b19	HSTA19	Hardware startup condition 19	Condition: HRPWM_TRIGB port is sampled to the falling edge 0: Hardware startup is invalid when conditions match 1: Hardware startup is valid when conditions match	RW
b18	HSTA18	Hardware startup condition 18	Condition: HRPWM_TRIGB port is sampled to the rising edge 0: Hardware startup is invalid when conditions match 1: Hardware startup is valid when conditions match	RW
b17	HSTA17	Hardware startup condition 17	Condition: HRPWM_TRIGA port is sampled to the falling edge 0: Hardware startup is invalid when conditions match 1: Hardware startup is valid when conditions match	RW
b16	HSTA16	Hardware startup condition 16	Condition: HRPWM_TRIGA port is sampled to the rising edge 0: Hardware startup is invalid when conditions match 1: Hardware startup is valid when conditions match	RW
b15~b12	Res	Reserved Bit	Read as "0", write as "0"	RW
b11	HSTA11	Hardware startup condition 11	Condition: Internal Hardware Trigger Event 3 Valid 0: Hardware startup is invalid when conditions match 1: Hardware startup is valid when conditions match	RW
b10	HSTA10	Hardware startup condition 10	Condition: Internal Hardware Trigger Event 2 Valid 0: Hardware startup is invalid when conditions match 1: Hardware startup is valid when conditions match	RW
b9	HSTA9	Hardware startup condition 9	Condition: Internal Hardware Trigger Event 1Valid 0: Hardware startup is invalid when conditions match 1: Hardware startup is valid when conditions match	RW
b8	HSTA8	Hardware startup condition 8	Condition: Internal Hardware Trigger Event 0Valid 0: Hardware startup is invalid when conditions match 1: Hardware startup is valid when conditions match	RW
b7	STAS	Hardware enable	0: hardware startup is invalid 1: Hardware startup is valid Note: When the hardware boot is valid, the setting of SSTAR is invalid	RW
b6~b0	Res	Reserved Bit	Read as "0", write as "0"	RW

19.6.1.15 Hardware Clear Register 1 (HRPWM<t>_HCLRR1)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res								HCLE 23	HCLE 22	HCLE 21	HCLE 20	HCLE 19	HCLE 18	HCLE 17	HCLE 16

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res				HCLE 11	HCLE 10	HCLE 9	HCLE 8	CLES	Res						

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b24	Res	Reserved Bit	Read as "0", write as "0"	RW
b23	HCLE23	Hardware Clear condition 23	Condition: HRPWM_TRIGD port is sampled to the falling edge 0: Hardware Clearis invalid when conditions match 1: Hardware Clearis valid when conditions match	RW
b22	HCLE22	Hardware Clear condition 22	Condition: HRPWM_TRIGD port is sampled to the rising edge 0: Hardware Clearis invalid when conditions match 1: Hardware Clearis valid when conditions match	RW
b21	HCLE21	Hardware Clear condition 21	Condition: HRPWM_TRIGC port is sampled to the falling edge 0: Hardware Clear is invalid when conditions match 1: Hardware Clear is valid when conditions match	RW
b20	HCLE20	Hardware Clear condition 20	Condition: HRPWM_TRIGC port is sampled to the rising edge 0: Hardware Clear is invalid when conditions match 1: Hardware Clear is valid when conditions match	RW
b19	HCLE19	Hardware Clear condition 19	Condition: HRPWM_TRIGB port is sampled to the falling edge 0: Hardware Clear is invalid when conditions match 1: Hardware Clear is valid when conditions match	RW
b18	HCLE18	Hardware Clear condition 18	Condition: HRPWM_TRIGB port is sampled to the rising edge 0: Hardware Clear is invalid when conditions match 1: Hardware Clear is valid when conditions match	RW
b17	HCLE17	Hardware Clear condition 17	Condition: HRPWM_TRIGA port is sampled to the falling edge 0: Hardware Clear is invalid when conditions match 1: Hardware Clear is valid when conditions match	RW
b16	HCLE16	Hardware Clear condition 16	Condition: HRPWM_TRIGA port is sampled to the rising edge 0: Hardware Clear is invalid when conditions match 1: Hardware Clear is valid when conditions match	RW
b15~b12	Res	-	Read as "0", write as "0"	RW
b11	HCLE11	Hardware Clear condition 11	Condition: Internal Hardware Trigger Event 3 Valid 0: Hardware Clear is invalid when conditions match 1: Hardware Clear is valid when conditions match	RW
b10	HCLE10	Hardware Clear condition 10	Condition: Internal Hardware Trigger Event 2 Valid 0: Hardware Clear is invalid when conditions match 1: Hardware Clear is valid when conditions match	RW
b9	HCLE9	Hardware Clear condition 9	Condition: Internal Hardware Trigger Event 1Valid 0: Hardware Clear is invalid when conditions match 1: Hardware Clear is valid when conditions match	RW
b8	HCLE8	Hardware Clear condition 8	Condition: Internal Hardware Trigger Event0Valid 0: Hardware Clear is invalid when conditions match 1: Hardware Clear is valid when conditions match	RW
b7	CLES	Hardware clear enable	0: Hardware reset is invalid 1: Hardware clear is valid Note: When the hardware clear is valid, the setting of SCLRR is invalid	RW
b6~b0	Res	-	Read as "0", write as "0"	RW

19.6.1.16 Hardware Capture A Register 1 (HRPWM<t>_HCPAR1)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res								HCPA23	HCPA22	HCPA21	HCPA20	HCPA19	HCPA18	HCPA17	HCPA16

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res				HCPA11	HCPA10	HCPA9	HCPA8	Res							

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b24	Res	Reserved Bit	Read as "0", write as "0"	RW
b23	HCPA23	Hardware Capture A Condition 23	Condition: HRPWM_TRIGD port is sampled to the falling edge 0: When the condition is matched, the hardware capture A is invalid 1: When the condition is matched, the hardware capture A is valid	RW
b22	HCPA22	Hardware Capture A Condition 22	Condition: HRPWM_TRIGD port is sampled to the rising edge 0: When the condition is matched, the hardware capture A is invalid 1: When the condition is matched, the hardware capture A is valid	RW
b21	HCPA21	Hardware Capture A Condition 21	Condition: HRPWM_TRIGC port is sampled to the falling edge 0: When the condition is matched, the hardware capture A is invalid 1: When the condition is matched, the hardware capture A is valid	RW
b20	HCPA20	Hardware Capture A Condition 20	Condition: HRPWM_TRIGC port is sampled to the rising edge 0: When the condition is matched, the hardware capture A is invalid 1: When the condition is matched, the hardware capture A is valid	RW
b19	HCPA19	Hardware Capture A Condition 19	Condition: HRPWM_TRIGB port is sampled to the falling edge 0: When the condition is matched, the hardware capture A is invalid 1: When the condition is matched, the hardware capture A is valid	RW
b18	HCPA18	Hardware Capture A Condition 18	Condition: HRPWM_TRIGB port is sampled to the rising edge 0: When the condition is matched, the hardware capture A is invalid 1: When the condition is matched, the hardware capture A is valid	RW
b17	HCPA17	Hardware Capture A Condition 17	Condition: HRPWM_TRIGA port is sampled to the falling edge 0: When the condition is matched, the hardware capture A is invalid 1: When the condition is matched, the hardware capture A is valid	RW
b16	HCPA16	Hardware Capture A Condition 16	Condition: HRPWM_TRIGA port is sampled to the rising edge 0: When the condition is matched, the hardware capture A is invalid 1: When the condition is matched, the hardware capture A is valid	RW
b15~b12	Res	Reserved Bit	Read as "0", write as "0"	RW
b11	HCPA11	Hardware Capture A Condition 11	Condition: Internal Hardware Trigger Event 3 Valid 0: When the condition is matched, the hardware capture A is invalid 1: When the condition is matched, the hardware capture A is valid	RW
b10	HCPA10	Hardware Capture A Condition 10	Condition: Internal Hardware Trigger Event 2 Valid 0: When the condition is matched, the hardware capture A is invalid 1: When the condition is matched, the hardware capture A is valid	RW

b9	HCPA9	Hardware Capture A Condition 9	Condition: Internal Hardware Trigger Event 1 Valid 0: When the condition is matched, the hardware capture A is invalid 1: When the condition is matched, the hardware capture A is valid	RW
b8	HCPA8	Hardware Capture A Condition 8	Condition: Internal Hardware Trigger Event 0 Valid 0: When the condition is matched, the hardware capture A is invalid 1: When the condition is matched, the hardware capture A is valid	RW
b7~b0	Res	Reserved Bit	Read as "0", write as "0"	RW

19.6.1.17 Hardware Capture B Register 1 (HRPWM<t>_HCPBR1)

Reset Value: 0x00000000h

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res								HCPB23	HCPB22	HCPB21	HCPB20	HCPB19	HCPB18	HCPB17	HCPB16

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res				HCPB11	HCPB10	HCPB9	HCPB8	Res							

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b24	Res	Reserved Bit	Read as "0", write as "0"	RW
b23	HCPB23	Hardware Capture B Condition 23	Condition: HRPWM_TRIGD port is sampled to the falling edge 0: When the condition is matched, the hardware capture B is invalid 1: When the condition is matched, the hardware capture B is valid	RW
b22	HCPB22	Hardware Capture B Condition22	Condition: HRPWM_TRIGD port is sampled to the rising edge 0: When the condition is matched, the hardware capture B is invalid 1: When the condition is matched, the hardware capture B is valid	RW
b21	HCPB21	Hardware Capture B Condition21	Condition: HRPWM_TRIGC port is sampled to the falling edge 0: When the condition is matched, the hardware capture B is invalid 1: When the condition is matched, the hardware capture B is valid	RW
b20	HCPB20	Hardware Capture B Condition20	Condition: HRPWM_TRIGC port is sampled to the rising edge 0: When the condition is matched, the hardware capture B is invalid 1: When the condition is matched, the hardware capture B is valid	RW
b19	HCPB19	Hardware Capture B Condition19	Condition: HRPWM_TRIGB port is sampled to the falling edge 0: When the condition is matched, the hardware capture B is invalid 1: When the condition is matched, the hardware capture B is valid	RW
b18	HCPB18	Hardware Capture B Condition18	Condition: HRPWM_TRIGB port is sampled to the rising edge 0: When the condition is matched, the hardware capture B is invalid 1: When the condition is matched, the hardware capture B is valid	RW
b17	HCPB17	Hardware Capture B Condition17	Condition: HRPWM_TRIGA port is sampled to the falling edge 0: When the condition is matched, the hardware capture B is invalid 1: When the condition is matched, the hardware capture B is valid	RW
b16	HCPB16	Hardware Capture B Condition16	Condition: HRPWM_TRIGA port is sampled to the rising edge 0: When the condition is matched, the hardware capture B is invalid 1: When the condition is matched, the hardware capture B is valid	RW
b15~b12	Res	Reserved Bit	Read as "0", write as "0"	RW
b11	HCPB11	Hardware Capture B Condition 11	Condition: Internal Hardware Trigger Event 3 Valid 0: When the condition is matched, the hardware capture B is invalid 1: When the condition is matched, the hardware capture B is valid	RW
b10	HCPB10	Hardware Capture B Condition10	Condition: Internal Hardware Trigger Event 2 Valid 0: When the condition is matched, the hardware capture B is invalid 1: When the condition is matched, the hardware capture B is valid	RW

b9	HCPB9	Hardware Capture B Condition9	Condition: Internal Hardware Trigger Event 1 Valid 0: When the condition is matched, the hardware capture B is invalid 1: When the condition is matched, the hardware capture B is valid	RW
b8	HCPB8	Hardware Capture B Condition8	Condition: Internal Hardware Trigger Event 0 Valid 0: When the condition is matched, the hardware capture B is invalid 1: When the condition is matched, the hardware capture B is valid	RW
b7~b0	Res	Reserved Bit	Read as "0", write as "0"	RW

19.6.1.18 Hardware Capture A Register 2 (HRPWM<t>_HCPAR2)

Reset Value: 0x00000000h

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res					HCPAEN	HCPAEEV10	HCPAEEV9	HCPAEEV8	HCPAEEV7	HCPAEEV6	HCPAEEV5	HCPAEEV4	HCPAEEV3	HCPAEEV2	HCPAEEV1

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b11	Res	Reserved Bit	Read as "0", write as "0"	RW
b10	HCPAEN	Hardware capture A enable	0: Hardware capture A invalid 1: Hardware capture A valid	RW
b9	HCPAEEV10	External event 10 hardware capture A enable	Condition: External event 10 valid 0: When the condition is matched, the hardware capture A is invalid 1: When the condition is matched, the hardware capture A is valid	RW
b8	HCPAEEV9	External event 9 hardware capture A enable	Condition: External event 9 valid 0: When the condition is matched, the hardware capture A is invalid 1: When the condition is matched, the hardware capture A is valid	RW
b7	HCPAEEV8	External event 8 hardware capture A enable	Condition: External event 8 valid 0: When the condition is matched, the hardware capture A is invalid 1: When the condition is matched, the hardware capture A is valid	RW
b6	HCPAEEV7	External event 7 hardware capture A enable	Condition: External event 7 valid 0: When the condition is matched, the hardware capture A is invalid 1: When the condition is matched, the hardware capture A is valid	RW
b5	HCPAEEV6	External event 6 hardware capture A enable	Condition: External event 6 valid 0: When the condition is matched, the hardware capture A is invalid 1: When the condition is matched, the hardware capture A is valid	RW
b4	HCPAEEV5	External event 4 hardware capture A enable	Condition: External event 5 valid 0: When the condition is matched, the hardware capture A is invalid 1: When the condition is matched, the hardware capture A is valid	RW
b3	HCPAEEV4	External event 4 hardware capture A enable	Condition: External event 4 valid 0: When the condition is matched, the hardware capture A is invalid 1: When the condition is matched, the hardware capture A is valid	RW
b2	HCPAEEV3	External event 3 hardware capture A enable	Condition: External event 3 valid 0: When the condition is matched, the hardware capture A is invalid 1: When the condition is matched, the hardware capture A is valid	RW
b1	HCPAEEV2	External event 2 hardware capture A enable	Condition: External event 2 valid 0: When the condition is matched, the hardware capture A is invalid 1: When the condition is matched, the hardware capture A is valid	RW
b0	HCPAEEV1	External event 1 hardware capture A enable	Condition: External event 1 valid 0: When the condition is matched, the hardware capture A is invalid 1: When the condition is matched, the hardware capture A is valid	RW

19.6.1.19 Hardware Capture B Register 2 (HRPWM<t>_HCPBR2)

Reset Value: 0x00000000h

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res					HCPBEN	HCPBEEV10	HCPBEEV9	HCPBEEV8	HCPBEEV7	HCPBEEV6	HCPBEEV5	HCPBEEV4	HCPBEEV3	HCPBEEV2	HCPBEEV1

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b11	Res	Reserved Bit	Read as "0", write as "0"	RW
b10	HCPBEN	Hardware capture B enable	0: Hardware capture B invalid 1: Hardware capture B valid	RW
b9	HCPBEEV10	External event 10 hardware capture B enable	Condition: External event 10 valid 0: When the condition is matched, the hardware capture B is invalid 1: When the condition is matched, the hardware capture B is valid	RW
b8	HCPBEEV9	External event 9 hardware capture B enable	Condition: External event 9 valid 0: When the condition is matched, the hardware capture B is invalid 1: When the condition is matched, the hardware capture B is valid	RW
b7	HCPBEEV8	External event 8 hardware capture B enable	Condition: External event 8 valid 0: When the condition is matched, the hardware capture B is invalid 1: When the condition is matched, the hardware capture B is valid	RW
b6	HCPBEEV7	External event 7 hardware capture B enable	Condition: External event 7 valid 0: When the condition is matched, the hardware capture B is invalid 1: When the condition is matched, the hardware capture B is valid	RW
b5	HCPBEEV6	External event 6 hardware capture B enable	Condition: External event 6 valid 0: When the condition is matched, the hardware capture B is invalid 1: When the condition is matched, the hardware capture B is valid	RW
b4	HCPBEEV5	External event 4 hardware capture B enable	Condition: External event 5 valid 0: When the condition is matched, the hardware capture B is invalid 1: When the condition is matched, the hardware capture B is valid	RW
b3	HCPBEEV4	External event 4 hardware capture B enable	Condition: External event 4 valid 0: When the condition is matched, the hardware capture B is invalid 1: When the condition is matched, the hardware capture B is valid	RW
b2	HCPBEEV3	External event 3 hardware capture B enable	Condition: External event 3 valid 0: When the condition is matched, the hardware capture B is invalid 1: When the condition is matched, the hardware capture B is valid	RW
b1	HCPBEEV2	External event 2 hardware capture B enable	Condition: External event 2 valid 0: When the condition is matched, the hardware capture B is invalid 1: When the condition is matched, the hardware capture B is valid	RW
b0	HCPBEEV1	External event 1 hardware capture B enable	Condition: External event 1 valid 0: When the condition is matched, the hardware capture B is invalid 1: When the condition is matched, the hardware capture B is valid	RW

19.6.1.20 Hardware Start Register 2 (HRPWM<t>_HSTAR2)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res				HSTASCMB6	HSTASCMB5	HSTASCMB4	HSTASCMB3	HSTASCMB2	HSTASCMB1	HSTAGCMB6	HSTAGCMA6	HSTAGCMB5	HSTAGCMA5	HSTAGCMB4	HSTAGCMA4
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
HSTAGCMB3	HSTAGCMA3	HSTAGCMB2	HSTAGCMA2	HSTAGCMB1	HSTAGCMA1	HSTAEXEV10	HSTAEXEV9	HSTAEXEV8	HSTAEXEV7	HSTAEXEV6	HSTAEXEV5	HSTAEXEV4	HSTAEXEV3	HSTAEXEV2	HSTAEXEV1

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b28	Res	Reserved Bit	Read as "0", write as "0"	RW
b27	HSTASCMB6	Unit 6 SCMPB enable	Condition: Unit 6 SCMPB B valid 0: Hardware startup is invalid when conditions match 1: Hardware startup is valid when conditions match	RW
b26	HSTASCMB5	Unit 5 SCMPB start enable	Condition: Unit 5 SCMPB valid 0: Hardware startup is invalid when conditions match 1: Hardware startup is valid when conditions match	RW
b25	HSTASCMB4	Unit 4 SCMPB start enable	Condition: Unit 4 SCMPB valid 0: Hardware startup is invalid when conditions match 1: Hardware startup is valid when conditions match	RW
b24	HSTASCMB3	Unit 3 SCMPB start enable	Condition: Unit 3 SCMPB valid 0: Hardware startup is invalid when conditions match 1: Hardware startup is valid when conditions match	RW
b23	HSTASCMB2	Unit 2 SCMPB start enable	Condition: Unit 2 SCMPB valid 0: Hardware startup is invalid when conditions match 1: Hardware startup is valid when conditions match	RW
b22	HSTASCMB1	Unit 1 SCMPB start enable	Condition: Unit 1 SCMPB valid 0: Hardware startup is invalid when conditions match 1: Hardware startup is valid when conditions match	RW
b21	HSTAGCMB6	Unit 6 CMP B start enable	Condition: Unit 6 CMP B valid 0: Hardware startup is invalid when conditions match 1: Hardware startup is valid when conditions match	RW
b20	HSTAGCMA6	Unit 6 CMP A start enable	Condition: Unit 6 CMP A valid 0: Hardware startup is invalid when conditions match 1: Hardware startup is valid when conditions match	RW
b19	HSTAGCMB5	Unit 5 CMP B start enable	Condition: Unit 5 CMP B valid 0: Hardware startup is invalid when conditions match 1: Hardware startup is valid when conditions match	RW
b18	HSTAGCMA5	Unit 5 CMP A start enable	Condition: Unit 5 CMP A valid 0: Hardware startup is invalid when conditions match 1: Hardware startup is valid when conditions match	RW
b17	HSTAGCMB4	Unit 4 CMP B start enable	Condition: Unit 4 CMP B valid 0: Hardware startup is invalid when conditions match 1: Hardware startup is valid when conditions match	RW
b16	HSTAGCMA4	Unit 4 CMP A start enable	Condition: Unit 4 CMP A valid 0: Hardware startup is invalid when conditions match 1: Hardware startup is valid when conditions match	RW
b15	HSTAGCMB3	Unit 3 CMP B start enable	Condition: Unit 3 CMP B valid 0: Hardware startup is invalid when conditions match 1: Hardware startup is valid when conditions match	RW
b14	HSTAGCMA3	Unit 3 CMP A start enable	Condition: Unit 3 CMP A valid 0: Hardware startup is invalid when conditions match 1: Hardware startup is valid when conditions match	RW
b13	HSTAGCMB2	Unit 2 CMP B start enable	Condition: Unit 2 CMP B valid 0: Hardware startup is invalid when conditions match 1: Hardware startup is valid when conditions match	RW
b12	HSTAGCMA2	Unit 2 CMP A start enable	Condition: Unit 2 CMP A valid 0: Hardware startup is invalid when conditions match 1: Hardware startup is valid when conditions match	RW
b11	HSTAGCMB1	Unit 1 CMP B start enable	Condition: Unit 1 CMP B valid 0: Hardware startup is invalid when conditions match 1: Hardware startup is valid when conditions match	RW
b10	HSTAGCMA1	Unit 1 CMP A start enable	Condition: Unit 1 CMP A valid 0: Hardware startup is invalid when conditions match 1: Hardware startup is valid when conditions match	RW
b9	HSTAEXEV10	External event 10 start enable	Condition: External event 10 valid 0: Hardware startup is invalid when conditions match 1: Hardware startup is valid when conditions match	RW
b8	HSTAEXEV9	External event 9 start enable	Condition: External event 9 valid 0: Hardware startup is invalid when conditions match	RW

			1: Hardware startup is valid when conditions match	
b7	HSTAEXEV8	External event 8 start enable	Condition: External event 8 valid 0: Hardware startup is invalid when conditions match 1: Hardware startup is valid when conditions match	RW
b6	HSTAEXEV7	External event 7 start enable	Condition: External event 7 valid 0: Hardware startup is invalid when conditions match 1: Hardware startup is valid when conditions match	RW
b5	HSTAEXEV6	External event 6 start enable	Condition: External event 6 valid 0: Hardware startup is invalid when conditions match 1: Hardware startup is valid when conditions match	RW
b4	HSTAEXEV5	External event 5 start enable	Condition: External event 5 valid 0: Hardware startup is invalid when conditions match 1: Hardware startup is valid when conditions match	RW
b3	HSTAEXEV4	External event 4 start enable	Condition: External event 4 valid 0: Hardware startup is invalid when conditions match 1: Hardware startup is valid when conditions match	RW
b2	HSTAEXEV3	External event 3 start enable	Condition: External event 3 valid 0: Hardware startup is invalid when conditions match 1: Hardware startup is valid when conditions match	RW
b1	HSTAEXEV2	External event 2 start enable	Condition: External event 2 valid 0: Hardware startup is invalid when conditions match 1: Hardware startup is valid when conditions match	RW
b0	HSTAEXEV1	External event 1 start enable	Condition: External event 1 valid 0: Hardware startup is invalid when conditions match 1: Hardware startup is valid when conditions match	RW

19.6.1.21 Hardware Clear Register 2 (HRPWM<t>_HCLRR2)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res				HCLRSCMB6	HCLRSCMB5	HCLRSCMB4	HCLRSCMB3	HCLRSCMB2	HCLRSCMB1	HCLRGCMB6	HCLRGCMA6	HCLRGCMB5	HCLRGCMA5	HCLRGCMB4	HCLRGCMA4
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
HCLRGCMB3	HCLRGCMA3	HCLRGCMB2	HCLRGCMA2	HCLRGCMB1	HCLRGCMA1	HCLREXEV10	HCLREXEV9	HCLREXEV8	HCLREXEV7	HCLREXEV6	HCLREXEV5	HCLREXEV4	HCLREXEV3	HCLREXEV2	HCLREXEV1

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b28	Res	Reserved Bit	Read as "0", write as "0"	RW
b27	HCLRSCMB6	Unit 6 SCMPB clear enable	Condition: Unit 6 SCMPB valid 0: Hardware Clear is invalid when conditions match 1: Hardware Clear is valid when conditions match	RW
b26	HCLRSCMB5	Unit 5 SCMPB clear enable	Condition: Unit 5 SCMPB valid 0: Hardware Clear is invalid when conditions match 1: Hardware Clear is valid when conditions match	RW
b25	HCLRSCMB4	Unit 4 SCMPB clear enable	Condition: Unit 4 SCMPB valid 0: Hardware Clear is invalid when conditions match 1: Hardware Clear is valid when conditions match	RW
b24	HCLRSCMB3	Unit 3 SCMPB clear enable	Condition: Unit 3 SCMPB valid 0: Hardware Clear is invalid when conditions match 1: Hardware Clear is valid when conditions match	RW
b23	HCLRSCMB2	Unit 2 SCMPB clear enable	Condition: Unit 2 SCMPB valid 0: Hardware Clear is invalid when conditions match 1: Hardware Clear is valid when conditions match	RW
b22	HCLRSCMB1	Unit 1 SCMPB clear enable	Condition: Unit 1 SCMPB valid 0: Hardware Clear is invalid when conditions match 1: Hardware Clear is valid when conditions match	RW
b21	HCLRGCMB6	Unit 6 CMP B clear enable	Condition: Unit 6 CMP B valid 0: Hardware Clear is invalid when conditions match 1: Hardware Clear is valid when conditions match	RW
b20	HCLRGCMA6	Unit 6 CMP A clear enable	Condition: Unit 6 CMP A valid 0: Hardware Clear is invalid when conditions match 1: Hardware Clear is valid when conditions match	RW
b19	HCLRGCMB5	Unit 5 CMP B clear enable	Condition: Unit 5 CMP B valid 0: Hardware Clear is invalid when conditions match 1: Hardware Clear is valid when conditions match	RW
b18	HCLRGCMA5	Unit 5 CMP A clear enable	Condition: Unit 5 CMP A valid 0: Hardware Clear is invalid when conditions match 1: Hardware Clear is valid when conditions match	RW
b17	HCLRGCMB4	Unit 4 CMP B clear enable	Condition: Unit 4 CMP B valid 0: Hardware Clear is invalid when conditions match 1: Hardware Clear is valid when conditions match	RW
b16	HCLRGCMA4	Unit 4 CMP A clear enable	Condition: Unit 4 CMP A valid 0: Hardware Clear is invalid when conditions match 1: Hardware Clear is valid when conditions match	RW
b15	HCLRGCMB3	Unit 3 CMP B start enable	Condition: Unit 3 CMP B valid 0: Hardware Clear is invalid when conditions match 1: Hardware Clear is valid when conditions match	RW
b14	HCLRGCMA3	Unit 3 CMP A clear enable	Condition: Unit 3 CMP A valid 0: Hardware Clear is invalid when conditions match 1: Hardware Clear is valid when conditions match	RW
b13	HCLRGCMB2	Unit 2 CMP B clear enable	Condition: Unit 2 CMP B valid 0: Hardware Clear is invalid when conditions match 1: Hardware Clear is valid when conditions match	RW
b12	HCLRGCMA2	Unit 2 CMP A clear enable	Condition: Unit 2 CMP A valid 0: Hardware Clear is invalid when conditions match 1: Hardware Clear is valid when conditions match	RW
b11	HCLRGCMB1	Unit 1 CMP B clear enable	Condition: Unit 1 CMP B valid 0: Hardware Clear is invalid when conditions match 1: Hardware Clear is valid when conditions match	RW
b10	HCLRGCMA1	Unit 1 CMP A clear enable	Condition: Unit 1 CMP A valid 0: Hardware Clear is invalid when conditions match 1: Hardware Clear is valid when conditions match	RW
b9	HCLREXEV10	External event 10 clear enable	Condition: External event 10 valid 0: Hardware Clear is invalid when conditions match 1: Hardware Clear is valid when conditions match	RW
b8	HCLREXEV9	External event 9 clear enable	Condition: External event 9 valid 0: Hardware Clear is invalid when conditions match	RW

			1: Hardware Clear is valid when conditions match	
b7	HCLREXEV8	External event 8 clear enable	Condition: External event 8 valid 0: Hardware Clear is invalid when conditions match 1: Hardware Clear is valid when conditions match	RW
b6	HCLREXEV7	External event 7 clear enable	Condition: External event 7 valid 0: Hardware Clear is invalid when conditions match 1: Hardware Clear is valid when conditions match	RW
b5	HCLREXEV6	External event 6 clear enable	Condition: External event 6 valid 0: Hardware Clear is invalid when conditions match 1: Hardware Clear is valid when conditions match	RW
b4	HCLREXEV5	External event 5 clear enable	Condition: External event 5 valid 0: Hardware Clear is invalid when conditions match 1: Hardware Clear is valid when conditions match	RW
b3	HCLREXEV4	External event 4 clear enable	Condition: External event 4 valid 0: Hardware Clear is invalid when conditions match 1: Hardware Clear is valid when conditions match	RW
b2	HCLREXEV3	External event 3 clear enable	Condition: External event 3 valid 0: Hardware Clear is invalid when conditions match 1: Hardware Clear is valid when conditions match	RW
b1	HCLREXEV2	External event 2 clear enable	Condition: External event 2 valid 0: Hardware Clear is invalid when conditions match 1: Hardware Clear is valid when conditions match	RW
b0	HCLREXEV1	External event 1 clear enable	Condition: External event 1 valid 0: Hardware Clear is invalid when conditions match 1: Hardware Clear is valid when conditions match	RW

19.6.1.22 External Event Filter Offset m Register (HRPWM<t>_EEFOFFSETmR, m=A~B)

Reset value: 0x0040 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res									OFFSET DIR	OFFSET [21:16]					

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
OFFSET [15:6]										OFFSET[5:0]					

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b23	Res	Reserved Bit	Read as "0"	RW
b22	OFFSETDIR	Offset counting direction	0: The offset value is in the down counting range 1: The offset value is in the up counting range	RW
b21~b6	OFFSET [21:6]	Offset value	Blanking/Window Offset Value	RW
b5~b0	OFFSET[5:0]	Offset value	Read as "0"	R

19.6.1.23 External Event Filter Window m Register (HRPWM<t>_EEFWINmR, m=A~B)

Reset value: 0x0040 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res									WINDIR	WIN[21:16]					

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
WIN[15:6]										WIN[5:0]					

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b23	Res	Reserved Bit	Read as "0"	R
b22	WINDIR	Window value counting direction	0: The window value is in the downward counting range 1: The window value is in the up counting range	RW
b21~b6	WIN[21:6]	Window Value	Blanking/Window signal window value	RW
b5~b0	WIN[5:0]	Window Value	Read as "0"	R

19.6.1.24 External Event Filter Control Register (HRPWM<t>_EEFLTCR1)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res	EEINTPOL	EE5TMO	EE5FM[3:0]				EE5LAT	EE4TMO	EE4FM[3:0]				EE4LAT	EE3TMO	EE3FM[3]
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
EE3FM[2:0]			EE3LAT	EE2TM	EE2FM[3:0]				EE2LAT	EE1TMO	EE1FM[3:0]				EE1LAT

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b30	Res	Reserved Bit	Read as "0", write as "0"	RW
b30	EEINTPOL	External event filter initial level	0: Initial level is low 1: The initial level is high	RW
b29	EE5TMO	External event timeout function enable	Refer to EE1TMO instructions	RW
b28~b25	EE5FM[3:0]	Hardware trigger event 5 filter mode	Refer to EE1FM[3:0] description	RW
b24	EE5LAT	External event 5 storage function enable	Refer to EE1LAT instructions	RW
b23	EE4TMO	External event timeout function enable	Refer to EE1TMO instructions	RW
b22~b19	EE4FM[3:0]	Hardware trigger event 4 filter mode	Refer to EE1FM[3:0] description	RW
b18	EE4LAT	External event 4 storage function enable	Refer to EE1LAT instructions	RW
b17	EE3TMO	External event timeout function enable	Refer to EE1TMO instructions	RW
b16~b13	EE3FM[3:0]	Hardware trigger event 3 filter mode	Refer to EE1FM[3:0] description	RW
b12	EE3LAT	External event 3 storage function enable	Refer to EE1LAT instructions	RW
b11	EE2TMO	External event timeout function enable	Refer to EE1TMO instructions	RW
b10~b7	EE2FM[3:0]	Hardware trigger event 2 filter mode	Refer to EE1FM[3:0] description	RW
b6	EE2LAT	External event 2 storage function enable	Refer to EE1LAT instructions	RW
b5	EE1TMO	External event timeout function enable	0: Timeout function disable 1: Timeout Function Enable	RW
b4~b1	EE1FM[3:0]	External event 1 filter mode	0000: Filter mode disabled 0001: Blanking mode. Blanking time refers to the duration of low level blanking signal of this unit. 0010: Blanking mode. Blanking time refers to the duration of the low level of unit 1 blanking signal. 0011: Blanking mode. The blanking signal is HRPWM_1_B of unit 1. 0100: Blanking mode. Blanking time refers to the duration of the low level of unit 2 blanking signal 0101: Blanking mode. The blanking signal is HRPWM_2_B of unit 2. 0110: Blanking mode. Blanking time refers to the duration of the low level of unit 3 blanking signal. 0111: Blanking mode. The blanking signal is HRPWM_3_B of unit 3. 1000: Blanking mode. Blanking time refers to the duration of the low level of unit 4 blanking signal. 1001: Blanking mode. The blanking signal is HRPWM_4_B of unit 4. 1010: Blanking mode. Blanking time refers to the duration of the low level of unit 5 blanking signal. 1011: Blanking mode. The blanking signal is HRPWM_5_B of unit 5. 1100: Blanking mode. Blanking time refers to the duration of the low level of unit 6 blanking signal. 1101: Blanking mode. The blanking signal is HRPWM_6_B of unit 6. 1110: Window mode. Window time refers to the duration	RW

			of the high level of the window signal of this unit. 1111: Window mode. The window signal comes from other unit, TMRWIN source (Refer to Table 19-8). Filter mode 2 can be set by setting the EEFM and EEFREF bits of HRPWM<t>_GCONR1. For details, refer to Table 19-7 and Table 19-9.	
b0	EE1LAT	Event postpone mode enable	0: Event postpone mode disable 1: Event postpone mode Enable	RW

19.6.1.25 External Event Filter Control Register (HRPWM<t>_EEFLTCR2)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res		EE10TMO	EE10FM[3:0]				EE10LAT	EE9TMO	EE9FM[3:0]				EE9LAT	EE8TMO	EE8FM[3]

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
EE8FM[2:0]			EE8LAT	EE7TMO	EE7FM[3:0]				EE7LAT	EE6TMO	EE6FM[3:0]				EE6LAT

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b30	Res	Reserved Bit	Read as "0", write as "0"	RW
b29	EE10TMO	External event timeout function enable	Refer to EE1TMO instructions	RW
b28~b25	EE10FM[3:0]	Hardware trigger event 10 filter mode	Refer to EE1FM[3:0] description	RW
b24	EE10LAT	External event 10 storage function enable	Refer to EE1LAT instructions	RW
b23	EE9TMO	External event timeout function enable	Refer to EE1TMO instructions	RW
b22~b19	EE9FM[3:0]	Hardware trigger event 9 filter mode	Refer to EE1FM[3:0] description	RW
b18	EE9LAT	External event 9 storage function enable	Refer to EE1LAT instructions	RW
b17	EE8TMO	External event timeout function enable	Refer to EE1TMO instructions	RW
b16~b13	EE8FM[3:0]	Hardware trigger event 8 filter mode	Refer to EE1FM[3:0] description	RW
b12	EE8LAT	External event 8 storage function enable	Refer to EE1LAT instructions	RW
b11	EE7TMO	External event timeout function enable	Refer to EE1TMO instructions	RW
b10~b7	EE7FM[3:0]	Hardware trigger event 7 filter mode	Refer to EE1FM[3:0] description	RW
b6	EE7LAT	External event 7 storage function enable	Refer to EE1LAT instructions	RW
b5	EE6TMO	External event timeout function enable	Refer to EE1TMO instructions	RW
b4~b1	EE6FM[3:0]	External event 6 filter mode	Refer to EE1FM[3:0] description	RW
b0	EE6LAT	External event 6 storage function enable	Refer to EE1LAT instructions	RW

19.6.1.26 Idle Control Register (HRPWM<t>_IDLECR)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res						FOLLOW	INTENDLYPRT	Res				DLYCHB	DIDLB	IDLESB	IDLEBMB

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res			DLYSTRG	DLYEVSEL[2:0]			DLYPRTEN	Res				DLYCHA	DIDLA	IDLESA	IDLEBMA

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b26	Res	Reserved Bit	Read as "0", write as "0"	RW
b25	FOLLOW	Follow Enable	0: Following disable 1: Follow enable	RW
b24	INTENDLYPRT	Delayed Idle Interrupt Enable	0: Delayed idle entry interrupt disable 1: Delayed idle entry interrupt enable	RW
b23~b20	Res	Reserved Bit	Read as "0", write as "0"	RW
b19	DLYCHB	Delayed Idle Output B Enable	0: Output B does not enter delayed idle 1: Output B is triggered into delayed idle by the trigger source selected by DLYEVSEL[3:0]	RW
b18	DIDLB	Burst Mode delay entry enable B	0: Output is immediately idle 1: Output delay enters Burst Mode idle state	RW
b17	IDLESB	Port status setting B when idle	0: Set to low level when idle 1: Set to high level when idle	RW
b16	IDLEBMB	Burst Mode B enable	0: Output is not affected by Burst Mode 1: Burst Mode idle state is set by IDLESB	RW
b15~b13	Res	Reserved Bit	Read as "0", write as "0"	RW
b12	DLYSTRG	Delayed Idle Software Trigger	0: No impact 1: Software triggered delayed idle When using it, you need to set DLYEVSEL[2:0] to 100 first, and then write 1 to this bit.	RW
b11~b9	DLYEVSEL[2:0]	Delayed idle trigger source selection	000: Trigger source is external event 6 001: Trigger source is external event 7 010: Trigger source is external event 8 011: Trigger source is external event 9 100: The trigger source is software trigger (writing 1 to DLYSTRG generates a software trigger)	RW
b8	DLYPRTEN	Delayed Idle Enable	0: Delayed idle disable 1: Delay idle enable	RW
b7~b4	Res	Reserved Bit	Read as "0", write as "0"	RW
b3	DLYCHA	Delayed Idle Output A Enable	0: Output A does not enter delayed idle 1: Output A is triggered into delayed idle by the trigger source selected by DLYEVSEL[2:0]	RW
b2	DIDLA	Burst Mode delay entry enable A	0: Output is immediately idle 1: Output delay enters Burst Mode idle state	RW
b1	IDLESA	When idle, the port status is set to A	0: Set to low level when idle 1: Set to high level when idle	RW
b0	IDLEBMA	Burst Mode A enable	0: Output is not affected by Burst Mode 1: Burst Mode idle state is set by IDLESA	RW

19.6.1.27 General Control Register (HRPWM<t>_GCONR)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res							OVSTP	Res					MODE	Res	START

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b10	Res	Reserved Bit	Read as "0", write as "0"	RW
b8	OVSTP	Count overflow stop control	0: When the count value in triangle count mode is equal to 1 in down counting or the count value in sawtooth wave mode is equal to the period value, continue counting 1: In triangle count mode, when the downward counting value is equal to 1, counting stops. In sawtooth mode, the counter stops counting when the count value equals the period value.	RW
b7~b3	Res	Reserved Bit	Read as "0", write as "0"	RW
b2	MODE	Counting mode	0: Sawtooth counter mode 1: Triangular counter mode	RW
b1	Res	Reserved Bit	Read as "0", write as "0"	RW
b0	START	Timer start	0: Timer off 1: Timer Startup This bit will automatically change to 0 if the hardware stop condition is valid	RW

19.6.1.28 General Control Register (HRPWM<t>_GCONR1, t=1)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res									BFSEL[2:0]			CMSCBDEN	CMSCBUEN	CMSCADEN	CMSCAUEN
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res					PRDSEL[1:0]		Res	EEFREF	EEFM	Res	HALF	SWAPMD	SWAPEN	INVCBEN	INVCAEN

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b23	Res	Reserved Bit	Read as "0", write as "0"	RW
b22-b20	BFSEL[2:0]	The global Shadow completion flag Selection	BFSEL[0] : 0: Not set when a cycle buffer transfer point occurs 1: Set when a cycle buffer transfer point occurs BFSEL[1] : 0: Not set when a zero-buffer transfer point occurs 1: Set when a zero buffer transfer point occurs BFSEL[2] : 0: Not set when a single buffer transfer point occurs 1: Set when a single buffer transfer point occurs Point	RW
b19	CMSCBDEN	SCMPB down-counting output event enable	0: During down counting, when the SCMBR register and the count value are equal, the event output invalid 1: During down counting, when the SCMBR register and the count value are equal, the event output valid	RW
b18	CMSCBUEN	SCMPB up-counting output event enable B	0: During up counting, when the SCMBR register and the count value are equal, the event output invalid 1: During the up counting period, when the SCMBR register and the count value are equal, the event output valid	RW
b17	CMSCADEN	SCMPA down-counting output event enable A	0: During down counting, when the SCMAR register and the count value are equal, the event output invalid 1: During down counting, when the SCMAR register and the count value are equal, the event output valid	RW
b16	CMSCAUEN	SCMPA up-counting output event enable A	0: During up counting, when the SCMAR register and the count value are equal, the event output invalid 1: During up counting, when the SCMAR register and the count value are equal, the event output valid	RW
b15~b11	Res	Reserved Bit	Read as "0", write as "0"	RW
b10~b9	PRDSEL[1:0]	Full period time selection	00: The count value in sawtooth wave mode is equal to the period value or hardware cleared; the count value in triangle count mode is equal to 0 01: The count value is equal to the period value or the sawtooth wave is cleared by hardware 10: The count value is equal to 0 11: The count value is equal to the period value or the sawtooth wave hardware is cleared or the count value is equal to 0	RW
b8	Res	Reserved Bit	Read as "0", write as "0"	RW
b7	EEFREF	External event filter mode reference slection	0: The reference point of the external event filter mode is the period matching point 1: External event filter mode reference point is zero matching point	RW
b6	EEFM	External event filtering mode selection	0: External event filtering mode 1 1: External event filtering mode 2	RW
b5	Res	Reserved Bit	Read as "0", write as "0"	RW
b4	HALF	Half-wave mode enable	0: Half-wave mode disable 1: half-wave mode enabled	RW
b3	SWAPMD	Exchange Mode	0: Full period point swap or no swap 1: Immediate exchange or no exchange	RW
b2	SWAPEN	Output A and B swap enable	0: Outputs A and B are not swapped 1: Output A and output B are swapped	RW
b1	INVCBEN	Output B polarity inversion enable	0: Output B polarity is not reversed 1: Output B polarity inversion	RW
b0	INVCAEN	Output A polarity inversion enable	0: Output A polarity is not reversed 1: Output A polarity inversion	RW

19.6.1.29 General Control Shadow Register (HRPWM<t>_BGCONR1, t=1)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res									BFSEL[2:0]			CMSCBDEN	CMSCBUEN	CMSCADEN	CMSCAUEN
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res					PRDSEL[1:0]		Res	EEFREF	EEFM	Res	HALF	SWAPMD	SWAPEN	INVCBEN	INVCAEN

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b22	Res	Reserved Bit	Read as "0", write as "0"	RW
b22-b20	BFSEL[2:0]	The global Shadow completion flag Selection	BFSEL[0] : 0: Not set when a cycle buffer transfer point occurs 1: Set when a cycle buffer transfer point occurs BFSEL[1] : 0: Not set when a zero-buffer transfer point occurs 1: Set when a zero buffer transfer point occurs BFSEL[2] : 0: Not set when a single buffer transfer point occurs 1: Set when a single buffer transfer point occurs Point	RW
b19	CMSCBDEN	SCMPB down-counting output event enable B	0: During down counting, when the SCMBR register and the count value are equal, the event output invalid 1: During down counting, when the SCMBR register and the count value are equal, the event output valid	RW
b18	CMSCBUEN	SCMPB up-counting output event enable B	0: During up counting, when the SCMBR register and the count value are equal, the event output invalid 1: During the up counting period, when the SCMBR register and the count value are equal, the event output valid	RW
b17	CMSCADEN	SCMPA down-counting output event enable A	0: During down counting, when the SCMAR register and the count value are equal, the event output invalid 1: During down counting, when the SCMAR register and the count value are equal, the event output valid	RW
b16	CMSCAUEN	SCMPA up-counting output event enable A	0: During up counting, when the SCMAR register and the count value are equal, the event output invalid 1: During up counting, when the SCMAR register and the count value are equal, the event output valid	RW
b15~b11	Res	Reserved Bit	Read as "0", write as "0"	RW
b10~b9	PRDSEL[1:0]	Full period time selection	00: The count value in sawtooth wave mode is equal to the period value or hardware cleared; the count value in triangle count mode is equal to 0 01: The count value is equal to the period value or the sawtooth wave is cleared by hardware 10: The count value is equal to 0 11: The count value is equal to the period value or the sawtooth wave hardware is cleared or the count value is equal to 0	RW
b8	Res	Reserved Bit	Read as "0", write as "0"	RW
b7	EEFREF	External event filter mode reference point	0: The reference point of the external event filter mode is the period matching point 1: External event filter mode reference point is zero matching point	RW
b6	EEFM	External event filtering mode selection	0: External event filtering mode 1 1: External event filtering mode 2	RW
b5	Res	Reserved Bit	Read as "0", write as "0"	RW
b4	HALF	Half-wave mode enable	0: Half-wave mode disable 1: half-wave mode enabled	RW
b3	SWAPMD	Exchange Mode	0: Full period point swap or no swap 1: Immediate exchange or no exchange	RW
b2	SWAPEN	Output A and B swap enable	0: Outputs A and B are not swapped 1: Output A and output B are swapped	RW
b1	INVCBEN	Output B polarity inversion enable	0: Output B polarity is not reversed 1: Output B polarity inversion	RW
b0	INVCAEN	Output A polarity inversion enable	0: Output A polarity is not reversed 1: Output A polarity inversion	RW

19.6.1.30 Universal Control Register (HRPWM<t>_GCONR1, t=2~6)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res								BFSEL[3:0]				CMSCBDEN	CMSCBUEN	CMSCADEN	CMSCAUEN
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res					PRDSEL[1:0]		Res	EEFREF	EEFM	PRDLK	HALF	SWAPMD	SWAPEN	INVCBEN	INVCAEN

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b24	Res	Reserved Bit	Read as "0", write as "0"	RW
b23-b20	BFSEL[3:0]	The global Shadow completion flag Selection	BFSEL[0] : 0: Not set when a cycle buffer transfer point occurs 1: Set when a cycle buffer transfer point occurs BFSEL[1] : 0: Not set when a zero-buffer transfer point occurs 1: Set when a zero buffer transfer point occurs BFSEL[2] : 0: Not set when a single buffer transfer point occurs in unit 1 1: Set when a single buffer transfer point occurs in unit 1 Point BFSEL[3] : 0: Not set when the sawtooth wave period **Shadow Transfer Time** or triangle wave two-point **Shadow Transfer Time** of this unit occurs after the unit 1 Single Shadow Transfer Time 1: Set when a sawtooth wave period **Shadow Transfer Time** or a triangle wave two-point **Shadow Transfer Time** occurs after a Single Shadow Transfer Time of unit 1	RW
b19	CMSCBDEN	SCMPB down-counting output event enable B	0: During down counting, when the SCMBR register and the count value are equal, the event output invalid 1: During down counting, when the SCMBR register and the count value are equal, the event output valid	RW
b18	CMSCBUEN	SCMPB up-counting output event enable B	0: During up counting, when the SCMBR register and the count value are equal, the event output invalid 1: During the up counting period, when the SCMBR register and the count value are equal, the event output valid	RW
b17	CMSCADEN	SCMPA down-counting output event enable A	0: During down counting, when the SCMAR register and the count value are equal, the event output invalid 1: During down counting, when the SCMAR register and the count value are equal, the event output valid	RW
b16	CMSCAUEN	SCMPA up-counting output event enable A	0: During up counting, when the SCMAR register and the count value are equal, the event output invalid 1: During up counting, when the SCMAR register and the count value are equal, the event output valid	RW
b15~b11	Res	Reserved Bit	Read as "0", write as "0"	RW
b10~b9	PRDSEL[1:0]	Full period point selection	00: The count value in sawtooth wave mode is equal to the period value or hardware cleared; the count value in triangle count mode is equal to 0 01: The count value is equal to the period value or the sawtooth wave is cleared by hardware 10: The count value is equal to 0 11: The count value is equal to the period value or the sawtooth wave hardware is cleared or the count value is equal to 0	RW
b8	Res	Reserved Bit	Read as "0", write as "0"	RW
b7	EEFREF	External event filter mode reference point	0: The reference point of the external event filter mode is the period matching point 1: External event filter mode reference point is zero matching point	RW
b6	EEFM	External event filtering mode selection	0: External event filtering mode 1 1: External event filtering mode 2	RW

b5	PRDLK	Period link enable	0: Period link disable 1: Period link enable Only supports units 2~6	RW
b4	HALF	Half-wave mode enable	0: Half-wave mode disable 1: half-wave mode enabled	RW
b3	SWAPMD	Exchange Mode	0: Full period point swap or no swap 1: Immediate exchange or no exchange	RW
b2	SWAPEN	Output A and B swap enable	0: Outputs A and B are not swapped 1: Output A and output B are swapped	RW
b1	INVCBEN	Output B polarity inversion enable	0: Output B polarity is not reversed 1: Output B polarity inversion	RW
b0	INVCAEN	Output A polarity inversion enable	0: Output A polarity is not reversed 1: Output A polarity inversion	RW

19.6.1.31 General Control Shadow Register (HRPWM<t>_BGCONR1, t=2~6)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res								BFSEL[3:0]				CMSCBDEN	CMSCBUEN	CMSCADEN	CMSCAUEN

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res					PRDSEL[1:0]		Res	EEFREF	EEFM	PRDLK	HALF	SWAPMD	SWAPEN	INVCBEN	INVCAEN

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b24	Res	Reserved Bit	Read as "0", write as "0"	RW
b23~b20	BFSEL[3:0]	The global Shadow completion flag Selection	BFSEL[0] : 0: Not set when a cycle buffer transfer point occurs 1: Set when a cycle buffer transfer point occurs BFSEL[1] : 0: Not set when a zero-buffer transfer point occurs 1: Set when a zero buffer transfer point occurs BFSEL[2] : 0: Not set when a single buffer transfer point occurs in unit 1 1: Set when a single buffer transfer point occurs in unit 1 Point BFSEL[3] : 0: Not set when the sawtooth wave period **Shadow Transfer Time** or triangle wave two-point **Shadow Transfer Time** of this unit occurs after the unit 1 Single Shadow Transfer Time 1: Set when a sawtooth wave period **Shadow Transfer Time** or a triangle wave two-point **Shadow Transfer Time** occurs after a Single Shadow Transfer Time of unit 1	RW
b19	CMSCBDEN	SCMBR down-counting output event enable B	0: During down counting, when the SCMBR register and the count value are equal, the event output invalid 1: During down counting, when the SCMBR register and the count value are equal, the event output valid	RW
b18	CMSCBUEN	SCMBR up-counting output event enable B	0: During up counting, when the SCMBR register and the count value are equal, the event output invalid 1: During the up counting period, when the SCMBR register and the count value are equal, the event output valid	RW
b17	CMSCADEN	SCMAR down-counting output event enable A	0: During down counting, when the SCMAR register and the count value are equal, the event output invalid 1: During down counting, when the SCMAR register and the count value are equal, the event output valid	RW
b16	CMSCAUEN	SCMAR up-counting output event enable A	0: During up counting, when the SCMAR register and the count value are equal, the event output invalid 1: During up counting, when the SCMAR register and the count value are equal, the event output valid	RW
b15~b11	Res	Reserved Bit	Read as "0", write as "0"	RW
b10~b9	PRDSEL[1:0]	Full period point selection	00: The count value in sawtooth wave mode is equal to the period value or hardware cleared; the count value in triangle count mode is equal to 0 01: The count value is equal to the period value or the sawtooth wave is cleared by hardware 10: The count value is equal to 0 11: The count value is equal to the period value or the sawtooth wave hardware is cleared or the count value is equal to 0	RW
b8	Res	Reserved Bit	Read as "0", write as "0"	RW
b7	EEFREF	External event filter mode reference point	0: The reference point of the external event filter mode is the period matching point 1: External event filter mode reference point is zero matching point	RW
b6	EEFM	External event filtering mode selection	0: External event filtering mode 1 1: External event filtering mode 2	RW

b5	PRDLK	Period link enable	0: Period link disable 1: Period link enable Only supports units 2~6	RW
b4	HALF	Half-wave mode enable	0: Half-wave mode disable 1: half-wave mode enabled	RW
b3	SWAPMD	Exchange Mode	0: Full period point swap or no swap 1: Immediate exchange or no exchange	RW
b2	SWAPEN	Output A and B swap enable	0: Outputs A and B are not swapped 1: Output A and output B are swapped	RW
b1	INVCBEN	Output B polarity inversion enable	0: Output B polarity is not reversed 1: Output B polarity inversion	RW
b0	INVCAEN	Output A polarity inversion enable	0: Output A polarity is not reversed 1: Output A polarity inversion	RW

19.6.1.32 Shadow Control Register 1 (HRPWM<t>_BCONR1)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res								BTR DSPB	BTR USPB	Res	BEN SPB	BTR DSPA	BTR USPA	Res	BEN SPA

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res				BTR DP	BTR UP	Res	BEN P	BTR DBF	BTR UBF	Res	BEN BF	BTR DAE	BTR UAE	Res	BEN AE

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b24	Res	Reserved Bit	Read as "0", write as "0"	RW
b23	BTRDSPB	SCMBR shadow transfer time setting selection(ZERO)	Refer to BTRDAE instructions	RW
b22	BTRUSPB	SCMBR shadow transfer time setting selection(period)	Refer to BTRUAE instructions	RW
b21	Res	Reserved Bit	Read as "0", write as "0"	RW
b20	BENSPB	SCMBR shadow transfer enable	0: Buffer transmission is invalid 1: Buffer transfer enable	RW
b19	BTRDSPA	SCMAR shadow transfer time setting selection(ZERO)	Refer to BTRDAE instructions	RW
b18	BTRUSPA	SCMAR shadow transfer time setting selection(period)	Refer to BTRUAE instructions	RW
b17	Res	Reserved Bit	Read as "0", write as "0"	RW
b16	BENSPA	SCMAR shadow transfer enable	0: Buffer transmission is invalid 1: Buffer transfer enable	RW
b15~b12	Res	Reserved Bit	Read as "0", write as "0"	RW
b11	BTRDP	Period shadow transfer time setting selection(ZERO)	Refer to BTRDAE instructions	RW
b10	BTRUP	Period shadow transfer time setting selection(period)	Refer to BTRUAE instructions	RW
b9	Res	Reserved Bit	Read as "0", write as "0"	RW
b8	BENP	Period shadow transfer enable	0: Buffer transmission is invalid 1: Buffer transfer enable	RW
b7	BTRDBF	General compare BF shadow transfer time setting selection(zero)	Refer to BTRDAE instructions	RW
b6	BTRUBF	General compare BF shadow transfer time setting selection(period)	Refer to BTRUAE instructions	RW
b5	Res	Reserved Bit	Read as "0", write as "0"	RW
b4	BENBF	General compare BF shadow transfer enable	0: shadow transmission is invalid 1: shadow transfer enable	RW
b3	BTRDAE	General compare AE shadow transfer time setting selection(zero)	0: Zero **Shadow Transfer Time**, Shadow value is not transferred 1: Zero **Shadow Transfer Time**, a Shadow value transfer occurs In sawtooth wave mode, the Shadow point is when the count value is equal to 0; in triangle count mode, the Shadow point is when the count value is equal to 64 or the hardware is cleared.	RW
b2	BTRUAE	General compare AE shadow transfer time setting selection(period)	0: Period **Shadow Transfer Time**, Shadow value is not transferred 1: Period **Shadow Transfer Time**, a Shadow value transfer occurs In sawtooth wave mode, the Shadow point is when the count value is equal to the period value or hardware reset occurs; in triangle count mode, the Shadow point is when the count value is equal to the period - 64.	RW
b1	Res	Reserved Bit	Read as "0", write as "0"	RW

b0	BENAE	General compare AE shadow transfer enable	0: shadow transmission is invalid 1: shadow transfer enable	RW

19.6.1.33 Shadow Control Register 2 (HRPWM<t>_BCONR2, t=1)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res	BTR DCTL	BTR UCTL	BEN CTL	Res						BTR DEEF WIN	BTR UEEF WIN	BEN EEFW IN	BTR DEEF OFF	BTR UEEF OFF	BEN EEFO FF

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b15	Res	Reserved Bit	Read as "0", write as "0"	RW
b14	BTRDCTL	Control register Shadow transfer time settingSelction(ZERO)	Refer to BTRDAE instructions	RW
b13	BTRUCTL	Control register Shadow transfer time settingSelction(Reriod)	Refer to BTRUAE instructions	RW
b12	BENCTL	Control register shadow transfer enable	0: Buffer transmission is invalid 1: Buffer transfer enable (BICONR-> ICONR BPCNAR1->PCNAR1 BPCNBR1->PCNBR1 BPCNAR2->PCNAR2 BPCNBR2->PCNBR2 BPCNAR3->PCNAR3 BPCNBR3->PCNBR3 BGCONR1->GCONR1)	RW
b11~b6	Res	Reserved Bit	Read as "0", write as "0"	RW
b5	BTRDEEFWIN	External event filter window Shadow transfer time settingSelction(ZERO)	Refer to BTRDAE instructions	RW
b4	BTRUEEFWIN	External event filter window Shadow transfer time setting Selction(Reriod)	Refer to BTRUAE instructions	RW
b3	BENEEFWIN	External event filter window shadow transfer enable	0: Buffer transmission is invalid 1: Shadow transfer enable (EEFWINBR->EEFWINAR)	RW
b2	BTRDEEFOFF	External event filter offset Shadow transfer time setting Selction(ZERO)	Refer to BTRDAE instructions	RW
b1	BTRUEEFOFF	External event filter offset v Shadow transfer time setting Selction(Reriod)	Refer to BTRUAE instructions	RW
b0	BENEEFOFF	External event filter offset shadow transfer enable	0: Buffer transmission is invalid 1: Buffer transfer enable (EEFOFFSETBR->EEFOFFSETAR)	RW

19.6.1.34 Shadow Control Register 2 (HRPWM<t>_BCONR2, t=2~6)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res		BTRU0PCTL	BTRU0PDT	BTRU0PSPB	BTRU0PSPA	BTRU0PP	BTRU0PBF	BTRU0PAE	BTRU0CTL	BTRU0DT	BTRU0SPB	BTRU0SPA	BTRU0P	BTRU0BF	BTRU0AE
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res	BTRDCTL	BTRUCTL	BENCTL	Res		BTRU0PEEFWIN	BTRU0PEEFOFF	BTRU0EEFWIN	BTRU0EEFOFF	BTRDEEFWIN	BTRUEEFWIN	BENEEFWIN	BTRDEEFOFF	BTRUEEFOFF	BENEEFOFF

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b30	Res	Reserved Bit	Read as "0", write as "0"	RW
b29	BTRU0PCTL	Control register Shadow transfer U0PCTL	0: Single Shadow Transfer Time in unit 1, Shadow value is not transferred 1: A Shadow value transfer occurs at the unit 1 Single Shadow Transfer Time	RW
b28	BTRU0PDT	Dead time value buffer transfer U0PDT	0: Single Shadow Transfer Time in unit 1, Shadow value is not transferred 1: A Shadow value transfer occurs at the unit 1 Single Shadow Transfer Time	RW
b27	BTRU0PSPB	Special-compare buffer transfer time setting U0PSPB	0: Single Shadow Transfer Time in unit 1, Shadow value is not transferred 1: A Shadow value transfer occurs at the unit 1 Single Shadow Transfer Time	RW
b26	BTRU0PSPA	Special-compare buffer transfer time setting U0PSPA	0: Single Shadow Transfer Time in unit 1, Shadow value is not transferred 1: A Shadow value transfer occurs at the unit 1 Single Shadow Transfer Time	RW
b25	BTRU0PP	Period value buffer transfer time setting U0PP	0: Single Shadow Transfer Time in unit 1, Shadow value is not transferred 1: A Shadow value transfer occurs at the unit 1 Single Shadow Transfer Time	RW
b24	BTRU0PBF	General compare shadow transfer time setting U0PBF	0: Single Shadow Transfer Time in unit 1, Shadow value is not transferred 1: A Shadow value transfer occurs at the unit 1 Single Shadow Transfer Time	RW
b23	BTRU0PAE	General compare shadow transfer time setting U0PAE	0: Single Shadow Transfer Time in unit 1, Shadow value is not transferred 1: A Shadow value transfer occurs at the unit 1 Single Shadow Transfer Time	RW
b22	BTRU0CTL	Control register Shadow transfer U0CTL	0: After the Single Shadow Transfer Time of unit 1, the sawtooth wave period **Shadow Transfer Time** or triangle wave zero point **Shadow Transfer Time** of this unit is not transferred. 1: After the Single Shadow Transfer Time of unit 1, the sawtooth wave period **Shadow Transfer Time** or triangle wave zero point **Shadow Transfer Time** of this unit has a Shadow value transfer. **Note:** The setting valid only when GBCONR.OSTENU1=1	RW
b21	BTRU0DT	Dead time value buffer transfer U0DT	0: After the Single Shadow Transfer Time of unit 1, the Shadow value of the sawtooth wave period **Shadow Transfer Time** or the triangle wave zero point **Shadow Transfer Time** of this unit is not transferred 1: After the Single Shadow Transfer Time of unit 1, the sawtooth wave period **Shadow Transfer Time** or triangle wave zero point **Shadow Transfer Time** of this unit has a Shadow value transfer. **Note:** The setting valid only when GBCONR.OSTENU1=1	RW
b20	BTRU0SPB	Special-compare buffer transfer time setting U0SPB	0: After the Single Shadow Transfer Time of unit 1, the Shadow value of the sawtooth wave period **Shadow Transfer Time** or the triangle wave zero point **Shadow Transfer Time** of this unit is not transferred 1: After the Single Shadow Transfer Time of unit 1, the sawtooth wave period **Shadow Transfer Time** or triangle wave zero point **Shadow Transfer Time** of this unit has a Shadow value transfer. **Note:** The setting valid only when GBCONR.OSTENU1=1	RW

b19	BTRU0SPA	Special-compare buffer transfer time setting U0SPA	0: After the Single Shadow Transfer Time of unit 1, the Shadow value of the sawtooth wave period **Shadow Transfer Time** or the triangle wave zero point **Shadow Transfer Time** of this unit is not transferred 1: After the Single Shadow Transfer Time of unit 1, the sawtooth wave period **Shadow Transfer Time** or triangle wave zero point **Shadow Transfer Time** of this unit has a Shadow value transfer. **Note:** The setting valid only when GBCONR.OSTENU1=1	RW
b18	BTRU0P	Cycle value buffer transfer time setting U0P	0: After the Single Shadow Transfer Time of unit 1, the Shadow value of the sawtooth wave period **Shadow Transfer Time** or the triangle wave zero point **Shadow Transfer Time** of this unit is not transferred 1: After the Single Shadow Transfer Time of unit 1, the sawtooth wave period **Shadow Transfer Time** or triangle wave zero point **Shadow Transfer Time** of this unit has a Shadow value transfer. **Note:** The setting valid only when GBCONR.OSTENU1=1	RW
b17	BTRU0BF	General compare shadow transfer time setting U0BF	0: After the Single Shadow Transfer Time of unit 1, the Shadow value of the sawtooth wave period **Shadow Transfer Time** or the triangle wave zero point **Shadow Transfer Time** of this unit is not transferred 1: After the Single Shadow Transfer Time of unit 1, the sawtooth wave period **Shadow Transfer Time** or triangle wave zero point **Shadow Transfer Time** of this unit has a Shadow value transfer. Note: This setting is only valid when GBCONR.OSTENU1=1	RW
b16	BTRU0AE	General compare shadow transfer time setting U0AE	0: After the Single Shadow Transfer Time of unit 1, the Shadow value of the sawtooth wave period **Shadow Transfer Time** or the triangle wave zero point **Shadow Transfer Time** of this unit is not transferred 1: After the Single Shadow Transfer Time of unit 1, the sawtooth wave period **Shadow Transfer Time** or triangle wave zero point **Shadow Transfer Time** of this unit has a Shadow value transfer. **Note:** The setting valid only when GBCONR.OSTENU1=1	RW
b15	Res	Reserved Bit	Read as "0", write as "0"	RW
b14	BTRDCTL	Control register Shadow transfer time setting DCTL	Refer to BTRDAE instructions	RW
b13	BTRUCTL	Control register Shadow transfer time setting UCTL	Refer to BTRUAE instructions	RW
b12	BENCTL	Control register buffer transfer	0: Buffer transmission is invalid 1: Buffer transfer enable (BICONR-> ICONR BPCNAR1->PCNAR1 BPCNBR1->PCNBR1 BPCNAR2->PCNAR2 BPCNBR2->PCNBR2 BPCNAR3->PCNAR3 BPCNBR3->PCNBR3 BGCONR1->GCONR1)	RW
b11~b10	Res	Reserved Bit	Read as "0", write as "0"	RW
b9	BTRU0PEEFWIN	External event filter window buffer transfer time setting U0PWIN	0: Single Shadow Transfer Time in unit 1, Shadow value is not transferred 1: A Shadow value transfer occurs at the unit 1 Single Shadow Transfer Time	RW
b8	BTRU0PEEFOFF	External event filter window buffer transfer time setting U0POFF	0: Single Shadow Transfer Time in unit 1, Shadow value is not transferred 1: A Shadow value transfer occurs at the unit 1 Single Shadow Transfer Time	RW
b7	BTRU0EEFWIN	External event filter window buffer transfer time setting U0WIN	0: After the Single Shadow Transfer Time of unit 1, the sawtooth wave period **Shadow Transfer Time** or triangle wave zero point **Shadow Transfer Time** of this unit, the Shadow value is not transferred 1: After the Single Shadow Transfer Time of unit 1, the sawtooth wave period **Shadow Transfer Time** or triangle wave zero point **Shadow Transfer Time** of this unit occurs a Shadow value transfer **Note:** The setting valid only when PHSCTL.PHSEN=1 and GBCONR.OSTENU1=1	RW

b6	BTRU0EEFOFF	External event filter window buffer transfer time setting U0OFF	0: After the Single Shadow Transfer Time of unit 1, the sawtooth wave period **Shadow Transfer Time** or triangle wave zero point **Shadow Transfer Time** of this unit, the Shadow value is not transferred 1: After the Single Shadow Transfer Time of unit 1, the sawtooth wave period **Shadow Transfer Time** or triangle wave zero point **Shadow Transfer Time** of this unit occurs a Shadow value transfer Note: This setting valid only when PHSCTL.PHSEN=1 and GBCONR.OSTENU1=1	RW
b5	BTRDEEFWIN	External event filter window buffer transfer time setting DWIN	Refer to BTRDAE instructions	RW
b4	BTRUEEFWIN	External event filter window Shadow transfer time setting UWIN	Refer to BTRUAE instructions	RW
b3	BENEEFWIN	External event filter window buffer transfer	0: Buffer transmission is invalid 1: Shadow transfer enable (EEFWINBR->EEFWINAR)	RW
b2	BTRDEEFOFF	External event filter offset value buffer transfer time setting DOFF	Refer to BTRDAE instructions	RW
b1	BTRUEEFOFF	External event filter offset value buffer transfer time setting UOFF	Refer to BTRUAE instructions	RW
b0	BENEEFOFF	External event filter offset value buffer transfer	0: Buffer transmission is invalid 1: Buffer transfer enable (EEFOFFSETBR->EEFOFFSETAR)	RW

19.6.1.35 Port A Control Register 1 (HRPWM<t>_PCNAR1)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res			OUTENA	Res	EMBS A[2:0]			EMBR A[1:0]		EMBC A[1:0]		Res		FORC A[1:0]	

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
CMBDC A[1:0]		CMBUC A[1:0]		CMADC A[1:0]		CMAUC A[1:0]		UDFC A[1:0]		OVFC A[1:0]		STPC A[1:0]		STAC A[1:0]	

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b29	Res	Reserved Bit	Read as "0", write as "0"	RW
b28	OUTENA	Output enable A	0: HRPWM_<t>_PWMA port output invalid in HRPWM function 1: HRPWM_<t>_PWMA port output valid when HRPWM function is in progress	RW
b27	Res	Reserved Bit	Read as "0", write as "0"	RW
b26~b24	EMBSA[2:0]	EMB selection	000: Select EMB event channel 0 to be valid 001: Select EMB event channel 1 to be valid 010: Select EMB event channel 2 to be valid 011: Select EMB event channel 3 to be valid 100: Select EMB event channel 4 to be valid 101: Select EMB event channel 5 to be valid	RW
b23~b22	EMBRA[1:0]	Select A for EMB release mode	00: When the EMB event of the selected channel is invalid, the HRPWM_<t>_PWMA port is released immediately (One Shot) 01: When the EMB event of the selected channel is invalid, the HRPWM_<t>_PWMA port is released when the cycle matches (Cycle By Cycle 1) 10: When the EMB event of the selected channel is invalid, the HRPWM_<t>_PWMA port is released when the zero point matches (Cycle By Cycle 2) 11: When the EMB event of the selected channel is invalid, the HRPWM_<t>_PWMA port is released when the cycle matches or the zero point matches (Cycle By Cycle 3)	RW
b21~b20	EMBCA[1:0]	PortA state set EMB event occurs	00: When an EMB event occurs on the selected channel, the HRPWM_<t>_PWMA port outputs normally. 01: When an EMB event occurs on the selected channel, the HRPWM_<t>_PWMA port outputs a high impedance state. 10: When an EMB event occurs on the selected channel, the HRPWM_<t>_PWMA port outputs a low level 11: When an EMB event occurs on the selected channel, the HRPWM_<t>_PWMA port outputs a high level	RW
b19~b18	Res	Reserved Bit	Read as "0", write as "0"	RW
b17~b16	FORCA[1:0]	Force port A status	0x: Set invalid 10: The next cycle starts, and the HRPWM_<t>_PWMA port output is set to a low level 11: The next cycle starts, and the HRPWM_<t>_PWMA port output is set to a high level Note 1: The next cycle refers to the Full period point configured by register HRPWM_<t>_GCONR1.PRDSEL Note 2: This register bit can be used to achieve 0% or 100% control of PWM output duty cycle.	RW
b15~b14	CMBDCA[1:0]	Action When Counter = SCMBR on Down Count	00: During down counting, when the timer count value is equal to GCMBR, the HRPWM_<t>_PWMA port output is set to low level 01: During down counting, when the timer count value is equal to GCMBR, the HRPWM_<t>_PWMA port output is set to high level 10: During down counting, when the timer count value is equal to GCMBR, the HRPWM_<t>_PWMA port output is not affected 11: During down counting, when the timer count value is equal to GCMBR, the HRPWM_<t>_PWMA port output is set to the inverted level	RW
b13~b12	CMBUCA[1:0]	Action When Counter = SCMBR on Up Count	00: During the up counting period, when the timer count value is equal to GCMBR, the HRPWM_<t>_PWMA port output is set to low level 01: During the up counting period, when the timer count value is equal to GCMBR, the HRPWM_<t>_PWMA port output is set to high level 10: During the up counting period, when the timer count value is equal to GCMBR, the HRPWM_<t>_PWMA port output is not affected	RW

			11: During the up counting period, when the timer count value is equal to GCMBR, the HRPWM_<t>_PWMA port output is set to the inverted level	
b11~b10	CMADCA[1:0]	Action When Counter = SCMAR on Down Count	00: During down counting, when the timer count value is equal to GCMAR, the HRPWM_<t>_PWMA port output is set to low level 01: During down counting, when the timer count value is equal to GCMAR, the HRPWM_<t>_PWMA port output is set to high level 10: During down counting, when the timer count value is equal to GCMAR, the HRPWM_<t>_PWMA port output is not affected 11: During down counting, when the timer count value is equal to GCMAR, the HRPWM_<t>_PWMA port output is set to the inverted level	RW
b9~b8	CMAUCA[1:0]	Action When Counter = SCMAR on Up Count	00: During the up counting period, when the timer count value is equal to GCMAR, the HRPWM_<t>_PWMA port output is set to low level 01: During the up counting period, when the timer count value is equal to GCMAR, the HRPWM_<t>_PWMA port output is set to high level 10: During the up counting period, when the timer count value is equal to GCMAR, the HRPWM_<t>_PWMA port output is not affected 11: During the up counting period, when the timer count value is equal to GCMAR, the HRPWM_<t>_PWMA port output is set to the inverted level	RW
b7~b6	UDFCA[1:0]	Action When Counter = 0	00: When the count value is equal to 0, the HRPWM_<t>_PWMA port output is set to low level 01: When the count value is equal to 0, the HRPWM_<t>_PWMA port output is set to high level 10: When the count value is equal to 0, the output of the HRPWM_<t>_PWMA port is not affected 11: When the count value is equal to 0, the HRPWM_<t>_PWMA port output is set to the inverted level	RW
b5~b4	OVFCA[1:0]	Action When Counter = HRPERAR	00: When the count value is equal to the period value or the sawtooth wave hardware is cleared, the HRPWM_<t>_PWMA port output is set to a low level 01: When the count value is equal to the period value or the sawtooth wave hardware is cleared, the HRPWM_<t>_PWMA port output is set to a high level 10: When the count value is equal to the period value or the sawtooth wave hardware is cleared, the HRPWM_<t>_PWMA port output is not affected 11: When the count value is equal to the period value or the sawtooth wave hardware is cleared, the HRPWM_<t>_PWMA port output is set to the inverted level	RW
b3~b2	STPCA[1:0]	Action When Counter Stop	00: When counting stops, the HRPWM_<t>_PWMA port output is set to low level 01: When counting stops, the HRPWM_<t>_PWMA port output is set to high level 10: When counting stops, the HRPWM_<t>_PWMA port output maintains the previous state 11: When counting stops, the HRPWM_<t>_PWMA port output maintains the previous state	RW
b1~b0	STACA[1:0]	Action When Counter Start	00: When counting starts, the HRPWM_<t>_PWMA port output is set to low level 01: When counting starts, the HRPWM_<t>_PWMA port output is set to high level 10: When counting starts, the HRPWM_<t>_PWMA port output maintains the previous state 11: When counting starts, the HRPWM_<t>_PWMA port output maintains the previous state -	RW

19.6.1.36 Port B Control Register 1 (HRPWM<t>_PCNBR1)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res			OUTENB	Res	EMBSB[1:0]			EMBRB[1:0]		EMBCB[1:0]		Res		FORCB[1:0]	
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
CMBDCB[1:0]		CMBUCB[1:0]		CMADCB[1:0]		CMAUCB[1:0]		UDFCB[1:0]		OVFCB[1:0]		STPCB[1:0]		STACB[1:0]	

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b29	Res	Reserved Bit	Read as "0", write as "0"	RW
b28	OUTENB	Output enable B	0: HRPWM_<t>_PWMB port output invalid in HRPWM function 1: HRPWM_<t>_PWMB port output valid when HRPWM function is in progress	RW
b27	Res	Reserved Bit	Read as "0", write as "0"	RW
b26~b24	EMBSB[2:0]	EMB selection	000: Select EMB event channel 0 to be valid 001: Select EMB event channel 1 to be valid 010: Select EMB event channel 2 to be valid 011: Select EMB event channel 3 to be valid 100: Select EMB event channel 4 to be valid 101: Select EMB event channel 5 to be valid	RW
b23~b22	EMBRB[1:0]	EMB release mode selection B	00: When the EMB event of the selected channel is invalid, the HRPWM_<t>_PWMB port is released immediately (One Shot) 01: When the EMB event of the selected channel is invalid, the HRPWM_<t>_PWMB port is released when the cycle matches (Cycle By Cycle 1) 10: When the EMB event of the selected channel is invalid, the HRPWM_<t>_PWMB port is released when the zero point matches (Cycle By Cycle 2) 11: When the EMB event of the selected channel is invalid, the HRPWM_<t>_PWMB port is released when the cycle matches or the zero point matches (Cycle By Cycle 3)	RW
b21~b20	EMBCB[1:0]	PortB state set EMB event occurs	00: When an EMB event occurs on the selected channel, the HRPWM_<t>_PWMB port outputs normally. 01: When an EMB event occurs on the selected channel, the HRPWM_<t>_PWMB port outputs a high impedance state. 10: When an EMB event occurs on the selected channel, the HRPWM_<t>_PWMB port outputs a low level 11: When an EMB event occurs on the selected channel, the HRPWM_<t>_PWMB port outputs a high level	RW
b19~b18	Res	Reserved Bit	Read as "0", write as "0"	RW
b17~b16	FORCB[1:0]	Forced port status setting B	0x: Set invalid 10: The next cycle starts, and the HRPWM_<t>_PWMB port output is set to a low level 11: The next cycle starts, and the HRPWM_<t>_PWMB port output is set to a high level Note 1: The next cycle refers to the Full period point configured by register HRPWM_<t>_GCONR1.PRDSEL Note 2: This register bit can be used to achieve 0% or 100% control of PWM output duty cycle.	RW
b15~b14	CMBDCB[1:0]	Action When Counter = CMPB on Down Count	00: During down counting, when the timer count value is equal to GCMBR, the HRPWM_<t>_PWMB port output is set to low level 01: During down counting, when the timer count value is equal to GCMBR, the HRPWM_<t>_PWMB port output is set to high level 10: During down counting, when the timer count value is equal to GCMBR, the HRPWM_<t>_PWMB port output is not affected 11: During down counting, when the timer count value is equal to GCMBR, the HRPWM_<t>_PWMB port output is set to the inverted level	RW
b13~b12	CMBUCB[1:0]	Action When Counter = CMPB on Up Count	00: During the up counting period, when the timer count value is equal to GCMBR, the HRPWM_<t>_PWMB port output is set to low level 01: During the up counting period, when the timer count value is equal to GCMBR, the HRPWM_<t>_PWMB port output is set to high level	RW

			10: During the up counting period, when the timer count value is equal to GCMBR, the HRPWM_<t>_PWMB port output is not affected 11: During the up counting period, when the timer count value is equal to GCMBR, the HRPWM_<t>_PWMB port output is set to the inverted level	
b11~b10	CMADCB[1:0]	Action When Counter = CMPA on Down Count	00: During down counting, when the timer count value is equal to GCMAR, the HRPWM_<t>_PWMB port output is set to low level 01: During down counting, when the timer count value is equal to GCMAR, the HRPWM_<t>_PWMB port output is set to high level 10: During down counting, when the timer count value is equal to GCMAR, the HRPWM_<t>_PWMB port output is not affected 11: During down counting, when the timer count value is equal to GCMAR, the HRPWM_<t>_PWMB port output is set to the inverted level	RW
b9~b8	CMAUCB[1:0]	Action When Counter = CMPA on Up Count	00: During the up counting period, when the timer count value is equal to GCMAR, the HRPWM_<t>_PWMB port output is set to low level 01: During the up counting period, when the timer count value is equal to GCMAR, the HRPWM_<t>_PWMB port output is set to high level 10: During the up counting period, when the timer count value is equal to GCMAR, the HRPWM_<t>_PWMB port output is not affected 11: During the up counting period, when the timer count value is equal to GCMAR, the HRPWM_<t>_PWMB port output is set to the inverted level	RW
b7~b6	UDFCB[1:0]	Action When Counter = 0	00: When the count value is equal to 0, the HRPWM_<t>_PWMB port output is set to low level 01: When the count value is equal to 0, the HRPWM_<t>_PWMB port output is set to high level 10: When the count value is equal to 0, the output of the HRPWM_<t>_PWMB port is not affected 11: When the count value is equal to 0, the HRPWM_<t>_PWMB port output is set to the inverted level	RW
b5~b4	OVFCB[1:0]	Action When Counter = HRPERAR	00: When the count value is equal to the period value or the sawtooth wave hardware is cleared, the HRPWM_<t>_PWMB port output is set to a low level 01: When the count value is equal to the period value or the sawtooth wave hardware is cleared, the HRPWM_<t>_PWMB port output is set to a high level 10: When the count value is equal to the period value or the sawtooth wave hardware is cleared, the HRPWM_<t>_PWMB port output is not affected 11: When the count value is equal to the period value or the sawtooth wave hardware is cleared, the HRPWM_<t>_PWMB port output is set to the inverted level	RW
b3~b2	STPCB[1:0]	Action When Counter Stop	00: When counting stops, the HRPWM_<t>_PWMB port output is set to low level 01: When counting stops, the HRPWM_<t>_PWMB port output is set to high level 10: When counting stops, the HRPWM_<t>_PWMB port output maintains the previous state 11: When counting stops, the HRPWM_<t>_PWMB port output maintains the previous state	RW
b1~b0	STBCB[1:0]	Action When Counter Start	00: When counting starts, the HRPWM_<t>_PWMB port output is set to low level 01: When counting starts, the HRPWM_<t>_PWMB port output is set to high level 10: When counting starts, the HRPWM_<t>_PWMB port output maintains the previous state 11: When counting starts, the HRPWM_<t>_PWMB port output maintains the previous state	RW

19.6.1.37 Shadow PortA Control Register A1 (HRPWM<t>_BPCNAR1)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res			OUT	Res	EMBS			EMBR		EMBC		Res		FORC	

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
CMBDC		CMBUC		CMADC		CMAUC		UDFC		OVFC		STPC		STAC	

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b29	Res	Reserved Bit	Read as "0", write as "0"	RW
b28	OUTENA	EMB selection	0: HRPWM_<t>_PWMA port output invalid in HRPWM function 1: HRPWM_<t>_PWMA port output valid when HRPWM function is in progress	RW
b27	Res	Select A for EMB release mode	Read as "0", write as "0"	RW
b26~b24	EMBSA[2:0]	PortA state set EMB event occurs	000: Select EMB event channel 0 to be valid 001: Select EMB event channel 1 to be valid 010: Select EMB event channel 2 to be valid 011: Select EMB event channel 3 to be valid 100: Select EMB event channel 4 to be valid 101: Select EMB event channel 5 to be valid	RW
b23~b22	EMBRA[1:0]	Reserved Bit	00: When the EMB event of the selected channel is invalid, the HRPWM_<t>_PWMA port is released immediately (One Shot) 01: When the EMB event of the selected channel is invalid, the HRPWM_<t>_PWMA port is released when the cycle matches (Cycle By Cycle 1) 10: When the EMB event of the selected channel is invalid, the HRPWM_<t>_PWMA port is released when the zero point matches (Cycle By Cycle 2) 11: When the EMB event of the selected channel is invalid, the HRPWM_<t>_PWMA port is released when the cycle matches or the zero point matches (Cycle By Cycle 3)	RW
b21~b20	EMBCA[1:0]	Force port A status	00: When an EMB event occurs on the selected channel, the HRPWM_<t>_PWMA port outputs normally. 01: When an EMB event occurs on the selected channel, the HRPWM_<t>_PWMA port outputs a high impedance state. 10: When an EMB event occurs on the selected channel, the HRPWM_<t>_PWMA port outputs a low level 11: When an EMB event occurs on the selected channel, the HRPWM_<t>_PWMA port outputs a high level	RW
b19~b18	Res	Action When Counter = SCMBR on Down Count	Read as "0", write as "0"	RW
b17~b16	FORCA[1:0]	Action When Counter = SCMBR on Up Count	0x: Set invalid 10: The next cycle starts, and the HRPWM_<t>_PWMA port output is set to a low level 11: The next cycle starts, and the HRPWM_<t>_PWMA port output is set to a high level Note 1: The next cycle refers to the Full period point configured by register HRPWM_<t>_GCONR1.PRDSEL Note 2: This register bit can be used to achieve 0% or 100% control of PWM output duty cycle.	RW
b15~b14	CMBDCA[1:0]	Action When Counter = CMPB on Down Count	00: During down counting, when the timer count value is equal to GCMBR, the HRPWM_<t>_PWMA port output is set to low level 01: During down counting, when the timer count value is equal to GCMBR, the HRPWM_<t>_PWMA port output is set to high level 10: During down counting, when the timer count value is equal to GCMBR, the HRPWM_<t>_PWMA port output is not affected 11: During down counting, when the timer count value is equal to GCMBR, the HRPWM_<t>_PWMA port output is set to the inverted level	RW
b13~b12	CMBUCA[1:0]	Action When Counter = CMPB on Up Count	00: During the up counting period, when the timer count value is equal to GCMBR, the HRPWM_<t>_PWMA port output is set to low level 01: During the up counting period, when the timer count value is equal to GCMBR, the HRPWM_<t>_PWMA port output is set to high level	RW

			10: During the up counting period, when the timer count value is equal to GCMBR, the HRPWM_<t>_PWMA port output is not affected 11: During the up counting period, when the timer count value is equal to GCMBR, the HRPWM_<t>_PWMA port output is set to the inverted level	
b11~b10	CMADCA[1:0]	Action When Counter = CMPA on Down Count	00: During down counting, when the timer count value is equal to GCMAR, the HRPWM_<t>_PWMA port output is set to low level 01: During down counting, when the timer count value is equal to GCMAR, the HRPWM_<t>_PWMA port output is set to high level 10: During down counting, when the timer count value is equal to GCMAR, the HRPWM_<t>_PWMA port output is not affected 11: During down counting, when the timer count value is equal to GCMAR, the HRPWM_<t>_PWMA port output is set to the inverted level	RW
b9~b8	CMAUCA[1:0]	Action When Counter = CMPA on Up Count	00: During the up counting period, when the timer count value is equal to GCMAR, the HRPWM_<t>_PWMA port output is set to low level 01: During the up counting period, when the timer count value is equal to GCMAR, the HRPWM_<t>_PWMA port output is set to high level 10: During the up counting period, when the timer count value is equal to GCMAR, the HRPWM_<t>_PWMA port output is not affected 11: During the up counting period, when the timer count value is equal to GCMAR, the HRPWM_<t>_PWMA port output is set to the inverted level	RW
b7~b6	UDFCA[1:0]	Action When Counter = 0	00: When the count value is equal to 0, the HRPWM_<t>_PWMA port output is set to low level 01: When the count value is equal to 0, the HRPWM_<t>_PWMA port output is set to high level 10: When the count value is equal to 0, the output of the HRPWM_<t>_PWMA port is not affected 11: When the count value is equal to 0, the HRPWM_<t>_PWMA port output is set to the inverted level	RW
b5~b4	OVFCA[1:0]	Action When Counter = HRPERAR	00: When the count value is equal to the period value or the sawtooth wave hardware is cleared, the HRPWM_<t>_PWMA port output is set to a low level 01: When the count value is equal to the period value or the sawtooth wave hardware is cleared, the HRPWM_<t>_PWMA port output is set to a high level 10: When the count value is equal to the period value or the sawtooth wave hardware is cleared, the HRPWM_<t>_PWMA port output is not affected 11: When the count value is equal to the period value or the sawtooth wave hardware is cleared, the HRPWM_<t>_PWMA port output is set to the inverted level	RW
b3~b2	STPCA[1:0]	Action When Counter Stop	00: When counting stops, the HRPWM_<t>_PWMA port output is set to low level 01: When counting stops, the HRPWM_<t>_PWMA port output is set to high level 10: When counting stops, the HRPWM_<t>_PWMA port output maintains the previous state 11: When counting stops, the HRPWM_<t>_PWMA port output maintains the previous state	RW
b1~b0	STACA[1:0]	Action When Counter Start	00: When counting starts, the HRPWM_<t>_PWMA port output is set to low level 01: When counting starts, the HRPWM_<t>_PWMA port output is set to high level 10: When counting starts, the HRPWM_<t>_PWMA port output maintains the previous state 11: When counting starts, the HRPWM_<t>_PWMA port output maintains the previous state -	RW

19.6.1.38 Shadow PortB Control Buffer Register 1 (HRPWM<t>_BPCNBR1)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res			OUTENB	Res	EMBSB[1:0]			EMBRB[1:0]		EMBCB[1:0]		Res		FORCB[1:0]	
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
CMBDCB[1:0]		CMBUCB[1:0]		CMADCB[1:0]		CMAUCB[1:0]		UDFCB[1:0]		OVFCB[1:0]		STPCB[1:0]		STACB[1:0]	

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b29	Res	Reserved Bit	Read as "0", write as "0"	RW
b28	OUTENB	Output enable B	0: HRPWM_<t>_PWMB port output invalid in HRPWM function 1: HRPWM_<t>_PWMB port output valid when HRPWM function is in progress	RW
b27	Res	Reserved Bit	Read as "0", write as "0"	RW
b26~b24	EMBSB[2:0]	EMB selection	000: Select EMB event channel 0 to be valid 001: Select EMB event channel 1 to be valid 010: Select EMB event channel 2 to be valid 011: Select EMB event channel 3 to be valid 100: Select EMB event channel 4 to be valid 101: Select EMB event channel 5 to be valid	RW
b23~b22	EMBRB[1:0]	EMB release mode selection B	00: When the EMB event of the selected channel is invalid, the HRPWM_<t>_PWMB port is released immediately (One Shot) 01: When the EMB event of the selected channel is invalid, the HRPWM_<t>_PWMB port is released when the cycle matches (Cycle By Cycle 1) 10: When the EMB event of the selected channel is invalid, the HRPWM_<t>_PWMB port is released when the zero point matches (Cycle By Cycle 2) 11: When the EMB event of the selected channel is invalid, the HRPWM_<t>_PWMB port is released when the cycle matches or the zero point matches (Cycle By Cycle 3)	RW
b21~b20	EMBCB[1:0]	PortB state set EMB event occurs	00: When an EMB event occurs on the selected channel, the HRPWM_<t>_PWMB port outputs normally. 01: When an EMB event occurs on the selected channel, the HRPWM_<t>_PWMB port outputs a high impedance state. 10: When an EMB event occurs on the selected channel, the HRPWM_<t>_PWMB port outputs a low level 11: When an EMB event occurs on the selected channel, the HRPWM_<t>_PWMB port outputs a high level	RW
b19~b18	Res	Reserved Bit	Read as "0", write as "0"	RW
b17~b16	FORCB[1:0]	Forced port status setting B	0x: Set invalid 10: The next cycle starts, and the HRPWM_<t>_PWMB port output is set to a low level 11: The next cycle starts, and the HRPWM_<t>_PWMB port output is set to a high level Note 1: The next cycle refers to the Full period point configured by register HRPWM_<t>_GCONR1.PRDSEL Note 2: This register bit can be used to achieve 0% or 100% control of PWM output duty cycle.	RW
b15~b14	CMBDCB[1:0]	Action When Counter = CMPB on Down Count	00: During down counting, when the timer count value is equal to GCMBR, the HRPWM_<t>_PWMB port output is set to low level 01: During down counting, when the timer count value is equal to GCMBR, the HRPWM_<t>_PWMB port output is set to high level 10: During down counting, when the timer count value is equal to GCMBR, the HRPWM_<t>_PWMB port output is not affected 11: During down counting, when the timer count value is equal to GCMBR, the HRPWM_<t>_PWMB port output is set to the inverted level	RW
b13~b12	CMBUCB[1:0]	Action When Counter = CMPB on Up Count	00: During the up counting period, when the timer count value is equal to GCMBR, the HRPWM_<t>_PWMB port output is set to low level 01: During the up counting period, when the timer count value is equal to GCMBR, the HRPWM_<t>_PWMB port output is set to high level 10: During the up counting period, when the timer count value is equal to GCMBR, the HRPWM_<t>_PWMB port output is not affected	RW

			11: During the up counting period, when the timer count value is equal to GCMBR, the HRPWM_<t>_PWMB port output is set to the inverted level	
b11~b10	CMADCB[1:0]	Action When Counter = CMPA on Down Count	00: During down counting, when the timer count value is equal to GCMAR, the HRPWM_<t>_PWMB port output is set to low level 01: During down counting, when the timer count value is equal to GCMAR, the HRPWM_<t>_PWMB port output is set to high level 10: During down counting, when the timer count value is equal to GCMAR, the HRPWM_<t>_PWMB port output is not affected 11: During down counting, when the timer count value is equal to GCMAR, the HRPWM_<t>_PWMB port output is set to the inverted level	RW
b9~b8	CMAUCB[1:0]	Action When Counter = CMPA on Up Count	00: During the up counting period, when the timer count value is equal to GCMAR, the HRPWM_<t>_PWMB port output is set to low level 01: During the up counting period, when the timer count value is equal to GCMAR, the HRPWM_<t>_PWMB port output is set to high level 10: During the up counting period, when the timer count value is equal to GCMAR, the HRPWM_<t>_PWMB port output is not affected 11: During the up counting period, when the timer count value is equal to GCMAR, the HRPWM_<t>_PWMB port output is set to the inverted level	RW
b7~b6	UDFCB[1:0]	Action When Counter = 0	00: When the count value is equal to 0, the HRPWM_<t>_PWMB port output is set to low level 01: When the count value is equal to 0, the HRPWM_<t>_PWMB port output is set to high level 10: When the count value is equal to 0, the output of the HRPWM_<t>_PWMB port is not affected 11: When the count value is equal to 0, the HRPWM_<t>_PWMB port output is set to the inverted level	RW
b5~b4	OVFCB[1:0]	Action When Counter = HRPERAR	00: When the count value is equal to the period value or the sawtooth wave hardware is cleared, the HRPWM_<t>_PWMB port output is set to a low level 01: When the count value is equal to the period value or the sawtooth wave hardware is cleared, the HRPWM_<t>_PWMB port output is set to a high level 10: When the count value is equal to the period value or the sawtooth wave hardware is cleared, the HRPWM_<t>_PWMB port output is not affected 11: When the count value is equal to the period value or the sawtooth wave hardware is cleared, the HRPWM_<t>_PWMB port output is set to the inverted level	RW
b3~b2	STPCB[1:0]	Action When Counter Stop	00: When counting stops, the HRPWM_<t>_PWMB port output is set to low level 01: When counting stops, the HRPWM_<t>_PWMB port output is set to high level 10: When counting stops, the HRPWM_<t>_PWMB port output maintains the previous state 11: When counting stops, the HRPWM_<t>_PWMB port output maintains the previous state	RW
b1~b0	STBCB[1:0]	Action When Counter Start	00: When counting starts, the HRPWM_<t>_PWMB port output is set to low level 01: When counting starts, the HRPWM_<t>_PWMB port output is set to high level 10: When counting starts, the HRPWM_<t>_PWMB port output maintains the previous state 11: When counting starts, the HRPWM_<t>_PWMB port output maintains the previous state	RW

19.6.1.39 Port A Control Register 2 (HRPWM<t>_PCNAR2)

Reset value: 0x000A AAAA

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res				SCMBUCA[1:0]		SCMAUCA[1:0]		CMFUCA[1:0]		CMEUCA[1:0]		EXEV10UCA[1:0]		EXEV9UCA[1:0]	
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
EXEV8UCA[1:0]		EXEV7UCA[1:0]		EXEV6UCA[1:0]		EXEV5UCA[1:0]		EXEV4UCA[1:0]		EXEV3UCA[1:0]		EXEV2UCA[1:0]		EXEV1UCA[1:0]	

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b28	Rese	Reserved Bit	Read as "0", write as "0"	RW
b27~b26	SCMBUCA[1:0]	Action When Counter = SCMBR on Up Count	00: During the up counting period, when the timer count value is equal to SCMBR, the HRPWM_<t>_PWMA port output is set to low level 01: During the up counting period, when the timer count value is equal to SCMBR, the HRPWM_<t>_PWMA port output is set to high level 10: During the up counting period, when the timer count value is equal to SCMBR, the HRPWM_<t>_PWMA port output is not affected 11: During the up counting period, when the timer count value is equal to SCMBR, the HRPWM_<t>_PWMA port output is set to the inverted level	RW
b25~b24	SCMAUCA[1:0]	Action When Counter = SCMAR on Up Count	00: During the up counting period, when the timer count value is equal to SCMAR, the HRPWM_<t>_PWMA port output is set to low level 01: During the up counting period, when the timer count value is equal to SCMAR, the HRPWM_<t>_PWMA port output is set to high level 10: During the up counting period, when the timer count value is equal to SCMAR, the HRPWM_<t>_PWMA port output is not affected 11: During the up counting period, when the timer count value is equal to SCMAR, the HRPWM_<t>_PWMA port output is set to the inverted level	RW
b23~b22	CMFUCA[1:0]	Action When Counter = CMPF on Up Count	00: During the up counting period, when the timer count value is equal to HRGCMFR, the HRPWM_<t>_PWMA port output is set to low level 01: During the up counting period, when the timer count value is equal to HRGCMFR, the HRPWM_<t>_PWMA port output is set to high level 10: During the up counting period, when the timer count value is equal to HRGCMFR, the HRPWM_<t>_PWMA port output is not affected 11: During the up counting period, when the timer count value is equal to HRGCMFR, the HRPWM_<t>_PWMA port output is set to the inverted level	RW
b21~b20	CMEUCA[1:0]	Action When Counter = CMPE on Up Count	00: During the up counting period, when the timer count value is equal to HRGCMER, the HRPWM_<t>_PWMA port output is set to low level 01: During the up counting period, when the timer count value is equal to HRGCMER, the HRPWM_<t>_PWMA port output is set to high level 10: During the up counting period, when the timer count value is equal to HRGCMER, the HRPWM_<t>_PWMA port output is not affected 11: During the up counting period, when the timer count value is equal to HRGCMER, the HRPWM_<t>_PWMA port output is set to the inverted level	RW
b19~b18	EXEV10UCA[1:0]	Action When Count up && external event 10 occurs	Refer to EXEV1CA[1:0] description	RW
b17~b16	EXEV9UCA[1:0]	Action When Action When Count up && external event 9 occurs	Refer to EXEV1UCA[1:0] description	RW
b15~b14	EXEV8UCA[1:0]	Action When Count up && external event 8 occurs	Refer to EXEV1UCA[1:0] description	RW
b13~b12	EXEV7UCA[1:0]	Action When Count up && external event 7 occurs	Refer to EXEV1UCA[1:0] description	RW

b11~b10	EXEV6UCA[1:0]	Action When Count up && external event 6 occurs	Refer to EXEV1UCA[1:0] description	RW
b9~b8	EXEV5UCA[1:0]	Action When Count up && external event 5 occurs	Refer to EXEV1UCA[1:0] description	RW
b7~b6	EXEV4UCA[1:0]	Action When Count up && external event 4 occurs	Refer to EXEV1UCA[1:0] description	RW
b5~b4	EXEV3UCA[1:0]	Action When Count up && external event 3 occurs	Refer to EXEV1UCA[1:0] description	RW
b3~b2	EXEV2UCA[1:0]	Action When Count up && external event 2 occurs	Refer to EXEV1UCA[1:0] description	RW
b1~b0	EXEV1UCA[1:0]	Action When Count up && external event 1 occurs	00: During the up counting period, when external event 1 occurs, the HRPWM_<t>_PWMA port output is set to low level 01: During the up counting period, when external event 1 occurs, the HRPWM_<t>_PWMA port output is set to high level 10: During the up counting period, when external event 1 occurs, the output of the HRPWM_<t>_PWMA port is not affected 11: During the up-counting period, when external event 1 occurs, the HRPWM_<t>_PWMA port outputs an inverted level	RW

19.6.1.40 Port B Control Register 2 (HRPWM<t>_PCNBR2)

Reset value: 0x000A AAAA

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res				SCMBUCB[1:0]		SCMAUCB[1:0]		CMFUCB[1:0]		CMEUCB[1:0]		EXEV10UCB [1:0]		EXEV9UCB [1:0]	
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
EXEV8UCB [1:0]		EXEV7UCB [1:0]		EXEV6UCB [1:0]		EXEV5UCB [1:0]		EXEV4UCB [1:0]		EXEV3UCB [1:0]		EXEV2UCB [1:0]		EXEV1UCB [1:0]	

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b28	Res	Reserved Bit	Read as "0", write as "0"	RW
b27~b26	SCMBUCB[1:0]	Action When Counter = SCMBR on Up Count	00: During the up counting period, when the timer count value is equal to SCMBR, the HRPWM_<t>_PWMB port output is set to low level 01: During the up counting period, when the timer count value is equal to SCMBR, the HRPWM_<t>_PWMB port output is set to high level 10: During the up counting period, when the timer count value is equal to SCMBR, the HRPWM_<t>_PWMB port output is not affected 11: During the up counting period, when the timer count value is equal to SCMBR, the HRPWM_<t>_PWMB port output is set to the inverted level	RW
b25~b24	SCMAUCB[1:0]	Action When Counter = SCMAR on Up Count	00: During the up counting period, when the timer count value is equal to SCMAR, the HRPWM_<t>_PWMB port output is set to low level 01: During the up counting period, when the timer count value is equal to SCMAR, the HRPWM_<t>_PWMB port output is set to high level 10: During the up counting period, when the timer count value is equal to SCMAR, the HRPWM_<t>_PWMB port output is not affected 11: During the up counting period, when the timer count value is equal to SCMAR, the HRPWM_<t>_PWMB port output is set to the inverted level	RW
b23~b22	CMFUCB[1:0]	Action When Counter = CMPF on Up Count	00: During the up counting period, when the timer count value is equal to HRGCMFR, the HRPWM_<t>_PWMB port output is set to low level 01: During the up counting period, when the timer count value is equal to HRGCMFR, the HRPWM_<t>_PWMB port output is set to high level 10: During the up counting period, when the timer count value is equal to HRGCMFR, the HRPWM_<t>_PWMB port output is not affected 11: During the up counting period, when the timer count value is equal to HRGCMFR, the HRPWM_<t>_PWMB port output is set to the inverted level	RW
b21~b20	CMEUCB[1:0]	Action When Counter = CMPE on Up Count	00: During the up counting period, when the timer count value is equal to HRGCMER, the HRPWM_<t>_PWMB port output is set to low level 01: During the up counting period, when the timer count value is equal to HRGCMER, the HRPWM_<t>_PWMB port output is set to high level 10: During the up counting period, when the timer count value is equal to HRGCMER, the HRPWM_<t>_PWMB port output is not affected 11: During the up counting period, when the timer count value is equal to HRGCMER, the HRPWM_<t>_PWMB port output is set to the inverted level	RW
b19~b18	EXEV10UCB[1:0]	Action When Count up && external event 10 occurs	Refer to EXEV1CB[1:0] description	RW
b17~b16	EXEV9UCB[1:0]	Action When Action When Count up && external event 9 occurs	Refer to EXEV1CB[1:0] description	RW
b15~b14	EXEV8UCB[1:0]	Action When Count up && external event 8 occurs	Refer to EXEV1CB[1:0] description	RW
b13~b12	EXEV7UCB[1:0]	Action When Count up && external event 7 occurs	Refer to EXEV1CB[1:0] description	RW

b11~b10	EXEV6UCB[1:0]	Action When Count up && external event 6 occurs	Refer to EXEV1CB[1:0] description	RW
b9~b8	EXEV5UCB[1:0]	Action When Count up && external event 5 occurs	Refer to EXEV1CB[1:0] description	RW
b7~b6	EXEV4UCB[1:0]	Action When Count up && external event 4 occurs	Refer to EXEV1CB[1:0] description	RW
b5~b4	EXEV3UCB[1:0]	Action When Count up && external event 3 occurs	Refer to EXEV1CB[1:0] description	RW
b3~b2	EXEV2UCB[1:0]	Action When Count up && external event 2 occurs	Refer to EXEV1CB[1:0] description	RW
b1~b0	EXEV1UCB[1:0]	Action When Count up && external event 1 occurs	00: During the up counting period, when external event 1 occurs, the HRPWM_<t>_PWMB port output is set to low level 01: During the up counting period, when external event 1 occurs, the HRPWM_<t>_PWMB port output is set to high level 10: During the up counting period, when external event 1 occurs, the output of the HRPWM_<t>_PWMB port is not affected 11: During the up-counting period, when external event 1 occurs, the HRPWM_<t>_PWMB port outputs an inverted level	RW

19.6.1.41 Shadow Port Control Register A2 (HRPWM<t>_BPCNAR2)

Reset value: 0x000A AAAA

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res				SCMBUCA[1:0]		SCMAUCA[1:0]		CMFUCA[1:0]		CMEUCA[1:0]		EXEV10UCA[1:0]		EXEV9UCA[1:0]	
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
EXEV8UCA[1:0]		EXEV7UCA[1:0]		EXEV6UCA[1:0]		EXEV5UCA[1:0]		EXEV4UCA[1:0]		EXEV3UCA[1:0]		EXEV2UCA[1:0]		EXEV1UCA[1:0]	

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b28	Res	Reserved Bit	Read as "0", write as "0"	RW
b27~b26	SCMBUCA[1:0]	Action When Counter = SCMBR on Up Count	00: During the up counting period, when the timer count value is equal to SCMBR, the HRPWM_<t>_PWMA port output is set to low level 01: During the up counting period, when the timer count value is equal to SCMBR, the HRPWM_<t>_PWMA port output is set to high level 10: During the up counting period, when the timer count value is equal to SCMBR, the HRPWM_<t>_PWMA port output is not affected 11: During the up counting period, when the timer count value is equal to SCMBR, the HRPWM_<t>_PWMA port output is set to the inverted level	RW
b25~b24	SCMAUCA[1:0]	Action When Counter = SCMAR on Up Count	00: During the up counting period, when the timer count value is equal to SCMAR, the HRPWM_<t>_PWMA port output is set to low level 01: During the up counting period, when the timer count value is equal to SCMAR, the HRPWM_<t>_PWMA port output is set to high level 10: During the up counting period, when the timer count value is equal to SCMAR, the HRPWM_<t>_PWMA port output is not affected 11: During the up counting period, when the timer count value is equal to SCMAR, the HRPWM_<t>_PWMA port output is set to the inverted level	RW
b23~b22	CMFUCA[1:0]	Action When Counter = CMPF on Up Count	00: During the up counting period, when the timer count value is equal to HRGCMFR, the HRPWM_<t>_PWMA port output is set to low level 01: During the up counting period, when the timer count value is equal to HRGCMFR, the HRPWM_<t>_PWMA port output is set to high level 10: During the up counting period, when the timer count value is equal to HRGCMFR, the HRPWM_<t>_PWMA port output is not affected 11: During the up counting period, when the timer count value is equal to HRGCMFR, the HRPWM_<t>_PWMA port output is set to the inverted level	RW
b21~b20	CMEUCA[1:0]	Action When Counter = CMPE on Up Count	00: During the up counting period, when the timer count value is equal to HRGCMER, the HRPWM_<t>_PWMA port output is set to low level 01: During the up counting period, when the timer count value is equal to HRGCMER, the HRPWM_<t>_PWMA port output is set to high level 10: During the up counting period, when the timer count value is equal to HRGCMER, the HRPWM_<t>_PWMA port output is not affected 11: During the up counting period, when the timer count value is equal to HRGCMER, the HRPWM_<t>_PWMA port output is set to the inverted level	RW
b19~b18	EXEV10UCA[1:0]	Action When Count up && external event 10 occurs	Refer to EXEV1CA[1:0] description	RW
b17~b16	EXEV9UCA[1:0]	Action When Action When Count up && external event 9 occurs	Refer to EXEV1UCA[1:0] description	RW
b15~b14	EXEV8UCA[1:0]	Action When Count up && external event 8 occurs	Refer to EXEV1UCA[1:0] description	RW
b13~b12	EXEV7UCA[1:0]	Action When Count up && external event 7 occurs	Refer to EXEV1UCA[1:0] description	RW
b11~b10	EXEV6UCA[1:0]	Action When Count up && external event 6 occurs	Refer to EXEV1UCA[1:0] description	RW
b9~b8	EXEV5UCA[1:0]	Action When Count up && external event 5 occurs	Refer to EXEV1UCA[1:0] description	RW

b7~b6	EXEV4UCA[1:0]	Action When Count up && external event 4 occurs	Refer to EXEV1UCA[1:0] description	RW
b5~b4	EXEV3UCA[1:0]	Action When Count up && external event 3 occurs	Refer to EXEV1UCA[1:0] description	RW
b3~b2	EXEV2UCA[1:0]	Action When Count up && external event 2 occurs	Refer to EXEV1UCA[1:0] description	RW
b1~b0	EXEV1UCA[1:0]	Action When Count up && external event 1 occurs	00: During the up counting period, when external event 1 occurs, the HRPWM_<t>_PWMA port output is set to low level 01: During the up counting period, when external event 1 occurs, the HRPWM_<t>_PWMA port output is set to high level 10: During the up counting period, when external event 1 occurs, the output of the HRPWM_<t>_PWMA port is not affected 11: During the up-counting period, when external event 1 occurs, the HRPWM_<t>_PWMA port outputs an inverted level	RW

19.6.1.42 Shadow Port B Control Register B2 (HRPWM<t>_BPCNBR2)

Reset value: 0x000A AAAA

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res				SCMBDCB[1:0]		SCMADCB[1:0]		CMFUCB[1:0]		CMEUCB[1:0]		EXEV10UCB[1:0]		EXEV9UCB[1:0]	
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
EXEV8UCB[1:0]		EXEV7UCB[1:0]		EXEV6UCB[1:0]		EXEV5UCB[1:0]		EXEV4UCB[1:0]		EXEV3UCB[1:0]		EXEV2UCB[1:0]		EXEV1UCB[1:0]	

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b28	Res	Reserved Bit	Read as "0", write as "0"	RW
b27~b26	SCMBUCB[1:0]	Action When Counter = SCMBR on Up Count	00: During the up counting period, when the timer count value is equal to SCMBR, the HRPWM_<t>_PWMB port output is set to low level 01: During the up counting period, when the timer count value is equal to SCMBR, the HRPWM_<t>_PWMB port output is set to high level 10: During the up counting period, when the timer count value is equal to SCMBR, the HRPWM_<t>_PWMB port output is not affected 11: During the up counting period, when the timer count value is equal to SCMBR, the HRPWM_<t>_PWMB port output is set to the inverted level	RW
b25~b24	SCMAUCB[1:0]	Action When Counter = SCMAR on Up Count	00: During the up counting period, when the timer count value is equal to SCMAR, the HRPWM_<t>_PWMB port output is set to low level 01: During the up counting period, when the timer count value is equal to SCMAR, the HRPWM_<t>_PWMB port output is set to high level 10: During the up counting period, when the timer count value is equal to SCMAR, the HRPWM_<t>_PWMB port output is not affected 11: During the up counting period, when the timer count value is equal to SCMAR, the HRPWM_<t>_PWMB port output is set to the inverted level	RW
b23~b22	CMFUCB[1:0]	Action When Counter = CMPF on Up Count	00: During the up counting period, when the timer count value is equal to HRGCMFR, the HRPWM_<t>_PWMB port output is set to low level 01: During the up counting period, when the timer count value is equal to HRGCMFR, the HRPWM_<t>_PWMB port output is set to high level 10: During the up counting period, when the timer count value is equal to HRGCMFR, the HRPWM_<t>_PWMB port output is not affected 11: During the up counting period, when the timer count value is equal to HRGCMFR, the HRPWM_<t>_PWMB port output is set to the inverted level	RW
b21~b20	CMEUCB[1:0]	Action When Counter = CMPE on Up Count	00: During the up counting period, when the timer count value is equal to HRGCMER, the HRPWM_<t>_PWMB port output is set to low level 01: During the up counting period, when the timer count value is equal to HRGCMER, the HRPWM_<t>_PWMB port output is set to high level 10: During the up counting period, when the timer count value is equal to HRGCMER, the HRPWM_<t>_PWMB port output is not affected 11: During the up counting period, when the timer count value is equal to HRGCMER, the HRPWM_<t>_PWMB port output is set to the inverted level	RW
b19~b18	EXEV10UCB[1:0]	Action When Count up && external event 10 occurs	Refer to EXEV1CB[1:0] description	RW
b17~b16	EXEV9UCB[1:0]	Action When Action When Count up && external event 9 occurs	Refer to EXEV1CB[1:0] description	RW
b15~b14	EXEV8UCB[1:0]	Action When Count up && external event 8 occurs	Refer to EXEV1CB[1:0] description	RW
b13~b12	EXEV7UCB[1:0]	Action When Count up && external event 7 occurs	Refer to EXEV1CB[1:0] description	RW

b11~b10	EXEV6UCB[1:0]	Action When Count up && external event 6 occurs	Refer to EXEV1CB[1:0] description	RW
b9~b8	EXEV5UCB[1:0]	Action When Count up && external event 5 occurs	Refer to EXEV1CB[1:0] description	RW
b7~b6	EXEV4UCB[1:0]	Action When Count up && external event 4 occurs	Refer to EXEV1CB[1:0] description	RW
b5~b4	EXEV3UCB[1:0]	Action When Count up && external event 3 occurs	Refer to EXEV1CB[1:0] description	RW
b3~b2	EXEV2UCB[1:0]	Action When Count up && external event 2 occurs	Refer to EXEV1CB[1:0] description	RW
b1~b0	EXEV1UCB[1:0]	Action When Count up && external event 1 occurs	00: During the up counting period, when external event 1 occurs, the HRPWM_<t>_PWMB port output is set to low level 01: During the up counting period, when external event 1 occurs, the HRPWM_<t>_PWMB port output is set to high level 10: During the up counting period, when external event 1 occurs, the output of the HRPWM_<t>_PWMB port is not affected 11: During the up-counting period, when external event 1 occurs, the HRPWM_<t>_PWMB port outputs an inverted level	RW

19.6.1.43 Port A Control Register3 (HRPWM<t>_PCNAR3)

Reset value: 0x000A AAAA

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res				SCMBDCA[1:0]		SCMADCA[1:0]		CMFDCA[1:0]		CMEDCA[1:0]		EXEV10DCA[1:0]		EXEV9DCA[1:0]	
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
EXEV8DCA[1:0]		EXEV7DCA[1:0]		EXEV6DCA[1:0]		EXEV5DCA[1:0]		EXEV4DCA[1:0]		EXEV3DCA[1:0]		EXEV2DCA[1:0]		EXEV1DCA[1:0]	

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b28	Res	Reserved Bit	Read as "0", write as "0"	RW
b27~b26	SCMBDCA[1:0]	Action When Counter = SCMBR on Down Count	00: During down counting, when the timer count value is equal to SCMBR, the HRPWM_<t>_PWMA port output is set to low level 01: During down counting, when the timer count value is equal to SCMBR, the HRPWM_<t>_PWMA port output is set to high level 10: During down counting, when the timer count value is equal to SCMBR, the HRPWM_<t>_PWMA port output is not affected 11: During down counting, when the timer count value is equal to SCMBR, the HRPWM_<t>_PWMA port output is set to the inverted level	RW
b25~b24	SCMADCA[1:0]	Action When Counter = SCMAR on Down Count	00: During down counting, when the timer count value is equal to SCMAR, the HRPWM_<t>_PWMA port output is set to low level 01: During down counting, when the timer count value is equal to SCMAR, the HRPWM_<t>_PWMA port output is set to high level 10: During down counting, when the timer count value is equal to SCMAR, the HRPWM_<t>_PWMA port output is not affected 11: During down counting, when the timer count value is equal to SCMAR, the HRPWM_<t>_PWMA port output is set to the inverted level	RW
b23~b22	CMFDCA[1:0]	Action When Counter = CMPF on Down Count	00: During down counting, when the timer count value is equal to HRGCMFR, the HRPWM_<t>_PWMA port output is set to low level 01: During down counting, when the timer count value is equal to HRGCMFR, the HRPWM_<t>_PWMA port output is set to high level 10: During down counting, when the timer count value is equal to HRGCMFR, the HRPWM_<t>_PWMA port output is not affected 11: During down counting, when the timer count value is equal to HRGCMFR, the HRPWM_<t>_PWMA port output is set to the inverted level	RW
b21~b20	CMEDCA[1:0]	Action When Counter = CMPE on Down Count	00: During down counting, when the timer count value is equal to HRGCMER, the HRPWM_<t>_PWMA port output is set to low level 01: During down counting, when the timer count value is equal to HRGCMER, the HRPWM_<t>_PWMA port output is set to high level 10: During down counting, when the timer count value is equal to HRGCMER, the HRPWM_<t>_PWMA port output is not affected 11: During down counting, when the timer count value is equal to HRGCMER, the HRPWM_<t>_PWMA port output is set to the inverted level	RW
b19~b18	EXEV10DCA[1:0]	Action When Count Down && external event 10 occurs	Refer to EXEV1DCA[1:0] description	RW
b17~b16	EXEV9DCA[1:0]	Action When Action When Count Down && external event 9 occurs	Refer to EXEV1DCA[1:0] description	RW
b15~b14	EXEV8DCA[1:0]	Action When Count Down && external event 8 occurs	Refer to EXEV1DCA[1:0] description	RW
b13~b12	EXEV7DCA[1:0]	Action When Count Down && external event 7 occurs	Refer to EXEV1DCA[1:0] description	RW
b11~b10	EXEV6DCA[1:0]	Action When Count Down && external	Refer to EXEV1DCA[1:0] description	RW

		event 6 occurs		
b9~b8	EXEV5DCA[1:0]	Action When Count Down && external event 5 occurs	Refer to EXEV1DCA[1:0] description	RW
b7~b6	EXEV4DCA[1:0]	Action When Count Down && external event 4 occurs	Refer to EXEV1DCA[1:0] description	RW
b5~b4	EXEV3DCA[1:0]	Action When Count Down && external event 3 occurs	Refer to EXEV1DCA[1:0] description	RW
b3~b2	EXEV2DCA[1:0]	Action When Count Down && external event 2 occurs	Refer to EXEV1DCA[1:0] description	RW
b1~b0	EXEV1DCA[1:0]	Action When Count Down && external event 1 occurs	00: During down counting, when external event 1 occurs, the HRPWM_<t>_PWMA port output is set to low level 01: During down counting, when external event 1 occurs, the HRPWM_<t>_PWMA port output is set to high level 10: During down counting, when external event 1 occurs, the output of HRPWM_<t>_PWMA port is not affected 11: During down counting, when external event 1 occurs, the HRPWM_<t>_PWMA port outputs an inverted level	RW

19.6.1.44 Port B Control Register 3 (HRPWM<t>_PCNBR3)

Reset value: 0x000A AAAA

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved				SCMBDCB[1:0]		SCMADCB[1:0]		CMFDCB[1:0]		CMEDCB[1:0]		EXEV10DCA[1:0]		EXEV9DCA[1:0]	
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
EXEV8DCB[1:0]		EXEV7DCB[1:0]		EXEV6DCB[1:0]		EXEV5DCB[1:0]		EXEV4DCB[1:0]		EXEV3DCB[1:0]		EXEV2DCB[1:0]		EXEV1DCB[1:0]	

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b28	Rese	Reserved Bit	Read as "0", write as "0"	RW
b27~b26	SCMBDCB[1:0]	Action When Counter = SCMBR on Down Count	00: During down counting, when the timer count value is equal to SCMBR, the HRPWM_<t>_PWMB port output is set to low level 01: During down counting, when the timer count value is equal to SCMBR, the HRPWM_<t>_PWMB port output is set to high level 10: During down counting, when the timer count value is equal to SCMBR, the HRPWM_<t>_PWMB port output is not affected 11: During down counting, when the timer count value is equal to SCMBR, the HRPWM_<t>_PWMB port output is set to the inverted level	RW
b25~b24	SCMADCB[1:0]	Action When Counter = SCMAR on Down Count	00: During down counting, when the timer count value is equal to SCMAR, the HRPWM_<t>_PWMB port output is set to low level 01: During down counting, when the timer count value is equal to SCMAR, the HRPWM_<t>_PWMB port output is set to high level 10: During down counting, when the timer count value is equal to SCMAR, the HRPWM_<t>_PWMB port output is not affected 11: During down counting, when the timer count value is equal to SCMAR, the HRPWM_<t>_PWMB port output is set to the inverted level	RW
b23~b22	CMFDCB[1:0]	Action When Counter = CMPF on Down Count	00: During down counting, when the timer count value is equal to HRGCMFR, the HRPWM_<t>_PWMB port output is set to low level 01: During down counting, when the timer count value is equal to HRGCMFR, the HRPWM_<t>_PWMB port output is set to high level 10: During down counting, when the timer count value is equal to HRGCMFR, the HRPWM_<t>_PWMB port output is not affected 11: During down counting, when the timer count value is equal to HRGCMFR, the HRPWM_<t>_PWMB port output is set to the inverted level	RW
b21~b20	CMEDCB[1:0]	Action When Counter = CMPE on Down Count	00: During down counting, when the timer count value is equal to HRGCMER, the HRPWM_<t>_PWMB port output is set to low level 01: During down counting, when the timer count value is equal to HRGCMER, the HRPWM_<t>_PWMB port output is set to high level 10: During down counting, when the timer count value is equal to HRGCMER, the HRPWM_<t>_PWMB port output is not affected 11: During down counting, when the timer count value is equal to HRGCMER, the HRPWM_<t>_PWMB port output is set to the inverted level	RW
b19~b18	EXEV10DCB[1:0]	Action When Count Down && external event 10 occurs	Refer to EXEV1DCB[1:0] description	RW
b17~b16	EXEV9DCB[1:0]	Action When Action When Count Down && external event 9 occurs	Refer to EXEV1DCB[1:0] description	RW
b15~b14	EXEV8DCB[1:0]	Action When Count Down && external event 8 occurs	Refer to EXEV1DCB[1:0] description	RW
b13~b12	EXEV7DCB[1:0]	Action When Count Down && external event 7 occurs	Refer to EXEV1DCB[1:0] description	RW
b11~b10	EXEV6DCB[1:0]	Action When Count Down && external	Refer to EXEV1DCB[1:0] description	RW

		event 6 occurs		
b9~b8	EXEV5DCB[1:0]	Action When Count Down && external event 5 occurs	Refer to EXEV1DCB[1:0] description	RW
b7~b6	EXEV4DCB[1:0]	Action When Count Down && external event 4 occurs	Refer to EXEV1DCB[1:0] description	RW
b5~b4	EXEV3DCB[1:0]	Action When Count Down && external event 3 occurs	Refer to EXEV1DCB[1:0] description	RW
b3~b2	EXEV2DCB[1:0]	Action When Count Down && external event 2 occurs	Refer to EXEV1DCB[1:0] description	RW
b1~b0	EXEV1DCB[1:0]	Action When Count Down && external event 1 occurs	00: During down counting, when external event 1 occurs, the HRPWM_<t>_PWMB port output is set to low level 01: During down counting, when external event 1 occurs, the HRPWM_<t>_PWMB port output is set to high level 10: During down counting, when external event 1 occurs, the output of HRPWM_<t>_PWMB port is not affected 11: During down counting, when external event 1 occurs, the HRPWM_<t>_PWMB port outputs an inverted level	RW

19.6.1.45 Shadow Port Control Register A3 (HRPWM<t>_BPCNAR3)

Reset value: 0x000A AAAA

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res				SCMBDCA[1:0]		SCMADCA[1:0]		CMFDCA[1:0]		CMEDCA[1:0]		EXEV10DCA[1:0]		EXEV9DCA[1:0]	
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
EXEV8DCA[1:0]		EXEV7DCA[1:0]		EXEV6DCA[1:0]		EXEV5DCA[1:0]		EXEV4DCA[1:0]		EXEV3DCA[1:0]		EXEV2DCA[1:0]		EXEV1DCA[1:0]	

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b28	Res	Reserved Bit	Read as "0", write as "0"	RW
b27~b26	SCMBDCA[1:0]	Action When Counter = SCMBR on Down Count	00: During down counting, when the timer count value is equal to SCMBR, the HRPWM_<t>_PWMA port output is set to low level 01: During down counting, when the timer count value is equal to SCMBR, the HRPWM_<t>_PWMA port output is set to high level 10: During down counting, when the timer count value is equal to SCMBR, the HRPWM_<t>_PWMA port output is not affected 11: During down counting, when the timer count value is equal to SCMBR, the HRPWM_<t>_PWMA port output is set to the inverted level	RW
b25~b24	SCMADCA[1:0]	Action When Counter = SCMAR on Down Count	00: During down counting, when the timer count value is equal to SCMAR, the HRPWM_<t>_PWMA port output is set to low level 01: During down counting, when the timer count value is equal to SCMAR, the HRPWM_<t>_PWMA port output is set to high level 10: During down counting, when the timer count value is equal to SCMAR, the HRPWM_<t>_PWMA port output is not affected 11: During down counting, when the timer count value is equal to SCMAR, the HRPWM_<t>_PWMA port output is set to the inverted level	RW
b23~b22	CMFDCA[1:0]	Action When Counter = CMPF on Down Count	00: During down counting, when the timer count value is equal to HRGCMFR, the HRPWM_<t>_PWMA port output is set to low level 01: During down counting, when the timer count value is equal to HRGCMFR, the HRPWM_<t>_PWMA port output is set to high level 10: During down counting, when the timer count value is equal to HRGCMFR, the HRPWM_<t>_PWMA port output is not affected 11: During down counting, when the timer count value is equal to HRGCMFR, the HRPWM_<t>_PWMA port output is set to the inverted level	RW
b21~b20	CMEDCA[1:0]	Action When Counter = CMPE on Down Count	00: During down counting, when the timer count value is equal to HRGCMER, the HRPWM_<t>_PWMA port output is set to low level 01: During down counting, when the timer count value is equal to HRGCMER, the HRPWM_<t>_PWMA port output is set to high level 10: During down counting, when the timer count value is equal to HRGCMER, the HRPWM_<t>_PWMA port output is not affected 11: During down counting, when the timer count value is equal to HRGCMER, the HRPWM_<t>_PWMA port output is set to the inverted level	RW
b19~b18	EXEV10DCA[1:0]	Action When Count Down && external event 10 occurs	Refer to EXEV1DCA[1:0] description	RW
b17~b16	EXEV9DCA[1:0]	Action When Action When Count Down && external event 9 occurs	Refer to EXEV1DCA[1:0] description	RW
b15~b14	EXEV8DCA[1:0]	Action When Count Down && external event 8 occurs	Refer to EXEV1DCA[1:0] description	RW
b13~b12	EXEV7DCA[1:0]	Action When Count Down && external event 7 occurs	Refer to EXEV1DCA[1:0] description	RW
b11~b10	EXEV6DCA[1:0]	Action When Count Down && external	Refer to EXEV1DCA[1:0] description	RW

		event 6 occurs		
b9~b8	EXEV5DCA[1:0]	Action When Count Down && external event 5 occurs	Refer to EXEV1DCA[1:0] description	RW
b7~b6	EXEV4DCA[1:0]	Action When Count Down && external event 4 occurs	Refer to EXEV1DCA[1:0] description	RW
b5~b4	EXEV3DCA[1:0]	Action When Count Down && external event 3 occurs	Refer to EXEV1DCA[1:0] description	RW
b3~b2	EXEV2DCA[1:0]	Action When Count Down && external event 2 occurs	Refer to EXEV1DCA[1:0] description	RW
b1~b0	EXEV1DCA[1:0]	Action When Count Down && external event 1 occurs	00: During down counting, when external event 1 occurs, the HRPWM_<t>_PWMA port output is set to low level 01: During down counting, when external event 1 occurs, the HRPWM_<t>_PWMA port output is set to high level 10: During down counting, when external event 1 occurs, the output of HRPWM_<t>_PWMA port is not affected 11: During down counting, when external event 1 occurs, the HRPWM_<t>_PWMA port outputs an inverted level	RW

19.6.1.46 Shadow Port B Control Buffer Register 3 (HRPWM<t>_BPCNBR3)

Reset value: 0x000A AAAA

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res				SCMBDCB[1:0]		SCMADCB[1:0]		CMFDCB[1:0]		CMEDCB[1:0]		EXEV10DCA[1:0]		EXEV9DCA[1:0]	
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
EXEV8DCB[1:0]		EXEV7DCB[1:0]		EXEV6DCB[1:0]		EXEV5DCB[1:0]		EXEV4DCB[1:0]		EXEV3DCB[1:0]		EXEV2DCB[1:0]		EXEV1DCB[1:0]	

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b28	Res	Reserved Bit	Read as "0", write as "0"	RW
b27~b26	SCMBDCB[1:0]	Action When Counter = SCMBR on Down Count	00: During down counting, when the timer count value is equal to SCMBR, the HRPWM_<t>_PWMB port output is set to low level 01: During down counting, when the timer count value is equal to SCMBR, the HRPWM_<t>_PWMB port output is set to high level 10: During down counting, when the timer count value is equal to SCMBR, the HRPWM_<t>_PWMB port output is not affected 11: During down counting, when the timer count value is equal to SCMBR, the HRPWM_<t>_PWMB port output is set to the inverted level	RW
b25~b24	SCMADCB[1:0]	Action When Counter = SCMAR on Down Count	00: During down counting, when the timer count value is equal to SCMAR, the HRPWM_<t>_PWMB port output is set to low level 01: During down counting, when the timer count value is equal to SCMAR, the HRPWM_<t>_PWMB port output is set to high level 10: During down counting, when the timer count value is equal to SCMAR, the HRPWM_<t>_PWMB port output is not affected 11: During down counting, when the timer count value is equal to SCMAR, the HRPWM_<t>_PWMB port output is set to the inverted level	RW
b23~b22	CMFDCB[1:0]	Action When Counter = CMPF on Down Count	00: During down counting, when the timer count value is equal to HRGCMFR, the HRPWM_<t>_PWMB port output is set to low level 01: During down counting, when the timer count value is equal to HRGCMFR, the HRPWM_<t>_PWMB port output is set to high level 10: During down counting, when the timer count value is equal to HRGCMFR, the HRPWM_<t>_PWMB port output is not affected 11: During down counting, when the timer count value is equal to HRGCMFR, the HRPWM_<t>_PWMB port output is set to the inverted level	RW
b21~b20	CMEDCB[1:0]	Action When Counter = CMPE on Down Count	00: During down counting, when the timer count value is equal to HRGCMER, the HRPWM_<t>_PWMB port output is set to low level 01: During down counting, when the timer count value is equal to HRGCMER, the HRPWM_<t>_PWMB port output is set to high level 10: During down counting, when the timer count value is equal to HRGCMER, the HRPWM_<t>_PWMB port output is not affected 11: During down counting, when the timer count value is equal to HRGCMER, the HRPWM_<t>_PWMB port output is set to the inverted level	RW
b19~b18	EXEV10DCB[1:0]	Action When Count Down && external event 10 occurs	Refer to EXEV1DCB[1:0] description	RW
b17~b16	EXEV9DCB[1:0]	Action When Action When Count Down && external event 9 occurs	Refer to EXEV1DCB[1:0] description	RW
b15~b14	EXEV8DCB[1:0]	Action When Count Down && external event 8 occurs	Refer to EXEV1DCB[1:0] description	RW
b13~b12	EXEV7DCB[1:0]	Action When Count Down && external event 7 occurs	Refer to EXEV1DCB[1:0] description	RW
b11~b10	EXEV6DCB[1:0]	Action When Count Down && external	Refer to EXEV1DCB[1:0] description	RW

		event 6 occurs		
b9~b8	EXEV5DCB[1:0]	Action When Count Down && external event 5 occurs	Refer to EXEV1DCB[1:0] description	RW
b7~b6	EXEV4DCB[1:0]	Action When Count Down && external event 4 occurs	Refer to EXEV1DCB[1:0] description	RW
b5~b4	EXEV3DCB[1:0]	Action When Count Down && external event 3 occurs	Refer to EXEV1DCB[1:0] description	RW
b3~b2	EXEV2DCB[1:0]	Action When Count Down && external event 2 occurs	Refer to EXEV1DCB[1:0] description	RW
b1~b0	EXEV1DCB[1:0]	Action When Count Down && external event 1 occurs	00: During down counting, when external event 1 occurs, the HRPWM_<t>_PWMB port output is set to low level 01: During down counting, when external event 1 occurs, the HRPWM_<t>_PWMB port output is set to high level 10: During down counting, when external event 1 occurs, the output of HRPWM_<t>_PWMB port is not affected 11: During down counting, when external event 1 occurs, the HRPWM_<t>_PWMB port outputs an inverted level	RW

19.6.1.47 Special-Compare A Select Register (HRPWM<t>_SCMASELR)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res							EEVDET	UDF6	OVF6	CMSB6	CMSA6	UDF5	OVF5	CMSB5	CMSA5
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
UDF4	OVF4	CMSB4	CMSA4	UDF3	OVF3	CMSB3	CMSA3	UDF2	OVF2	CMSB2	CMSA2	UDF1	OVF1	CMSB1	CMSA1

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b25	Res	Reserved Bit	Read as "0", write as "0"	RW
b24	EEVDET	External event detection event enable	Refer to CMSA1 instructions	RW
b23	UDF6	Unit 6 zero event	Refer to CMSA1 instructions	RW
b22	OVF6	Unit 6 period event	Refer to CMSA1 instructions	RW
b21	CMSB6	Unit 6 SCMB event	Refer to CMSA1 instructions	RW
b20	CMSA6	Unit 6 SCMA event	Refer to CMSA1 instructions	RW
b19	UDF5	Unit 5 zero event	Refer to CMSA1 instructions	RW
b18	OVF5	Unit 5 period event	Refer to CMSA1 instructions	RW
b17	CMSB5	Unit 5 SCMB event	Refer to CMSA1 instructions	RW
b16	CMSA5	Unit 5 SCMA event	Refer to CMSA1 instructions	RW
b15	UDF4	Unit 4 zero event	Refer to CMSA1 instructions	RW
b14	OVF4	Unit 4 period event	Refer to CMSA1 instructions	RW
b13	CMSB4	Unit 4 SCMB event	Refer to CMSA1 instructions	RW
b12	CMSA4	Unit 4 SCMA event	Refer to CMSA1 instructions	RW
b11	UDF3	Unit 3 zero event	Refer to CMSA1 instructions	RW
b10	OVF3	Unit 3 period event	Refer to CMSA1 instructions	RW
b9	CMSB3	Unit 3 SCMB event	Refer to CMSA1 instructions	RW
b8	CMSA3	Unit 3 SCMA event	Refer to CMSA1 instructions	RW
b7	UDF2	Unit 2 zero event	Refer to CMSA1 instructions	RW
b6	OVF2	Unit 2 period event	Refer to CMSA1 instructions	RW
b5	CMSB2	Unit 2 SCMB event	Refer to CMSA1 instructions	RW
b4	CMSA2	Unit 2 SCMA event	Refer to CMSA1 instructions	RW
b3	UDF1	Unit 1 zero event	Refer to CMSA1 instructions	RW
b2	OVF1	Unit 1 period event	Refer to CMSA1 instructions	RW
b1	CMSB1	Unit 1 Unit 1 SCMB event	Refer to CMSA1 instructions	RW
b0	CMSA1	Unit 1 SCMA event	0: Unit 1 dedicated match event SCMA does not generate dedicated match event A 1: Unit 1 dedicated match event SCMA generates dedicated match event A	RW

19.6.1.48 High Resolution Control Register (HRPWM<t>_CR)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res	DACSRC[2：0]			Res		DACSYN2[1:0]		Res		DACSYNC1[1:0]		EN	Res		

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b15	Res	Reserved Bit	Read 0 when reading, write 0 when writing	RW
b14~b12	DACSRC	DAC Trigger Source	000: The timer value is equal to 0 001: counter value equals period value 010: Up count value equals SCMAR 011: Down count value equal to SCMAR 100: Upward count value equal to SCMBR 101: Down count value equals SCMBR Others: Prohibited settings	RW
b11~b10	Res	Reserved Bit	Read 0 when reading, write 0 when writing	RW
b9~b8	DACSYNC2[1:0]	DAC channel 2 sync source	00: Do not generate DAC synchronization trigger source 01: Generate a synchronous trigger source on dac_trg1_ch2 10: Setting prohibited 11: Setting prohibited	RW
b7~b6	Res	Reserved Bit	Read 0 when reading, write 0 when writing	RW
b5~b4	DACSYNC1[1:0]	DAC channel 1 sync source	00: Do not generate DAC synchronization trigger source 01: Generate a synchronous trigger source on dac_trg1_ch1 10: Generate a synchronous trigger source on dac_trg2_ch1 11: Setting prohibited	RW
b3	EN	High resolution mode enable bit	0: Disable high-precision HRPWM 1: Enable high-precision HRPWM	RW
b2~b0	Res	Reserved Bit	Read 0 when reading, write 0 when writing	RW

19.6.1.49 Phase Control Register (HRPWM<t>_PHSCTL, t=1)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res					BTR DPHS	BTR UPHS	BENP HS	Res							

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b11	Res	Reserved Bit	Read as "0"	R
b10	BTRDPHS	Phase shadow transfer time setting selection(ZERO)	0: Zero **Shadow Transfer Time**, Shadow value is not transferred 1: Zero **Shadow Transfer Time**, a Shadow value transfer occurs	RW
b9	BTRUPHS	Phase shadow transfer time setting selection(Period)	0: When the Period **Shadow Transfer Time** is used, the Shadow value is not transferred 1: Period **Shadow Transfer Time**, a Shadow value transfer occurs	RW
b8	BENPHS	Phase value shadow transmit enable	0: Buffer transmission is invalid 1: Buffer transfer enable	RW
b7~b0	Res	Reserved Bit	Read as "0", write as "0"	RW

19.6.1.50 Phase Control Register (HRPWM<t>_PHSCTL, t=2-6)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res								PHSFORCB	PHSFORCA	Res		PHCMPSEL[2:0]			PHSEN

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b8	Res	Reserved Bit	Read as "0"	RW
b7	PHSFORCB	output B is forced low when Phase match event occurs	In sawtooth mode: 0: When the unit 1 single buffer transfer point detects that the current count value is greater than or equal to the value set by the comparison value register HRGCMDR, output B has no effect 1: When the unit 1 single buffer transfer point detects that the current count value is greater than or equal to the value set by the comparison value register HRGCMDR, the output B is forced to be set to low In triangle count mode: 0: Output B has no effect when a phase matching event occurs 1: Output B is forced low when a phase match event occurs	RW
b6	PHSFORCA	output A is forced low when Phase match event occurs	In sawtooth mode: 0: When the unit 1 single buffer transfer point detects that the current count value is greater than or equal to the value set by the comparison value register HRGCMCR, output A has no effect 1: When the unit 1 single buffer transfer point detects that the current count value is greater than or equal to the value set by the comparison value register HRGCMCR, the output A is forced to be set to low In triangle count mode: 0: Output A has no effect when a phase matching event occurs 1: Output A is forced low when a phase match event occurs	RW
b5~b4	Res	Reserved Bit	Read as "0"	RW
b3~b1	PHCMPSEL[2:0]	Phase Matching Event Selection	000: Unit 1 Phase Matching Event 1 001: Unit 1 Phase Matching Event 2 010: Unit 1 Phase Matching Event 3 011: Unit 1 Phase Matching Event 4 100: Unit 1 Phase Matching Event 5	RW
b0	PHSEN	Phase Enable Bit	0: Disable the phase function 1: Enable phase function	RW

19.6.1.51 High-resolution phase Compare Register (HRPWM<t>_PHSCMPmn, m=1~5, n=A~B)

Reset value: 0x007F FFC0

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res									PHSCMPmn[22:16]						
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
PHSCMPmn[15:0]															

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b23	Res	Reserved Bit	Read 0 when reading, write 0 when writing	RW
b22~b0	PHSCMPmn	Phase compare value	Phase compare value setting and Shadow value The maximum value of HRPWM_PHSCMPmn in triangle count mode shall not be greater than 0x7FFFC0, and the maximum value in sawtooth wave mode shall not be greater than 0x3FFFC0	RW

19.6.2 HRPWM Common Registers

19.6.2.1 High Resolution Calibration Control Register (HRPWM_CALCR)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res					CALIE	ERRF	CALENF	Res		CALPRD[3：0]				CALEN	CAL

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b11	Res	Reserved Bit	Read 0 when reading, write 0 when writing	RW
b10	CALIE	Calibration complete interrupt enable	0: Calibration completion interrupt disable 1: Calibration completed interrupt enable	RW
b9	ERRF	Calibration error flag	0: No calibration error occurred 1: Calibration error Writing 0 to this bit clears it to 0, and writing 1 is invalid.	RW
b8	CALENF	Calibration completed flag	0: Calibration in progress or calibration not enable 1: Calibration completed Writing 0 to this bit clears it to 0, and writing 1 is invalid.	RW
b7~b6	resved	resved	Read 0 when reading, write 0 when writing	RW
b5~b2	CALPRD[3：0]	Periodic calibration period	0000: The period is 0xFFFFF, 8.7ms 0001: The period is 0xFFFFF*2, 17.4ms 0010: The period is 0xFFFFF*4, 34.8ms 0011: The period is 0xFFFFF*8, 69.6ms 0000: The period is 0xFFFFF*16, 139.2ms 0001: The period is 0xFFFFF*32, 278.4ms 0010: The period is 0xFFFFF*64, 556.8ms 0011: The period is 0xFFFFF*128, 1.11s 0100: The period is 0xFFFFF*256, 2.22s 0101: The period is 0xFFFFF*512, 4.45s 0110: The period is 0xFFFFF*1024, 8.91s 0111: The period is 0xFFFFF*2048, 17.81s 1100: The period is 0xFFFFF*4096, 35.63s 1101: The period is 0xFFFFF*8192, 71.27s 1110: The period is 0xFFFFF*16384, 142.54s 1111: The period is 0xFFFFF*32768, 285.08s	RW
b1	CALEN	Periodic calibration enable	0: Periodic calibration off 1: Periodic calibration	RW
b0	CAL	One-shot calibration enable	0: Single calibration off 1: Single calibration enable This bit must not be set to 1 at the same time as the CALEN bit.	W

19.6.2.2 Software Synchronous Capture Control Register (HRPWM_SCAPR)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res				SCAP 6B	SCAP 6A	SCAP 5B	SCAP 5A	SCAP 4B	SCAP 4A	SCAP 3B	SCAP 3A	SCAP 2B	SCAP 2A	SCAP 1B	SCAP 1A

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b12	Res	Reserved Bit	Read 0 when reading, write 0 when writing	RW
b11	SCAP6B	Software trigger unit 6 captures B	0: No impact 1: Software trigger unit 6 captures B	W
b10	SCAP6A	Software trigger unit 6 captures A	0: No impact 1: Software trigger unit 6 captures A	W
b9	SCAP5B	Software trigger unit 5 captures B	0: No impact 1: Software trigger unit 5 captures B	W
b8	SCAP5A	Software trigger unit 5 captures A	0: No impact 1: Software trigger unit 5 captures A	W
b7	SCAP4B	Software trigger unit 4 captures B	0: No impact 1: Software trigger unit 4 captures B	W
b6	SCAP4A	Software trigger unit 4 captures A	0: No impact 1: Software trigger unit 4 captures A	W
b5	SCAP3B	Software trigger unit 3 captures B	0: No impact 1: Software trigger unit 3 captures B	W
b4	SCAP3A	Software trigger unit 3 captures A	0: No impact 1: Software trigger unit 3 captures A	W
b3	SCAP2B	Software trigger unit 2 captures B	0: No impact 1: Software trigger unit 2 captures B	W
b2	SCAP2A	Software trigger unit 2 captures A	0: No impact 1: Software trigger unit 2 captures A	W
b1	SCAP1B	Software trigger unit 1 captures B	0: No impact 1: Software trigger unit 1 captures B	W
b0	SCAP1A	Software trigger unit 1 captures A	0: No impact 1: Software trigger unit 1 captures A	W

19.6.2.3 Software Synchronous Idle Control Register (HRPWM_SSTAIDLR)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res				SSTAIDL6B	SSTAIDL6A	SSTAIDL5B	SSTAIDL5A	SSTAIDL4B	SSTAIDL4A	SSTAIDL3B	SSTAIDL3A	SSTAIDL2B	SSTAIDL2A	SSTAIDL1B	SSTAIDL1A

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b12	Res	Reserved Bit	Read 0 when reading, write 0 when writing	RW
b11	SSTAIDL6B	Unit 6B channel immediate idle trigger	Refer to SSTAIDL1A functional description	W
b10	SSTAIDL6A	Unit 6 A channel immediate idle trigger	Refer to SSTAIDL1A functional description	W
b9	SSTAIDL5B	Unit 5B channel immediate idle trigger	Refer to SSTAIDL1A functional description	W
b8	SSTAIDL5A	Unit 5 A channel immediate idle trigger	Refer to SSTAIDL1A functional description	W
b7	SSTAIDL4B	Unit 4B channel immediate idle trigger	Refer to SSTAIDL1A functional description	W
b6	SSTAIDL4A	Unit 4 A channel immediate idle trigger	Refer to SSTAIDL1A functional description	W
b5	SSTAIDL3B	Unit 3B channel immediate idle trigger	Refer to SSTAIDL1A functional description	W
b4	SSTAIDL3A	Unit 3 A channel immediate idle trigger	Refer to SSTAIDL1A functional description	W
b3	SSTAIDL2B	Unit 2B channel immediate idle trigger	Refer to SSTAIDL1A functional description	W
b2	SSTAIDL2A	Unit 2 A channel immediate idle trigger	Refer to SSTAIDL1A functional description	W
b1	SSTAIDL1B	Unit 1B channel immediate idle trigger	Refer to SSTAIDL1A functional description	W
b0	SSTAIDL1A	Unit 1 A channel immediate idle trigger	0: No impact 1: Unit 1A is immediately idle.	W

19.6.2.4 Software Synchronous Operation Control Register 1 (HRPWM_SSTARUNR1)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res				SSTARUN6B	SSTARUN6A	SSTARUN5B	SSTARUN5A	SSTARUN4B	SSTARUN4A	SSTARUN3B	SSTARUN3A	SSTARUN2B	SSTARUN2A	SSTARUN1B	SSTARUN1A

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b12	Res	Reserved Bit	Read 0 when reading, write 0 when writing	RW
b11	SSTARUN6B	Unit 6 B immediately idle exit	Refer to SSTARUN1A function description	RW
b10	SSTARUN6A	Unit 6 A immediate idle exit	Refer to SSTARUN1A function description	RW
b9	SSTARUN5B	Unit 5 B immediately idle exit	Refer to SSTARUN1A function description	RW
b8	SSTARUN5A	Unit 5 A immediate idle exit	Refer to SSTARUN1A function description	RW
b7	SSTARUN4B	Unit 4 B immediately idle exit	Refer to SSTARUN1A function description	RW
b6	SSTARUN4A	Unit 4 A immediate idle exit	Refer to SSTARUN1A function description	RW
b5	SSTARUN3B	Unit 3 B immediately idle exit	Refer to SSTARUN1A function description	RW
b4	SSTARUN3A	Unit 3 A immediate idle exit	Refer to SSTARUN1A function description	RW
b3	SSTARUN2B	Unit 2 B immediately idle exit	Refer to SSTARUN1A function description	RW
b2	SSTARUN2A	Unit 2 A immediate idle exit	Refer to SSTARUN1A function description	RW
b1	SSTARUN1B	Unit 1 B immediately idle exit	Refer to SSTARUN1A function description	RW
b0	SSTARUN1A	Unit 1 A immediate idle exit	0: Unit 1PWMA Exit Immediate Idle Invalid 1: Unit 1PWMA exits immediately idle valid This bit can indicate the current state of PWM. When this bit is 1, it indicates that PWM has exited the immediate idle state. When this bit is 0, it indicates that PWM is in the immediate idle state. The reset default is 0. This bit can indicate whether the current channel PWM is in running or idle state during Burst Mode mode.	RW

19.6.2.5 Software Synchronous Operation Control Register 2 (HRPWM_SSTARUNR2)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res				SSTARUN6B2	SSTARUN6A2	SSTARUN5B2	SSTARUN5A2	SSTARUN4B2	SSTARUN4A2	SSTARUN3B2	SSTARUN3A2	SSTARUN2B2	SSTARUN2A2	SSTARUN1B2	SSTARUN1A2

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b12	Res	Reserved Bit	Read 0 when reading, write 0 when writing	RW
b11	SSTARUN6B2	Unit 6 B immediately idle exit	Refer to SSTARUN1A2 function description	W
b10	SSTARUN6A2	Unit 6 A immediate idle exit	Refer to SSTARUN1A2 function description	W
b9	SSTARUN5B2	Unit 5 B immediately idle exit	Refer to SSTARUN1A2 function description	W
b8	SSTARUN5A2	Unit 5 A immediate idle exit	Refer to SSTARUN1A2 function description	W
b7	SSTARUN4B2	Unit 4 B immediately idle exit	Refer to SSTARUN1A2 function description	W
b6	SSTARUN4A2	Unit 4 A immediate idle exit	Refer to SSTARUN1A2 function description	W
b5	SSTARUN3B2	Unit 3 B immediately idle exit	Refer to SSTARUN1A2 function description	W
b4	SSTARUN3A2	Unit 3 A immediate idle exit	Refer to SSTARUN1A2 function description	W
b3	SSTARUN2B2	Unit 2 B immediately idle exit	Refer to SSTARUN12A function description	W
b2	SSTARUN2A2	Unit 2 A immediate idle exit	Refer to SSTARUN1A2 function description	W
b1	SSTARUN1B2	Unit 1 B immediately idle exit	Refer to SSTARUN1A2 function description	W
b0	SSTARUN1A2	Unit 1 A immediate idle exit	0: No impact 1: Exit and idle immediately After writing 1 to this bit when the counter is stopped, the first counter start event PWMA encounters afterwards enters the running state. Writing 1 to this bit during the counter counting invalid.	W

19.6.2.6 Software Synchronization Delayed Idle Control Register (HRPWM_SSTADIDLR)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res										DLYSTRGU6	DLYSTRGU5	DLYSTRGU4	DLYSTRGU3	DLYSTRGU2	DLYSTRGU1

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b6	Res	-	Read 0 when reading, write 0 when writing	RW
b5	DLYSTRGU6	Unit 6 delayed idle software trigger	0: No impact 1: Software triggers unit 6 to delay idle When using it, you need to first set the corresponding unit DLYEVSEL[2:0] to 100, and then write 1 to this bit.	W
b4	DLYSTRGU5	Unit 5 delayed idle software trigger	0: No impact 1: Software triggers unit 5 to delay idle When using it, you need to first set the corresponding unit DLYEVSEL[2:0] to 100, and then write 1 to this bit.	W
b3	DLYSTRGU4	Unit 4 delayed idle software trigger	0: No impact 1: Software triggers unit 4 to delay idle When using it, you need to first set the corresponding unit DLYEVSEL[2:0] to 100, and then write 1 to this bit.	W
b2	DLYSTRGU3	Unit 3 delayed idle software trigger	0: No impact 1: Software triggers unit 3 to delay idle When using it, you need to first set the corresponding unit DLYEVSEL[2:0] to 100, and then write 1 to this bit.	W
b1	DLYSTRGU2	Unit 2 delayed idle software trigger	0: No impact 1: Software triggers unit 2 to delay idle When using it, you need to first set the corresponding unit DLYEVSEL[2:0] to 100, and then write 1 to this bit.	W
b0	DLYSTRGU1	Unit 1 delayed idle software trigger	0: No impact 1: Software triggers unit 1 to delay idle When using it, you need to first set the corresponding unit DLYEVSEL[2:0] to 100, and then write 1 to this bit.	W

19.6.2.7 High Resolution Global Control Register (HRPWM_GCTLR)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
EE10SRC2[1:0]		EE9SRC2[1:0]		EE8SRC2[1:0]		EE7SRC2[1:0]		EE6SRC2[1:0]		EE5SRC2[1:0]		EE4SRC2[1:0]		EE3SRC2[1:0]	
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res						BKDLY6	BKDLY5	BKDLY4	BKDLY3	BKDLY2	BKDLY1	EE2SRC2[1:0]		EE1SRC2[1:0]	

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b30	EE10SRC2[1:0]	External Event 10 Source 2 Selection	00: CMP1 01: CMP2 10: CMP3 11: No event source	RW
b29~b28	EE9SRC2[1:0]	External Event 9 Source 2 Selection	00: CMP1 01: CMP2 10: CMP3 11: No event source	RW
b27~b26	EE8SRC2[1:0]	External Event 8 Source 2 Selection	00: CMP1 01: CMP2 10: CMP3 11: No event source	RW
b25~b24	EE7SRC2[1:0]	External Event 7 Source 2 Selection	00: CMP1 01: CMP2 10: CMP3 11: No event source	RW
b23~b22	EE6SRC2[1:0]	External Event 6 Source 2 Selection	00: CMP1 01: CMP2 10: CMP3 11: No event source	RW
b21~b20	EE5SRC2[1:0]	External Event 5 Source 2 Selection	00: CMP1 01: CMP2 10: CMP3 11: No event source	RW
b19~b18	EE4SRC2[1:0]	External Event 4 Source 2 Selection	00: CMP1 01: CMP2 10: CMP3 11: No event source	RW
b17~b16	EE3SRC2[1:0]	External Event 3 Source 2 Selection	00: CMP1 01: CMP2 10: CMP3 11: No event source	RW
b15	Res	Reserved Bit	Read 0 when reading, write 0 when writing	RW
b14~b10	Res	Reserved Bit	Read 0 when reading, write 0 when writing	RW
b9	BKDLY6	Unit 6 Blanking Delay Mode	0: Blanking delay mode disable 1: Blanking delay mode enable	RW
b8	BKDLY5	Unit 5 Blanking Delay Mode	0: Blanking delay mode disable 1: Blanking delay mode enable	RW
b7	BKDLY4	Unit 4 Blanking Delay Mode	0: Blanking delay mode disable 1: Blanking delay mode enable	RW
b6	BKDLY3	Unit 3 Blanking Delay Mode	0: Blanking delay mode disable 1: Blanking delay mode enable	RW
b5	BKDLY2	Unit 2 Blanking Delay Mode	0: Blanking delay mode disable 1: Blanking delay mode enable	RW
b4	BKDLY1	Unit 1 Blanking Delay Mode	0: Blanking delay mode disable 1: Blanking delay mode enable	RW
b3~b2	EE2SRC2[1:0]	External Event 2 Source 2 Selection	00: CMP1 01: CMP2 10: CMP3 11: No event source	RW
b1~b0	EE1SRC2[1:0]	External Event 1 Source 2 Selection	00: CMP1 01: CMP2 10: CMP3 11: No event source	RW

19.6.2.8 Global Shadow Control Register (HRPWM_GBCONR)

Reset value: 0x0000 003F

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
OSTENU1	OSTBTRU1	Res													
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res		BTRU6SFT	BTRU5SFT	BTRU4SFT	BTRU3SFT	BTRU2SFT	BTRU1SFT	Res		BTRU6EN	BTRU5EN	BTRU4EN	BTRU3EN	BTRU2EN	BTRU1EN

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31	OSTENU1	Unit 1 one-shot Shadow enable	0: Unit 1 single Shadow disable 1: Unit 1 single Shadow enable Note: The **Shadow Transfer Time** of unit 1 is the first point after OSTBTRU1 is written 1 when the Sawtooth count mode value is equal to the period value or the sawtooth wave hardware is cleared or the triangle count mode value is equal to 0	RW
b30	OSTBTRU1	Software start unit 1 single Shadow trigger	0: No impact 1: Unit 1 single Shadow trigger	W
b29~b14	Res	Reserved Bit	Read as "0", write as "0"	RW
b13	BTRU6SFT	Unit 6 Software Shadow Trigger	0: No impact 1: Software triggers unit 6 to Shadow This bit is only valid when the counter is stopped.	RW
b12	BTRU5SFT	Unit 5 Software Shadow Trigger	0: No impact 1: Software triggers unit 5 to Shadow This bit is only valid when the counter is stopped.	RW
b11	BTRU4SFT	Unit 4 Software Shadow Trigger	0: No impact 1: Software triggers unit 4 to Shadow This bit is only valid when the counter is stopped.	RW
b10	BTRU3SFT	Unit 3 Software Shadow Trigger	0: No impact 1: Software triggers unit 3 to Shadow This bit is only valid when the counter is stopped.	RW
b9	BTRU2SFT	Unit 2 Software Shadow Trigger	0: No impact 1: Software triggers unit 2 to Shadow This bit is only valid when the counter is stopped.	RW
b8	BTRU1SFT	Unit 1 Software Shadow Trigger	0: No impact 1: Software triggers unit 1 to Shadow This bit is only valid when the counter is stopped.	RW
b7~b6	Res	Reserved Bit	Read as "0", write as "0"	RW
b5	BTRU6EN	Unit 6 Global Shadow Enable	Reference unit 1 global Shadow Enable Description	RW
b4	BTRU5EN	Unit 5 Global Shadow Enable	Reference unit 1 global Shadow Enable Description	RW
b3	BTRU4EN	Unit 4 Global Shadow Enable	Reference unit 1 global Shadow Enable Description	RW
b2	BTRU3EN	Unit 3 Global Shadow Enable	Reference unit 1 global Shadow Enable Description	RW
b1	BTRU2EN	Unit 2 Global Shadow Enable	Reference unit 1 global Shadow Enable Description	RW
b0	BTRU1EN	Unit 1 Global Shadow Enable	0: Unit 1 Shadow Enable masked 1: No impact When this bit is 0, HRPWM<t>_BCONR1. BENAE<BF><P><SPA><SPB>, HRPWM<t>_DCONR.DTBENU<D>, and HRPWM<t>_BCONR2.BENEEFWIN <BENEEFOFF><CTL> of unit 1 are masked to 0. At the same time, the HRPWM1_PHSCTL.BENPHS bit is set to 0	RW

19.6.2.9 Global Buffer Status Flag Register (HRPWM_GBSFLR)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res										U6BT RENF	U5BT RENF	U4BT RENF	U3BT RENF	U2BT RENF	U1BT RENF

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b6	Res	Reserved Bit	Read as "0", write as "0"	RW
b5	U6BTRENF	Unit 6 Global Shadow Completion Flag Bit	0: Unit 6 global Shadow not occurring 1: Unit 6 global Shadow occurs For the global Shadow completion flag setting conditions, refer to Table 19-12. Writing 0 to this bit clears it to 0, and writing 1 is invalid.	RW
b4	U5BTRENF	Unit 5 Global Shadow Completion Flag Bit	0: Unit 5 global Shadow not occurring 1: Unit 5 global Shadow occurs For the global Shadow completion flag setting conditions, refer to Table 19-12. Writing 0 to this bit clears it to 0, and writing 1 is invalid.	RW
b3	U4BTRENF	Unit 4 Global Shadow Completion Flag Bit	0: Unit 4 global Shadow not occurring 1: Unit 4 global Shadow occurs For the global Shadow completion flag setting conditions, refer to Table 19-12. Writing 0 to this bit clears it to 0, and writing 1 is invalid.	RW
b2	U3BTRENF	Unit 3 Global Shadow Completion Flag Bit	0: Unit 3 global Shadow not occurring 1: Unit 3 global Shadow occurs For the global Shadow completion flag setting conditions, refer to Table 19-12. Writing 0 to this bit clears it to 0, and writing 1 is invalid.	RW
b1	U2BTRENF	Unit 2 Global Shadow Completion Flag Bit	0: Unit 2 global Shadow not occurring 1: Unit 2 global Shadow occurs For the global Shadow completion flag setting conditions, refer to Table 19-12. Writing 0 to this bit clears it to 0, and writing 1 is invalid.	RW
b0	U1BTRENF	Unit 1 Global Shadow Completion Flag Bit	0: Unit 1 global Shadow not occurring 1: Unit 1 global Shadow occurs For the global Shadow completion flag setting conditions, refer to Table 19-12. Writing 0 to this bit clears it to 0, and writing 1 is invalid.	RW

19.6.2.10 Burst Mode Control Register (HRPWM_BMCR)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res										BMMD	BENBMP	BENBMCMP	INTENBMOVF	BMOVFF	BMOPTF
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
BMTMR6	BMTMR5	BMTMR4	BMTMR3	BMTMR2	BMTMR1	BMPSC[3:0]				BMCLKS[3:0]				BMCTN	BMEN

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b21	Res	Reserved Bit	Read as "0", write as "0"	RW
b21	BMMD	Burst Mode mode	0: Mode 1 1: Mode 2	RW
b20	BENBMP	Burst Mode Period shadow transfer enable	0: Buffer transmission is invalid 1: Buffer transfer enable (BMPERBR->BMPERAR) When the Burst Mode counter overflows, a buffer transfer occurs	RW
b19	BENBMCMP	Burst Mode Compare transfer enable	0: Buffer transmission is invalid 1: Buffer transfer enable (BMCMBR->BMCMAR) When the Burst Mode counter overflows, a buffer transfer occurs	RW
b18	INTENBMOVF	BM-counter overflow interrupt enable	0: When the BM-counter overflows, the interrupt is invalid. 1: When the BM-counter overflows, this interrupt is enabled	RW
b17	BMOVFF	BM-counter overflow match	0: BM-counter does not overflow 1: BM-counter overflow occurs Writing 0 to this bit clears it to 0, and writing 1 is invalid.	RW
b16	BMOPTF	Burst Mode operation flag	0: Normal operation 1: The Burst Mode is running. If this bit is written as 0, the Burst Mode ends early. This flag is set by hardware and cleared by hardware. When Burst Mode is enabled (BMEN = 1) and the trigger source selected by HRPWM_BMSTRG1/2 occurs, this bit is set. After writing 0 to this bit or setting BMEN to 0, and then all channels with Burst Mode enabled exit Burst Mode, this bit is automatically cleared.	RW
b15	BMTMR6	Burst Mode unit 6 counter control set	0: The Burst Mode unit 6 counter operates normally 1: The counter of unit 6 stops counting and is cleared to zero.	RW
b14	BMTMR5	Burst Mode unit 5 counter control set	0: The Burst Mode unit 5 counter operates normally 1: The counter of unit 6stops counting and is cleared to zero.	RW
b13	BMTMR4	Burst Mode unit 4 counter control set	0: The Burst Mode unit 4 counter operates normally 1: The counter of unit 6 stops counting and is cleared to zero.	RW
b12	BMTMR3	Burst Mode unit 3 counter control set	0: The Burst Mode unit 3 counter operates normally 1: The counter of unit 6 stops counting and is cleared to zero.	RW
b11	BMTMR2	Burst Mode unit 2 counter control set	0: The Burst Mode unit 2 counter operates normally 1: The counter of unit 6stops counting and is cleared to zero.	RW
b10	BMTMR1	Burst Mode unit 1 counter control set	0: The Burst Mode unit 1 counter operates normally 1: The counter of unit 1 stops counting and is cleared to zero.	RW
b9~b6	BMPSC[3:0]	Burst Mode clock division	BMPSC[3:0] Count clock selection 0000：PCLK0 0001：PCLK0/2 0010：PCLK0/4 0011：PCLK0/8 0100：PCLK0/16 0101：PCLK0/32 0110：PCLK0/64 0111：PCLK0/128 1000：PCLK0/256	RW

			1001：PCLK0/512 1010：PCLK0/1024 1011：PCLK0/2048 1100：PCLK0/4096 1101：PCLK0/8192 1110：PCLK0/16384 1111：PCLK0/32768	
b5~b2	BMCLKS[3:0]	Burst Mode count clock selection	0000: Unit 1 complete cycle point as counting condition 0001: Unit 2 complete cycle point as counting condition 0010: Unit 3 complete cycle point as counting condition 0011: Unit 4 saws the complete cycle point as the counting condition 0100: Unit 5 saws the complete cycle point as the counting condition 0101: Unit 6 saws the complete cycle point as the counting condition 0110：TMR0_1_CMPA 0111：TMR0_1_CMPB 1000：TMR0_2_CMPA 1001：TMR0_2_CMPB 1010: PCLK0 frequency division (according to BMPSC[3:0] setting) Note: The Full period point is defined by HRPWM<t>_GCONR1.PRDSEL	RW
b1	BMCTN	work mode with Burst Mode	0: Single-shot mode. When the BM counter reaches the value of HRPWM_BMCMAR, counting stops. 1: Continuous mode. When the BM counter reaches the value of HRPWM_BMCMAR, it rolls over to 0 and continues counting.	RW
b0	BMEN	Burst Mode enable	0: Burst Mode disable 1: Burst Mode enable When this bit is 1, the Burst Mode controller is ready to receive an Burst Mode start trigger. Writing 0 to this bit will terminate the Burst Mode	RW

19.6.2.11 Burst Mode Start Trigger Register 1 (HRPWM_BMSTRG1)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res	EEV8	EEV7	T4UDFEEV8	T4OVFEEV8	T1UDFEEV7	T1OVFEEV7	GCMB6	GCMA6	UDF6	OVF6	GCMB5	GCMA5	UDF5	OVF5	GCMB4
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
GCMA4	UDF4	OVF4	GCMB3	GCMA3	UDF3	OVF3	GCMB2	GCMA2	UDF2	OVF2	GCMB1	GCMA1	UDF1	OVF1	SSTRG

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31	Res	Reserved Bit	Read as "0", write as "0"	RW
b30	EEV8	External event 8 trigger Burst Mode	Condition: External event 8 Burst Mode valid (filtered by unit 4) 0: When the condition is matched, there is no effect on the Burst Mode 1: When the conditions are matched, start the Burst Mode	RW
b29	EEV7	External event 7 trigger Burst Mode	Condition: External event 7 Burst Mode valid (filtered by unit 1) 0: When the condition is matched, there is no effect on the Burst Mode 1: When the conditions are matched, start the Burst Mode	RW
b28	T4UDFEEV8	The Zero event of unit 4 after external event 8 triggers the Burst Mode	Condition: After external event 8, the count value of unit 4 is equal to 0, and the trigger Burst Mode valid 0: When the condition is matched, there is no effect on the Burst Mode 1: When the conditions are matched, start the Burst Mode	RW
b27	T4OVFEEV8	Unit 4 Period Event after external event 8 match event triggers Burst Mode	Condition: The count value of unit 4 after external event 8 is equal to the period value or the hardware clear in sawtooth counter mode trigger Burst Mode valid 0: When the condition is matched, there is no effect on the Burst Mode 1: When the conditions are matched, start the Burst Mode	RW
b26	T1UDFEEV7	Unit 1 zero match event after external event 7 triggers Burst Mode	Condition: After external event 7, the count value of unit 1 is equal to 0, and the trigger Burst Mode valid. 0: When the condition is matched, there is no effect on the Burst Mode 1: When the conditions are matched, start the Burst Mode	RW
b25	T1OVFEEV7	Unit 1 period after external event 7 match event triggers Burst Mode	Condition: The count value of unit 1 after external event 7 is equal to the period value or the sawtooth wave should clear the trigger Burst Mode to be valid 0: When the condition is matched, there is no effect on the Burst Mode 1: When the conditions are matched, start the Burst Mode	RW
b24	GCMB6	Unit 6 CMP B triggers Burst Mode	Condition: Unit 6 CMP B triggers Burst Mode 0: When the condition is matched, there is no effect on the Burst Mode 1: When the conditions are matched, start the Burst Mode	RW
b23	GCMA6	Unit 6 CMP A triggers Burst Mode	Condition: Unit 6 CMP A triggers Burst Mode 0: When the condition is matched, there is no effect on the Burst Mode 1: When the conditions are matched, start the Burst Mode	RW
b22	UDF6	Unit 6 zero match event triggers Burst Mode	Condition: Unit 6 zero match event triggers Burst Mode valid 0: When the condition is matched, there is no effect on the Burst Mode 1: When the conditions are matched, start the Burst Mode	RW
b21	OVF6	Unit 6 period match event triggers Burst Mode	Condition: Unit 6 periodmatch event trigger Burst Mode valid 0: When the condition is matched, there is no effect on the Burst Mode 1: When the conditions are matched, start the Burst Mode	RW
b20	GCMB5	Unit 5 CMP B triggers Burst Mode	Condition: Unit 3 CMP B triggers Burst Mode 0: When the condition is matched, there is no effect on the Burst Mode 1: When the conditions are matched, start the Burst Mode	RW
b19	GCMA5	Unit 5 CMP A triggers Burst Mode	Condition: Unit 3 CMP A triggers Burst Mode 0: When the condition is matched, there is no effect on the Burst Mode 1: When the conditions are matched, start the Burst Mode	RW
b18	UDF5	Unit 5 zero match event triggers Burst	Condition: Unit 4 zero match event triggers Burst Mode valid	RW

		Mode	0: When the condition is matched, there is no effect on the Burst Mode 1: When the conditions are matched, start the Burst Mode	
b17	OVF5	Unit 5 period match event triggers Burst Mode	Condition: Unit 4 periodmatch event trigger Burst Mode valid 0: When the condition is matched, there is no effect on the Burst Mode 1: When the conditions are matched, start the Burst Mode	RW
b16	GCMB4	Unit 4 CMP B triggers Burst Mode	Condition: Unit 3 CMP B triggers Burst Mode 0: When the condition is matched, there is no effect on the Burst Mode 1: When the conditions are matched, start the Burst Mode	RW
b15	GCMA4	Unit 4 CMP A triggers Burst Mode	Condition: Unit 3 CMP A triggers Burst Mode 0: When the condition is matched, there is no effect on the Burst Mode 1: When the conditions are matched, start the Burst Mode	RW
b14	UDF4	Unit 4 zero match event triggers Burst Mode	Condition: Unit 4 zero match event triggers Burst Mode valid 0: When the condition is matched, there is no effect on the Burst Mode 1: When the conditions are matched, start the Burst Mode	RW
b13	OVF4	Unit 4 period match event triggers Burst Mode	Condition: Unit 4 period match event trigger Burst Mode valid 0: When the condition is matched, there is no effect on the Burst Mode 1: When the conditions are matched, start the Burst Mode	RW
b12	GCMB3	Unit 3 CMP B triggers Burst Mode	Condition: Unit 3 CMP B triggers Burst Mode 0: When the condition is matched, there is no effect on the Burst Mode 1: When the conditions are matched, start the Burst Mode	RW
b11	GCMA3	Unit 3 CMP A triggers Burst Mode	Condition: Unit 3 CMP A triggers Burst Mode 0: When the condition is matched, there is no effect on the Burst Mode 1: When the conditions are matched, start the Burst Mode	RW
b10	UDF3	Unit 3 zero match event triggers Burst Mode	Condition: Unit 3 zero match event triggers Burst Mode valid 0: When the condition is matched, there is no effect on the Burst Mode 1: When the conditions are matched, start the Burst Mode	RW
b9	OVF3	Unit 3 period match event triggers Burst Mode	Condition: Unit 3 perio dmatch event trigger Burst Mode valid 0: When the condition is matched, there is no effect on the Burst Mode 1: When the conditions are matched, start the Burst Mode	RW
b8	GCMB2	Unit 2 CMP B triggers Burst Mode	Condition: Unit 2 CMP B triggers Burst Mode 0: When the condition is matched, there is no effect on the Burst Mode 1: When the conditions are matched, start the Burst Mode	RW
b7	GCMA2	Unit 2 CMP A triggers Burst Mode	Condition: Unit 2 CMP A triggers Burst Mode 0: When the condition is matched, there is no effect on the Burst Mode 1: When the conditions are matched, start the Burst Mode	RW
b6	UDF2	Unit 2 zero match event triggers Burst Mode	Condition: Unit 2 zero match event triggers Burst Mode valid 0: When the condition is matched, there is no effect on the Burst Mode 1: When the conditions are matched, start the Burst Mode	RW
b5	OVF2	Unit 2 period match event triggers Burst Mode	Condition: Unit 2 periodmatch event trigger Burst Mode valid 0: When the condition is matched, there is no effect on the Burst Mode 1: When the conditions are matched, start the Burst Mode	RW
b4	GCMB1	Unit 1 CMP B triggers Burst Mode	Condition: Unit 1 CMP B triggers Burst Mode 0: When the condition is matched, there is no effect on the Burst Mode 1: When the conditions are matched, start the Burst Mode	RW
b3	GCMA1	Unit 1 CMP A triggers Burst Mode	Condition: Unit 1 CMP A triggers Burst Mode 0: When the condition is matched, there is no effect on the Burst Mode 1: When the conditions are matched, start the Burst Mode	RW
b2	UDF1	Unit 1 zero match event triggers Burst Mode	Condition: Unit 1 zero match event triggers Burst Mode valid 0: When the condition is matched, there is no effect on the Burst Mode 1: When the conditions are matched, start the Burst Mode	RW
b1	OVF1	Unit 1 period match event triggers Burst Mode	Condition: Unit 1 period match event trigger Burst Mode valid 0: When the condition is matched, there is no effect on the Burst Mode	RW

			1: When the conditions are matched, start the Burst Mode	
b0	SSTRG	Software triggered Burst Mode	Writing 1 to this bit triggers the Burst Mode. If Burst Mode is not enabled (HRPWM_BMCR.BMEN=0), this bit is invalid. 0: No effect on Burst Mode 1: Software triggers the start of Burst Mode	W

19.6.2.12 Burst Mode Start Trigger Register 2 (HRPWM_BMSTRG2)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res												IEV3	IEV2	IEV1	IEV0

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b4	Res	Reserved Bit	Read as "0", write as "0"	RW
b3	IEV3	TMR0_2_CMPB trigger Burst Mode	Condition: TMR0_2_CMPB trigger Burst Mode valid 0: When the condition is matched, there is no effect on the Burst Mode 1: When the conditions are matched, start the Burst Mode	RW
b2	IEV2	TMR0_2_CMPA trigger Burst Mode	Condition: TMR0_2_CMPA trigger Burst Mode valid 0: When the condition is matched, there is no effect on the Burst Mode 1: When the conditions are matched, start the Burst Mode	RW
b1	IEV1	TMR0_1_CMPB trigger Burst Mode	Condition: TMR0_1_CMPB trigger Burst Mode valid 0: When the condition is matched, there is no effect on the Burst Mode 1: When the conditions are matched, start the Burst Mode	RW
b0	IEV0	TMR0_1_CMPA trigger Burst Mode	Condition: TMR0_1_CMPA trigger Burst Mode valid 0: When the condition is matched, there is no effect on the Burst Mode 1: When the conditions are matched, start the Burst Mode	RW

19.6.2.13 Burst Mode Period Register (HRPWM_BMPERmR, m=A~B)

Reset value: 0x0000 FFFF

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
BMPERA-C[15:0]															

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b16	Res	Reserved Bit	Read as "0"	R
b15~b0	BMPERA-B[15:0]	Burst Mode period value	Set the counting period value and corresponding Shadow value of each counting round. The period is the sum of the duration of the idle state and the running state. When Burst Mode is enabled, this register cannot be 0.	RW

19.6.2.14 Burst Mode Compare Register (HRPWM_BMCMmR, m=A~B)

Reset value: 0x0000 FFFF

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
BMCMA-B[15:0]															

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b16	Res	Reserved Bit	Read as "0"	R
b15~b0	BMCMA-B[15:0]	Burst Mode compare value	This bit contains the value compared with the BM counter and its buffered value and defines the duration of idle time at the Burst Mode.	RW

19.6.2.15 External Event Control Register 1 (HRPWM_EECR1)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res		EE5FAST	EE5SNS[1:0]		EE5POL	EE5SRC[1:0]		EE4FAST	EE4SNS[1:0]		EE4POL	EE4SRC[1:0]		EE3FAST	EE3SNS[1]

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
EE3SNS[0]	EE3POL	EE3SRC[1:0]		EE2FAST	EE2SNS[1:0]		EE2POL	EE2SRC[1:0]		EE1FAST	EE1SNS[1:0]		EE1POL	EE1SRC[1:0]	

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b30	Res	Reserved Bit	Read as "0", write as "0"	RW
b29	EE5FAST	External Event 5 Fast Asynchronous Mode	Refer to the description of EE1FAST	RW
b28~b27	EE5SNS[1:0]	External event 5 valid edge	Refer to the description of EE1SNS	RW
b26	EE5POL	External Event 5 Polarity	Refer to the description of EE1POL	RW
b25~b24	EE5SRC[1:0]	External Event 5 Sources	Refer to the description of EE1SRC[1:0]	RW
b23	EE4FAST	External Event 4 Fast Asynchronous Mode	Refer to the description of EE1FAST	RW
b22~b21	EE4SNS[1:0]	External event 4 valid edge	Refer to the description of EE1SNS	RW
b20	EE4POL	External Event 4 Polarity	Refer to the description of EE1POL	RW
b19~b18	EE4SRC[1:0]	External Event 4 Sources	Refer to the description of EE1SRC[1:0]	RW
b17	EE3FAST	External Event 3 Fast Asynchronous Mode	Refer to the description of EE1FAST	RW
b16~b15	EE3SNS[1:0]	External event 3 valid edge	Refer to the description of EE1SNS	RW
b14	EE3POL	External Event 3 Polarity	Refer to the description of EE1POL	RW
b13~b12	EE3SRC[1:0]	External Event 3 Sources	Refer to the description of EE1SRC[1:0]	RW
b11	EE2FAST	External Event 2 Fast Asynchronous Mode	Refer to the description of EE1FAST	RW
b10~b9	EE2SNS[1:0]	External event 2 valid edge	Refer to the description of EE1SNS	RW
b8	EE2POL	External Event 2 Polarity	Refer to the description of EE1POL	RW
b7~b6	EE2SRC[1:0]	External Event 2 Sources	Refer to the description of EE1SRC[1:0]	RW
b5	EE1FAST	External Event 1 Fast Asynchronous Mode	0: External event 1 fast asynchronous mode disable 1: External event 1 fast asynchronous mode enable	RW
b4~b3	EE1SNS[1:0]	External event 1 valid edge	00: Level valid, the valid level is configured by EE1POL 01: rising edge valid 10: Falling edge valid 11: Both edges are valid	RW
b2	EE1POL	External Event 1 Polarity	This bit is only valid when EE1SNS[1:0]=00. 0: External event 1 high level valid 1: External event 1 low level valid	
b1~b0	EE1SRC[1:0]	External Event 1 Source	00: The source of external event 1 is EE1src1 01: The source of external event 1 is EE1src2 10: The source of external event 1 is EE1src3 11: The source of external event 1 is EE1src4 When the register is set to 01, the source of EE1src2 is configured according to HRPWM_GCTLR.EE1SRC2[1:0]. The event source corresponding to each event is referenced Table 19-5	RW

19.6.2.16 External Event Control Register 2 (HRPWM_EECR2)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res			EE10SNS[1:0]		EE10POL	EE10SRC[1:0]		Res	EE9SNS[1:0]		EE9POL	EE9SRC[1:0]		Res	EE9SNS[1]
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
EE8SNS[0]	EE8POL	EE8SRC[1:0]		Res	EE7SNS[1:0]		EE7POL	EE7SRC[1:0]		Res	EE6SNS[1:0]		EE6POL	EE6SRC[1:0]	

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b29	Res	Reserved Bit	Read as "0", write as "0"	RW
b28~b27	EE10SNS[1:0]	External event 10 valid edge	Refer to the description of EE1SNS	RW
b26	EE10POL	External Event 10 Polarity	Refer to the description of EE1POL	RW
b25~b24	EE10SRC[1:0]	External Event 10 Source	Refer to the description of EE1SRC[1:0]	RW
b23	Res	Reserved Bit	Read as "0", write as "0"	RW
b22~b21	EE9SNS[1:0]	External event 9 valid edge	Refer to the description of EE1SNS	RW
b20	EE9POL	External Event 9 Polarity	Refer to the description of EE1POL	RW
b19~b18	EE9SRC[1:0]	External Event 9 Source	Refer to the description of EE1SRC[1:0]	RW
b17	Res	Reserved Bit	Read as "0", write as "0"	RW
b16~b15	EE8SNS[1:0]	External event 8 valid edge	Refer to the description of EE1SNS	RW
b14	EE8POL	External Event 8 Polarity	Refer to the description of EE1POL	RW
b13~b12	EE8SRC[1:0]	External Event 8 Source	Refer to the description of EE1SRC[1:0]	RW
b11	Res	Reserved Bit	Read as "0", write as "0"	RW
b10~b9	EE7SNS[1:0]	External event 7 valid edge	Refer to the description of EE1SNS	RW
b8	EE7POL	External Event 7 Polarity	Refer to the description of EE1POL	RW
b7~b6	EE7SRC[1:0]	External Event 7 Source	Refer to the description of EE1SRC[1:0]	RW
b5	Res	Reserved Bit	Read as "0", write as "0"	RW
b4~b3	EE6SNS[1:0]	External event 6 valid edge	Refer to the description of EE1SNS	RW
b2	EE6POL	External Event 6 Polarity	Refer to the description of EE1POL	RW
b1~b0	EE6SRC[1:0]	External Event 6 Source	Refer to the description of EE1SRC[1:0]	RW

19.6.2.17 External Event Control Register 3 (HRPWM_EECR3)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
EEVSD[1:0]		Res		EE10F[3:0]				Res		EE9F[3:0]				Res	
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
EE8F[3:0]				Res		EE7F[3:0]				Res		EE6F[3:0]			

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b30	EEVSD[1:0]	External event sampling clock division	00: $f_{EEVS}=f_{PCLK0}$ 01: $f_{EEVS}=f_{PCLK0}/2$ 10: $f_{EEVS}=f_{PCLK0}/4$ 11: $f_{EEVS}=f_{PCLK0}/8$	RW
b29~b28	Res	Reserved Bit	Read as "0", write as "0"	RW
b27~b24	EE10F[3:0]	External event 10 filtering	Refer to the description of EE6F	RW
b23~b22	Res	Reserved Bit	Read as "0", write as "0"	RW
b21~b18	EE9F[3:0]	External event 9 filtering	Refer to the description of EE6F	RW
b17~b16	Res	Reserved Bit	Read as "0", write as "0"	RW
b15~b12	EE8F[3:0]	External event 8 filtering	Refer to the description of EE6F	RW
b11~b10	Res	Reserved Bit	Read as "0", write as "0"	RW
b9~b6	EE7F[3:0]	External event 7 filtering	Refer to the description of EE6F	RW
b5~b4	Res	Reserved Bit	Read as "0", write as "0"	RW
b3~b0	EE6F[3:0]	External event 6 filtering	This bit defines how often external events are sampled, and the length of the filter applied to external events. 0000: No filtering 0001: sampling frequency is f_{PCLK0}, N=2 0010: sampling frequency is f_{PCLK0}, N=4 0011: sampling frequency is f_{PCLK0}, N=8 0100: sampling frequency is f_{EEVS} /2, N=6 0101: sampling frequency is f_{EEVS} /2, N=8 0110: sampling frequency is f_{EEVS} /4, N=6 0111: sampling frequency is f_{EEVS} /4, N=8 1000: sampling frequency is f_{EEVS} /8, N=6 1001: sampling frequency is f_{EEVS} /8, N=8 1010: sampling frequency is f_{EEVS} /16, N=5 1011: sampling frequency is f_{EEVS} /16, N=6 1100: sampling frequency is f_{EEVS} /16, N=8 1101: sampling frequency is f_{EEVS} /32, N=5 1110: sampling frequency is f_{EEVS} /32, N=6 1111: sampling frequency is f_{EEVS} /32, N=8	RW

19.6.2.18 Synchronous Output Control Register (HRPWM_SYNOCR)

Reset value: 0x0000 1000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
SYNCMP[7:0]								Res	SRCDIR	SYNOPLS[1:0]		SYNCSRC[3:0]			

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b16	Res	Reserved Bit	Read as "0", write as "0"	RW
b15~b8	SYNCMP[7:0]	Synchronous output pulse width	Generate a pulse length of SYNCMP[7:0] PCLK0 on HRPWM_SCOUT This bit must be set to greater than or equal to 16	RW
b7	Res	Reserved Bit	Read as "0", write as "0"	RW
b6	SRCDIR	Synchronous source counting direction	0: The synchronization source valid in the down counting interval 1: The synchronization source valid in the upward counting interval	RW
b5~b4	SYNOPLS[1:0]	Synchronous output pulse	0x: Pulse generation disabled 10: Generate a positive pulse on HRPWM_SCOUT with a length of SYNCMP[7:0] PCLK0 11: Generate a negative pulse on HRPWM_SCOUT with a length of SYNCMP[7:0] PCLK0	RW
b3~b0	SYNCSRC[3:0]	Synchronization Source	0000: The timer count value of unit 1 is 0 0001: The timer count value of unit 2 is 0 0010: The timer count value of unit 3 is 0 0011: The timer count value of unit 4 is 0 0100: The timer count value of unit 5 is 0 0101: The timer count value of unit 6 is 0 0110: The timer count value of unit 1 is SCMB 0111: The timer count value of unit 2 is SCMB 1000: The timer count value of unit 3 is SCMB 1001: The timer count value of unit 4 is SCMB 1010: The timer count value of unit 5 is SCMB 1011: The timer count value of unit 6 is SCMB When the setting value is 0110~1011, it is necessary to configure the SRCDIR bit to set the effective counting interval of the CMP.	RW

19.6.2.19 External Event Detection Selection register (HRPWM_EEDSELR)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res						EEV10	EEV9	EEV8	EEV7	EEV6	EEV5	EEV4	EEV3	EEV2	EEV1

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b10	Res	Reserved Bit	Read as "0", write as "0"	RW
b9	EEV10	External Events 10 selection	Refer to EEV1 instructions	RW
b8	EEV9	External Events 9 selection	Refer to EEV1 instructions	RW
b7	EEV8	External Events 8 selection	Refer to EEV1 instructions	RW
b6	EEV7	External Events 7 selection	Refer to EEV1 instructions	RW
b5	EEV6	External Events 6 selection	Refer to EEV1 instructions	RW
b4	EEV5	External Events 5 selection	Refer to EEV1 instructions	RW
b3	EEV4	External Events 4 selection	Refer to EEV1 instructions	RW
b2	EEV3	External Events 3 selection	Refer to EEV1 instructions	RW
b1	EEV2	External Events 2 selection	Refer to EEV1 instructions	RW
b0	EEV1	External Events 1 selection	0: When external event 1 occurs, it does not become an external event detection event 1: Generate an external event detection event when external event 1 occurs	RW

19.6.2.20 Filter Control Register (HRPWM_FCNTR)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res	NOFI CKTD[1:0]		NOFI ENTD	Res	NOFI CKTC[1:0]		NOFI ENTC	Res	NOFI CKTB[1:0]		NOFI ENTB	Res	NOFI CKTA[1:0]		NOFI ENTA

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b15	Res	Reserved Bit	Read as "0", write as "0"	RW
b14~b13	NOFICKTD[1:0]	Filter sampling reference clock selection TD	00：PCLK0 01：PCLK0/4 10：PCLK0/16 11：PCLK0/64	RW
b12	NOFIENTD	Capture input port filter TD	0: HRPWM_TRIGD port input filter function invalid 1: HRPWM_TRIGD port input filter Function Enable	RW
b11	Res	Reserved Bit	Read as "0", write as "0"	RW
b10~b9	NOFICKTC[1:0]	Filter sampling reference clock selection TC	00：PCLK0 01：PCLK0/4 10：PCLK0/16 11：PCLK0/64	RW
b8	NOFIENTC	Capture input port filter TC	0: HRPWM_TRIGC port input filter function invalid 1: HRPWM_TRIGC port input filter Function Enable	RW
b7	Res	Reserved Bit	Read as "0", write as "0"	RW
b6~b5	NOFICKTB[1:0]	Filter sampling reference clock selection TB	00：PCLK0 01：PCLK0/4 10：PCLK0/16 11：PCLK0/64	RW
b4	NOFIENTB	Capture input port filter TB	0: HRPWM_TRIGB port input filter function invalid 1: HRPWM_TRIGB port input filter Function Enable	RW
b3	Res	Reserved Bit	Read as "0", write as "0"	RW
b2~b1	NOFICKTA[1:0]	Filter sampling reference clock selection TA	00：PCLK0 01：PCLK0/4 10：PCLK0/16 11：PCLK0/64	RW
b0	NOFIENTA	Capture input port filter TA	0: HRPWM_TRIGA port input filter function invalid 1: HRPWM_TRIGA port input filter Function Enable	RW

19.6.2.21 Software Synchronous Start Control Register (HRPWM_SSTAR)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res										SSTA6	SSTA5	SSTA4	SSTA3	SSTA2	SSTA1

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b6	Res	Reserved Bit	Read as "0", write as "0"	RW
b5	SSTA6	Unit 6 Software Start	0: Software startup is invalid 1: Software startup enable	RW
b4	SSTA5	Unit 5 software start	0: Software startup is invalid 1: Software startup enable	RW
b3	SSTA4	Unit 4 software start	0: Software startup is invalid 1: Software startup enable	RW
b2	SSTA3	Unit 3 software start	0: Software startup is invalid 1: Software startup enable	RW
b1	SSTA2	Unit 2 software start	0: Software startup is invalid 1: Software startup enable	RW
b0	SSTA1	Unit 1 software start	0: Software startup is invalid 1: Software startup enable	RW

19.6.2.22 Software Synchronous Stop Control Register (HRPWM_SSTPR)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res										SSTP6	SSTP5	SSTP4	SSTP3	SSTP2	SSTP1

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b6	Res	Reserved Bit	Read as "0", write as "0"	RW
b5	SSTP6	Unit 6 Software Stop	0: Software stop is invalid 1: Software stop enable	W
b4	SSTP5	Unit 5 software stop	0: Software stop is invalid 1: Software stop enable	W
b3	SSTP4	Unit 4 software stop	0: Software stop is invalid 1: Software stop enable	W
b2	SSTP3	Unit 3 software stop	0: Software stop is invalid 1: Software stop enable	W
b1	SSTP2	Unit 2 software stop	0: Software stop is invalid 1: Software stop enable	W
b0	SSTP1	Unit 1 software stop	0: Software stop is invalid 1: Software stop enable	W

19.6.2.23 Software Synchronous Clear Control Register (HRPWM_SCLRR)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res										SCLE6	SCLE5	SCLE4	SCLE3	SCLE2	SCLE1

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b6	Res	Reserved Bit	Read as "0", write as "0"	RW
b5	SCLE6	Unit 6 software clear	0: Software reset is invalid 1: Software clear enable	W
b4	SCLE5	Unit 5 software clear	0: Software reset is invalid 1: Software clear enable	W
b3	SCLE4	Unit 4 software clear	0: Software reset is invalid 1: Software clear enable	W
b2	SCLE3	Unit 3 software clear	0: Software reset is invalid 1: Software clear enable	W
b1	SCLE2	Unit 2 software clear	0: Software reset is invalid 1: Software clear enable	W
b0	SCLE1	Unit 1 software clear	0: Software reset is invalid 1: Software clear enable	W

19.6.2.24 Software Synchronous Update Control Register (HRPWM_SUPDR)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res										SUPD6	SUPD5	SUPD4	SUPD3	SUPD2	SUPD1

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b6	Res	Reserved Bit	Read as "0", write as "0"	RW
b5	SUPD6	Unit 6 software update	0: Software update invalid 1: Software update enable	W
b4	SUPD5	Unit 5 software update	0: Software update invalid 1: Software update enable	W
b3	SUPD4	Unit 4 software update	0: Software update invalid 1: Software update enable	W
b2	SUPD3	Unit 3 software update	0: Software update invalid 1: Software update enable	W
b1	SUPD2	Unit 2 software update	0: Software update invalid 1: Software update enable	W
b0	SUPD1	Unit 1 software update	0: Software update invalid 1: Software update enable	W

19.7Precautions

1) The CPU cannot write to the CNTER register when the counter is counting, and can only write to the CNTER register when the counter stops. Software synchronous update can update the counter value to any value during the counter counting process, which may cause the loss of comparison match events. Please pay attention when using it.
2) When the corresponding buffer function of HRGCMCR and HRGCMDR valid, the interrupt output and event output are invalid.
3) There are certain limitations on the pulse width of high-precision PWM output. The minimum pulse is 3 PCLK0 cycles, and the maximum pulse width is the PWM cycle value - 3 PCLK0 cycles. When setting the reference value, ensure that the pulse width meets this limitation, otherwise PWM will not be able to guarantee high precision.
4) When using the complementary PWM output function, the user needs to calculate and set the relevant reference value and port output state to ensure that the port is in the expected output state.

In some specific situations, the control events of the counter may appear at the same time. At this time, the change control priority of the counter (CNTER) is shown in the following Table 19-17.In some specific situations, the control events of the counter may appear at the same time. At this time, the change control priority of the counter (CNTER) is shown in the following Table 19-17.

Table 19-17 Counter (CNTER) Control Priority

Timer Control Events		Counter (CNTER) value	Priority
The CPU performs a write operation on CNTER		CPU write value	high
Clear event (software clear or hardware clear)	Sawtooth mode	0	↓
	In triangle count mode	0	↓
Triangle count mode counting		Normal count value	↓
Sawtooth pattern counting		Normal count value	Low

In some specific situations, the control events of the PWM port output may appear at the same time. At this time, the output change control priorities of the two ports (HRPWM_<t>_PWMA and HRPWM_<t>_PWMB) are shown in Table 19-18.

Table 19-18 PWMA, B Port Output Control Priority

PWMA<B> port output control event		Port Status	Priority
EMB selected event occurs		PCNA<B>R1.EMBCA<B>Setting Status	high
Idle state		IDLECR.IDLESA<B>	↓
Set forced port output		PCNA<B>R1.FORCA<B>Setting status	↓
When counting starts		PCNA<B>R1.STACA<B>Setting status	↓
When the Sawtooth count modes up	When external events occur	PCNA<B>R2.EVEyUCA<B>Setting status	↓

PWMA<B> port output control event		Port Status	Priority
	Phase forced output	PHSCTL.PHSFORCA<B>Settings	↓
	The count value is HRPERAR	PCNA<B>R1.OVFCA<B>Set Status	↓
	The count value is HRGCMAR	PCNA<B>R1.CMAUCA<B>Setting status	↓
	The count value is HRGCMBR	PCNA<B>R1.CMBUCA<B>Setting status	↓
	The count value is HRGCMER	PCNA<B>R2.CMEUCA<B>Setting status	↓
	The count value is HRGCMFR	PCNA<B>R2.CMFUCA<B>Setting status	↓
	The count value is SCMAR	PCNA<B>R2.SCMAUCA<B>Setting status	↓
	The count value is SCMBR	PCNA<B>R2.SCMBUCA<B>Setting status	↓
	The count value is 0	PCNA<B>R1.UDFCA<B>Setting status	↓
When the triangle count modes up	When external events occur	PCNA<B>R2.EVEyUCA<B>Setting status	↓
	Phase forced output	PHSCTL.PHSFORCA<B>Settings	↓
	The count value is HRGCMAR	PCNA<B>R1.CMAUCA<B>Setting status	↓
	The count value is HRGCMBR	PCNA<B>R1.CMBUCA<B>Setting status	↓
	The count value is HRGCMER	PCNA<B>R2.CMEUCA<B>Setting status	↓
	The count value is HRGCMFR	PCNA<B>R2.CMFUCA<B>Setting status	↓
	The count value is SCMAR	PCNA<B>R2.SCMAUCA<B>Setting status	↓
	The count value is SCMBR	PCNA<B>R2.SCMBUCA<B>Setting status	↓
	The count value is 0	PCNA<B>R1.UDFCA<B>Setting status	↓
When the triangle count modes down	When external events occur	PCNA<B>R3.EVEyDCA<B>Setting status	↓
	Phase forced output	PHSCTL.PHSFORCA<B>Settings	↓
	The count value is HRGCMAR	PCNA<B>R1.CMADCA<B>Setting status	↓
	The count value is HRGCMBR	PCNA<B>R1.CMBDCA<B>Setting status	↓
	The count value is HRGCMER	PCNA<B>R3.CMEDCA<B>Setting status	↓
	The count value is HRGCMFR	PCNA<B>R3.CMFDCA<B>Setting status	↓
	The count value is SCMAR	PCNA<B>R3.SCMADCA<B>Setting status	↓
	The count value is SCMBR	PCNA<B>R3.SCMBDCA<B>Setting status	↓
	The count value is HRPERAR	PCNA<B>R1.OVFCA<B>Set Status	↓
When counting stops		PCNA<B>R1.STPCA<B>Setting status	Low

***Note**: When external events occur simultaneously, setting to 0 has higher priority than setting to 1 and higher priority than toggling.*

20 General Control Timer (Timer4)

20.1 Overview

General Control timer 4 (Timer4) is a timer module for three-phase motor control. It provides three-phase motor control schemes for various applications. The timer supports triangular wave and sawtooth wave modes and generates various PWM waveforms. Supporting cache function; Support EMB control. 1 unit of Timer 4 is carried in this product family.

20.2 Main Features

The basic functions and characteristics of Timer4 are shown in Table 20-1.

Table 20-1 Basic Functions And Features Of Timer4

Waveform mode	Sawtooth wave, triangular wave
Basic functions	Direction of increments and decrements
	Cache function
	Universal PWM output
	Generic compare match event
	Overflow, underflow, reload count match events
	Dedicated comparison matching event
	Hardware boot count
	EMB control
Interrupt type	Universal compare match interrupt
	Count cycle match interrupt
	Heavy load count matching interrupt
	Dedicated compare match interrupt

20.3Functional Description

20.3.1 Timer4 Block Diagram

As shown in Figure 20-1, the basic structure of the Universal Control timer Timer4 is described.

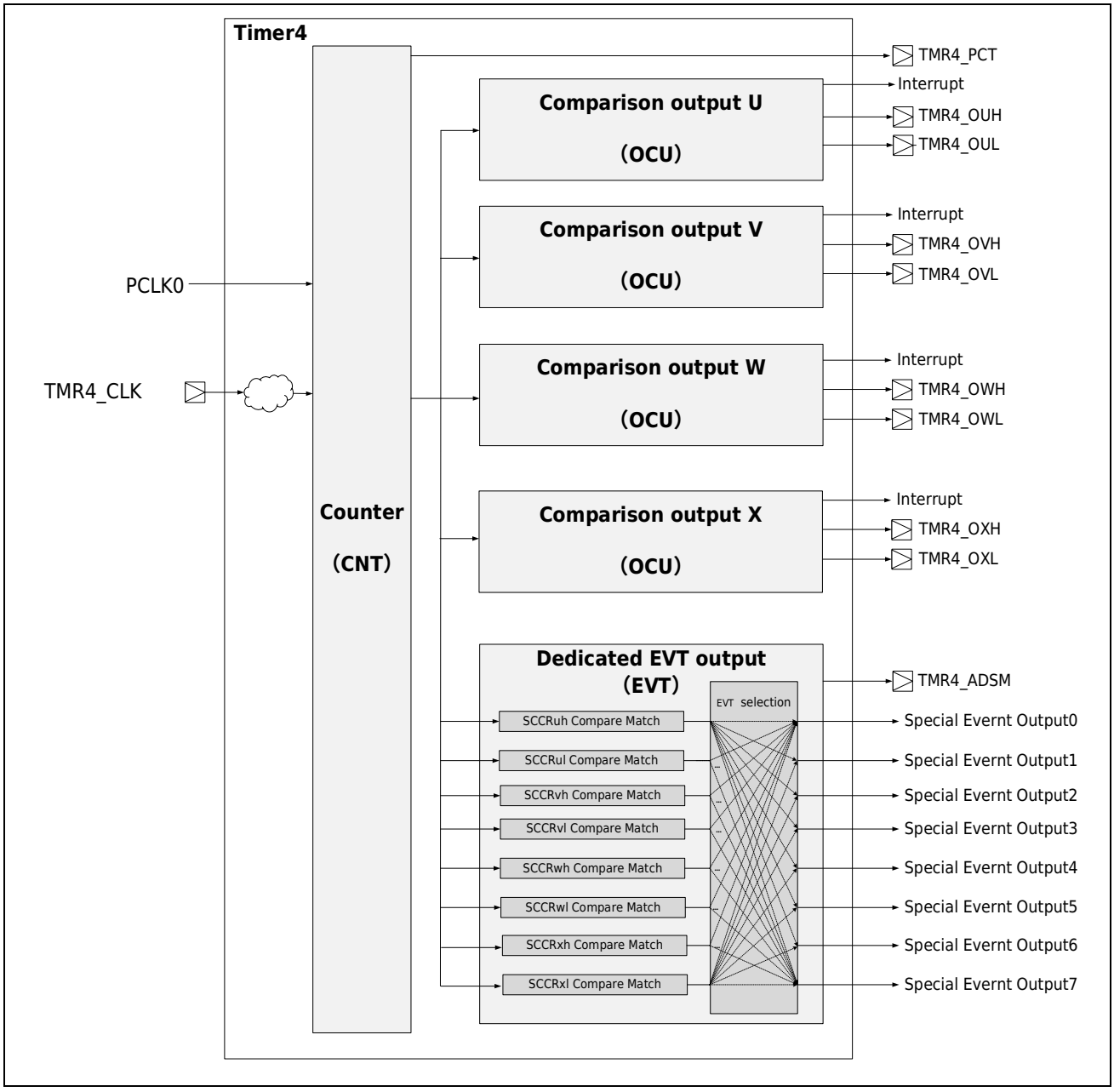


Figure 20-1 Timer4 Basic Block Diagram

Table 20-2 shows the list of input and output ports of Timer4.

Table 20-2 Timer4 port list

Port Name	Direction	Function
TMR4_CLK	Input	Counting clock input port
TMR4_PCT	Output	PWM cycle output monitoring
TMR4_ADSM	Output	Dedicated event output monitoring
TMR4_OUH	Output	PWM output port
TMR4_OUL		
TMR4_OVH		
TMR4_OVL		
TMR4_OWH		
TMR4_OWL		
TMR4_OXH		
TMR4_OXL		

20.3.2 Basic Action

20.3.2.1 Waveform Mode

Timer4 has two basic counting wave modes: Serrated wave mode and triangular wave mode. According to CCSR.MODE, two waveform modes can be implemented. Below Figure 20-2 and Figure 20-3 below are the waveform diagrams of sawtooth wave and triangular wave respectively.

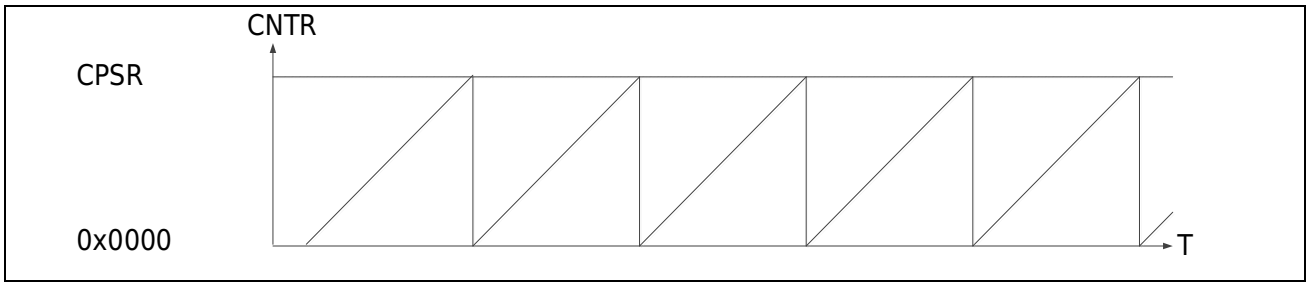


Figure 20-2 Timer4 Sawtooth Waveform

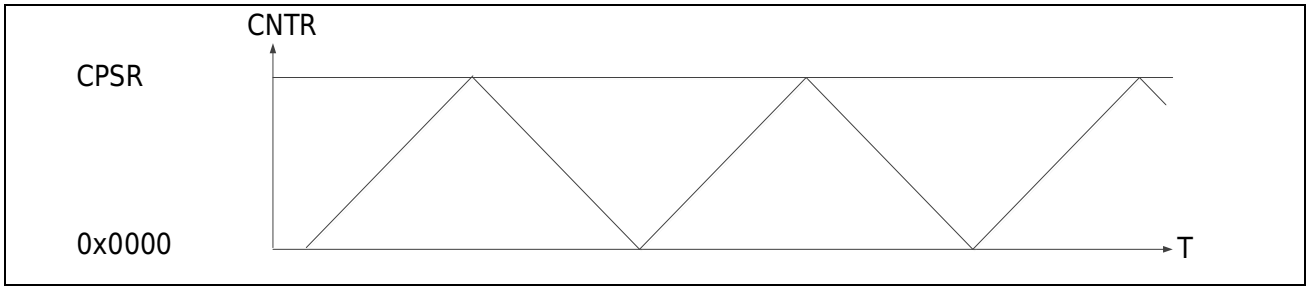


Figure 20-3 Timer4 Triangle Wave Waveform

20.3.2.2 Counting Action

1) The sawtooth wave counting operation and control flow are asFigure 20-4 shown.

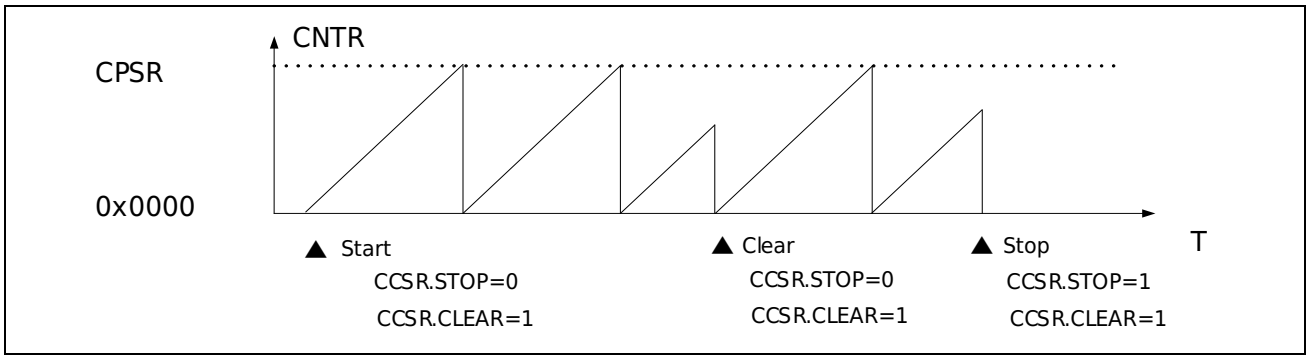


Figure 20-4 Timer4 Sawtooth Wave Mode Counting Action

1. Set mode CCSR.MODE = 0;
2. Sets the count peak CPSRregister;
3. Write CCSR.STOP = 0 and CCSR.CLEAR = 1, and the counter count value (CNTR) is initialized to 0x0000 and the counting operation is initiated. Counter values are counted incrementally from 0x0000. When the peak value (CPSR) is reached and the count value is returned to 0x0000, the operation is repeated in turn;
4. Counting cycle = (CPSR + 1) × Counting clock cycle
5. During counting, write CCSR.STOP = 0 and CCSR.CLEAR = 1 to initialize the count value to 0x0000 and continue counting operations;
6. During counting, write CCSR.STOP = 1 and CCSR.CLEAR = 1 to initialize the count value to 0x0000 and stop the counting operation.

2) The triangular wave counting operation and control flow are as Figure 20-5 shown.

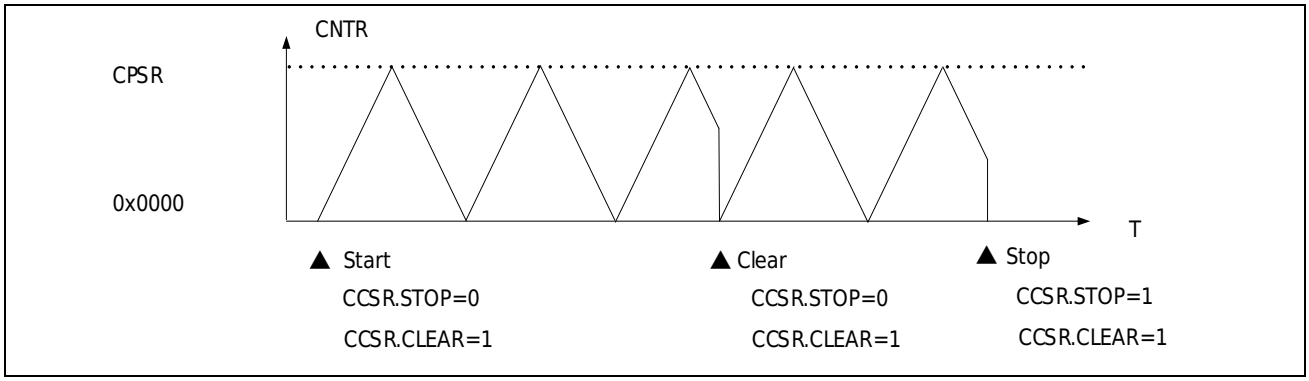


Figure 20-5 Timer4 Triangular Wave Mode Counting Action

1. Set mode CCSR.MODE = 1;
2. Sets the count peak CPSRregister;
3. Write CCSR.STOP = 0 and CCSR.CLEAR = 1, and the counter count value (CNTR) is initialized

to 0x0000 and the counting operation is initiated. 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. Count cycle = (CPSR) × 2 × count clock cycle;
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 counting, write CCSR.STOP = 1 and CCSR.CLEAR = 1 to initialize the count value to 0x0000 and stop the counting operation.

20.3.2.3 Comparison Output

1) The following Figure 20-6 is an example of the waveform output of the comparison output module in sawtooth wave count mode.

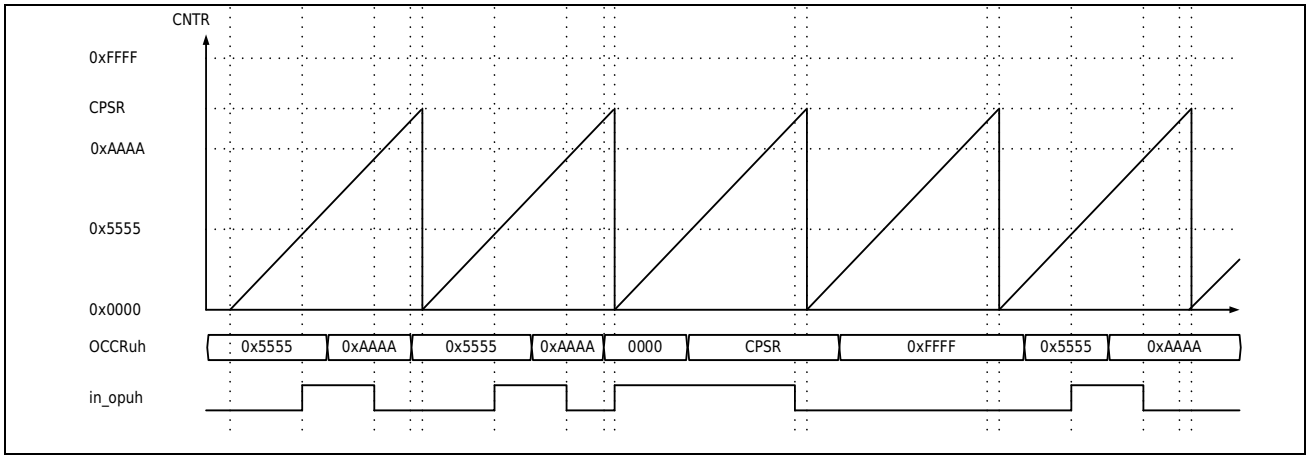


Figure 20-6 Waveform Output Example in Sawtooth Mode

2) The following Figure 20-7 is an example of the waveform output of the comparison output module in the triangle wave count mode.

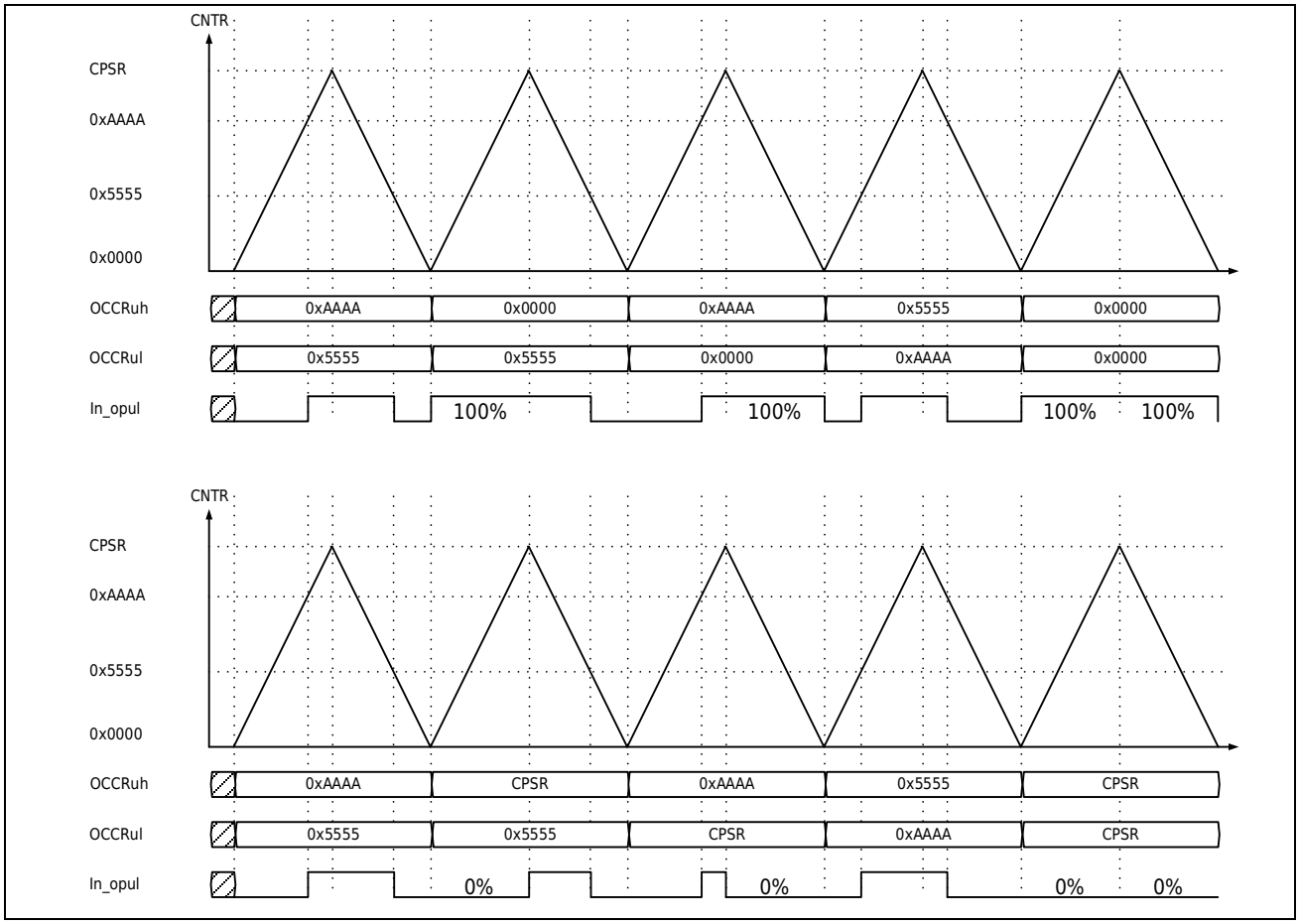


Figure 20-7 Triangle Wave Mode Waveform Output Example

20.3.3 Cache Function

Timer4's periodic reference register (CPSR), generic reference register (OCCR), generic mode control register (OCMR), dedicated reference register (SCCR) and dedicated mode control register (SCMR) all have caching capabilities.

20.3.3.1 Periodic Reference Register Cache Function

CPSR has a cache function register, and the written count peak data is first written into the buffer register. Data is transferred from the buffer register to the CPSRregister under the following conditions.

a) When buffer is disabled (CCSR.BUFEN = 0), write data is immediately transferred from buffer register to CPSRregister.

b) Data is transferred from the buffer register to the CPSRregister when the buffer function enables (CCSR.BUFEN = 1), when the counter stops (CCSR.STOP = 1), or when the counter count value is "0x0000".

Note:

– *When reading data from a CPSR, it is not the value of the CPSR buffer register, but the value of the CPSR register. When the buffer function is enabled, the value read before the transfer completes is not the most recently written value, but the last written CPSR value.*

Figure 20-8 As shown, the operation of modifying the count peak CPSR when sawtooth mode and the buffer function is disabled.

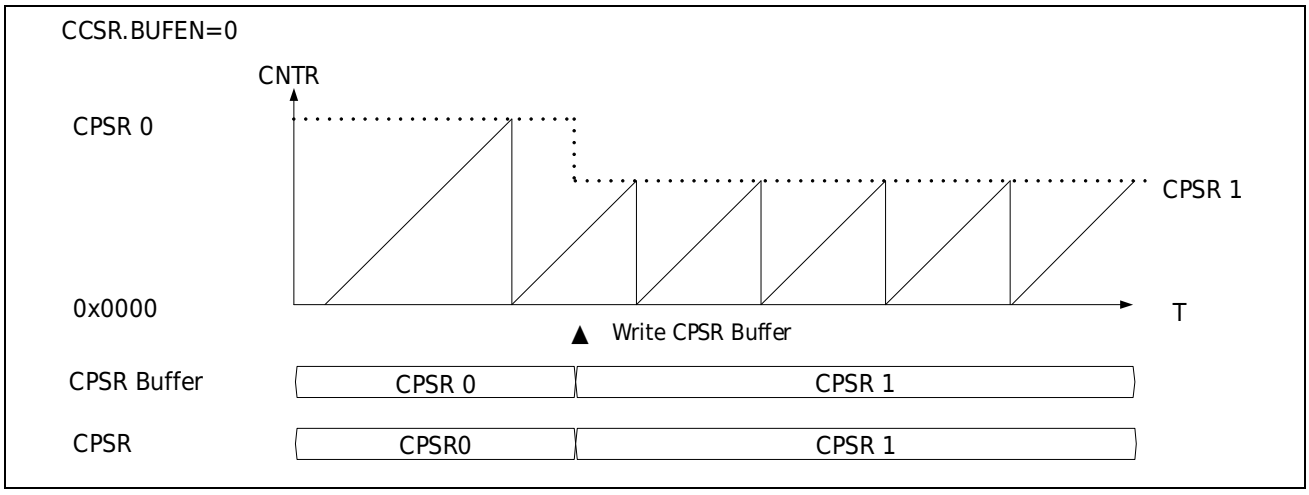


Figure 20-8 Modify The Sawtooth Count Period When The Cache Is Invalid

Note:

– *When the buffer function is disabled, write data is immediately transferred from the buffer register to the CPSRcounter, and the count period changes immediately after the write operation is completed. In this case, if the value written is less than the current counter count, the counter continues to count incrementally until it reaches "0xFFFF", which*

requires special attention.

Figure 20-9 As shown, the operation of modifying the count peak value CPSR when the sawtooth mode is enabled and the buffer function is enabled.

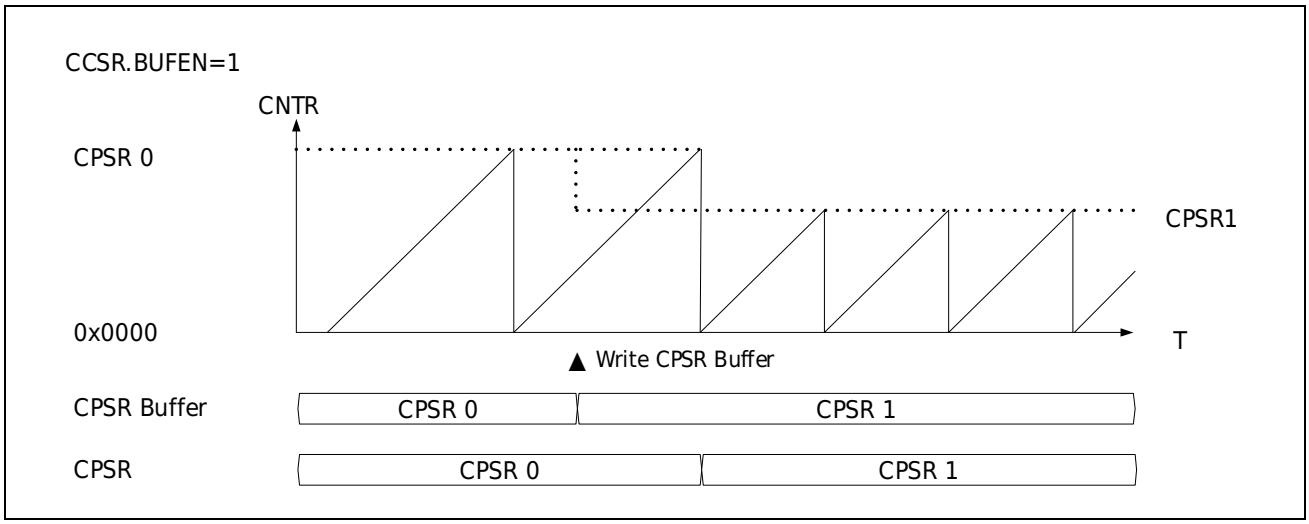


Figure 20-9 Modify The Sawtooth Count Period When The Cache Is Enabled

As shown in the figure, when the buffer function is enabled, the data written is transferred from the buffer register to the CPSR register when the counter stops or when the counter count value is "0x0000".

Figure 20-10 As shown, the operation of modifying the count peak CPSR when the triangle wave mode and the buffer function is enabled.

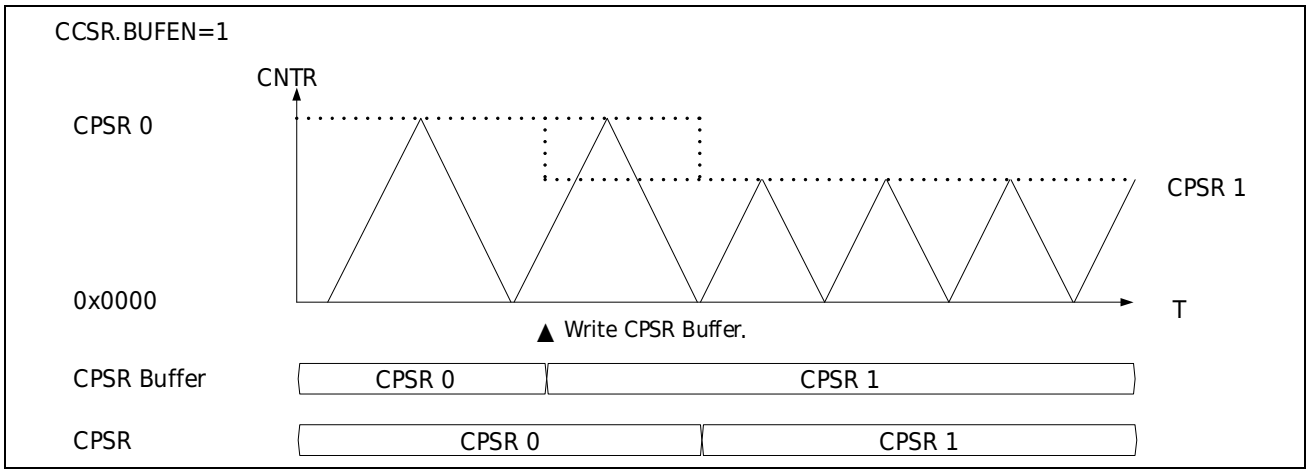


Figure 20-10 Modify The Triangular Wave Count Period When The Cache Is Enabled

As shown in the figure, in triangulation mode, when the buffer function is enabled, the written data is transferred from the buffer register to the CPSRregister when the counter stops or the next counter count value is "0x0000". The change in the counter cycle begins after the write operation completes the next counter cycle.

20.3.3.2 Universal Comparison Register Cache Function

Both the common baseline value register (OCCR) and the common mode control register (OCMR) have the buffer register function. When the buffer function is enabled, the transfer is loaded to the OCCR and OCMRregister at the specified transmission time. The OCCR buffer function can be used to change the comparison value synchronously during counting, and the OCMR buffer function can be used to change the internal PWM output synchronously during counting.

a) When the output compares the link transfer barring with the counter interval response function, the buffer value is loaded to register at the set count state. The load at this time is independent of the counter interval counter.

Figure 20-11 As shown, the waveform (x = L or H) is when the universal output compare OCCR buffer function is enabled, the counter is loaded at zero value (OCER.CxBUFEN = 0b01), and the counter cycle interval response link is disabled (OCER.LMCx = 0).

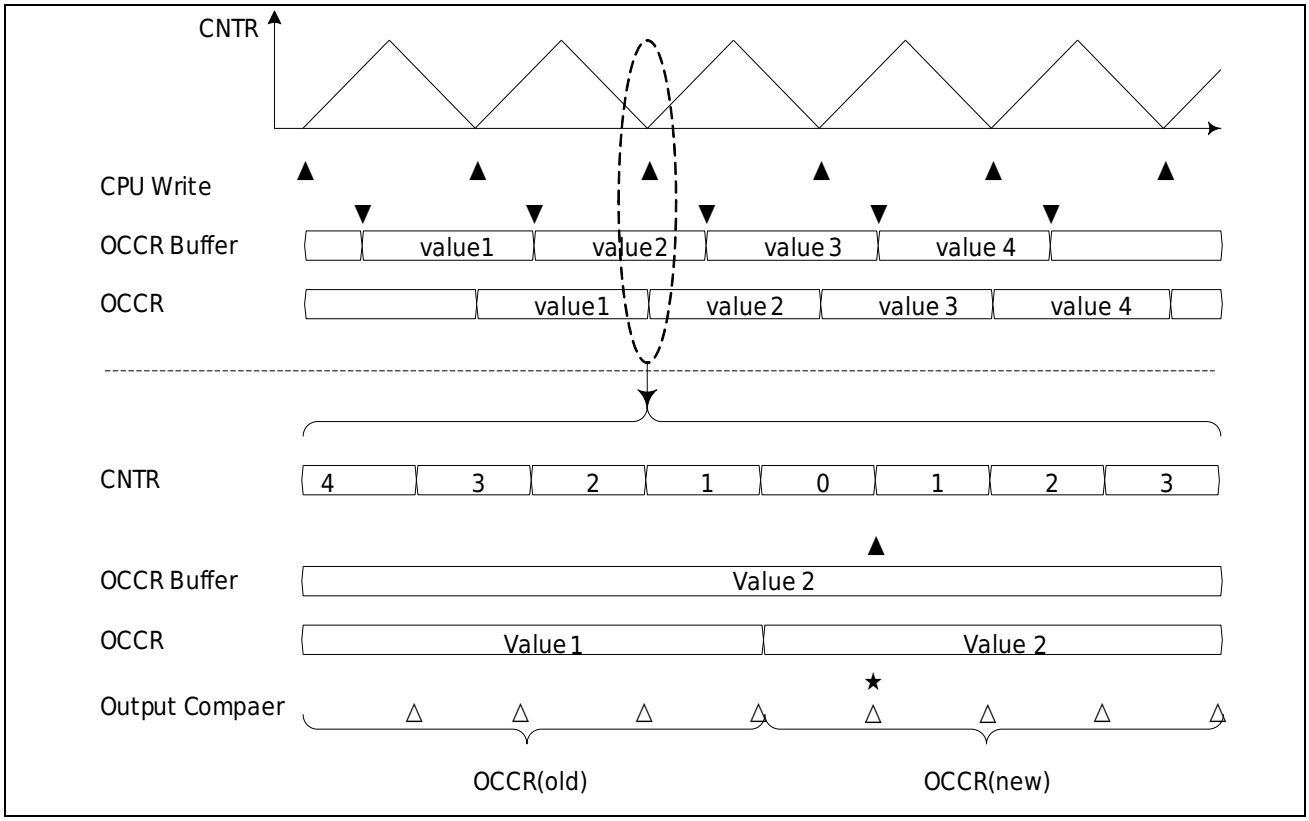


Figure 20-11 OCCR Buffer Data Transmission (When the Cycle Interval Response Link Disabled)

The first half of the diagram is a global diagram, and the second half is an enlarged diagram during the transfer operation.

The counter is in the triangular wave counting mode, and the underflow interrupt is generated at the moment of the mark ▲ (counting valley point). At ▼ time, the CPU rewrites the OCCR register, and the written data is stored in the OCCR buffer register. After that, when the counter count is 0x0000, the data is loaded from the buffer register to the OCCR register, and the interrupt flag IRQZF is generated.

At the △ moment, the output comparison is performed to change the PWM output and the set OCSR.OCFx bit (x = L or H) according to the event that the set OCCR register value matches the count value. After the time (count = 0x0000), the port output performs the operation based on the new OCCR data. Perform the operation according to the original OCCR data before the time.

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).

b) When the counter interval response link is enabled, the buffer register transfer operation is performed when the buffer value is in the set count state and the period interval counter count value is 0.

Figure 20-12 OCCR Buffered Data Transfer (OCER.CxBUFEN=0b01) shows the waveform (x = L or H) when the Universal output compares the OCCR buffer function enabled, the counter zero value loaded (OCER.CxBUFEN = 01), and the counter interval response link enabled (OCER.LMCx = 1).

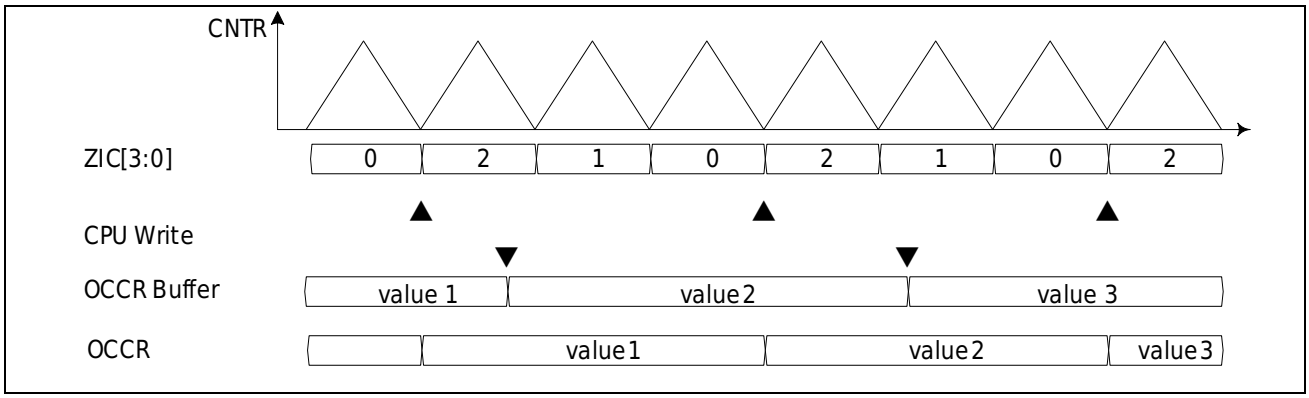


Figure 20-12 OCCR Buffer Data Transmission (Period Interval Response Link Enable)

The counter is in the triangular wave counting mode, the cycle interval counter (CVPR.ZIC) counts down from 2 to 0, and the underflow interrupt is generated at the moment of the mark ▲ (counting valley point). At?, the CPU rewrites the SCCRregister and the data written 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 channel link operation mode, enabling both OCCRuh and OCCRul buffers can produce various PWM output waveforms. Figure 20-13 shows that when the output compare registers

OCCRuh and OCCRul remain unchanged, changing the value of the OCMR register can make in_opul output different waveform signals.

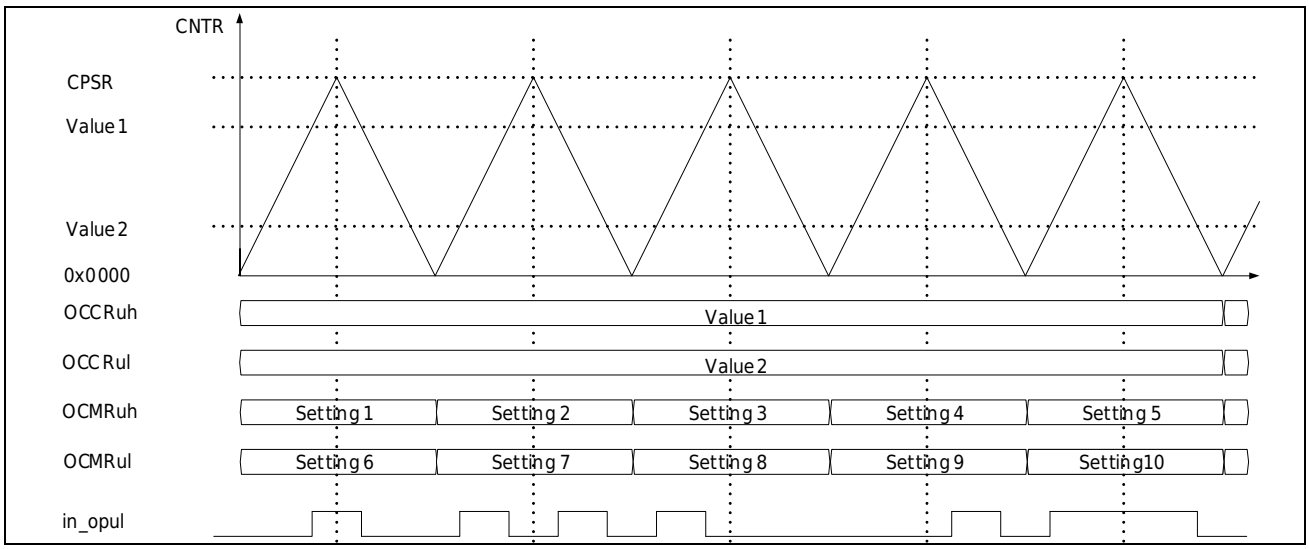


Figure 20-13 Output Compare Buffer Data Transfer (OCMR Buffer Enabled)

20.3.3.3 Special Comparison Register Cache Function

Both the dedicated reference register (SCCR) and the dedicated mode control register (SCMR) have a buffer work register. When the buffer function is enabled, the CPU writes the values of the SCCR and SCMR buffer registers to the SCCR and SCMRregisters in the set counter state.

a) When the counter interval response link transfer is disabled, the buffer transfer operation is only related to the counter state and is not affected by the counter interval counter.

Figure 20-14 As shown in SCCR Buffer Transport Operation (Periodic Interval Response Link Transport Disabled), enables the register buffer function to disable the transfer of the counter periodic interval response link (SCSR.LMC = 0) and the transfer of the counter zero value (SCSR.BUFEN = 01) to SCCR and SCMR register.

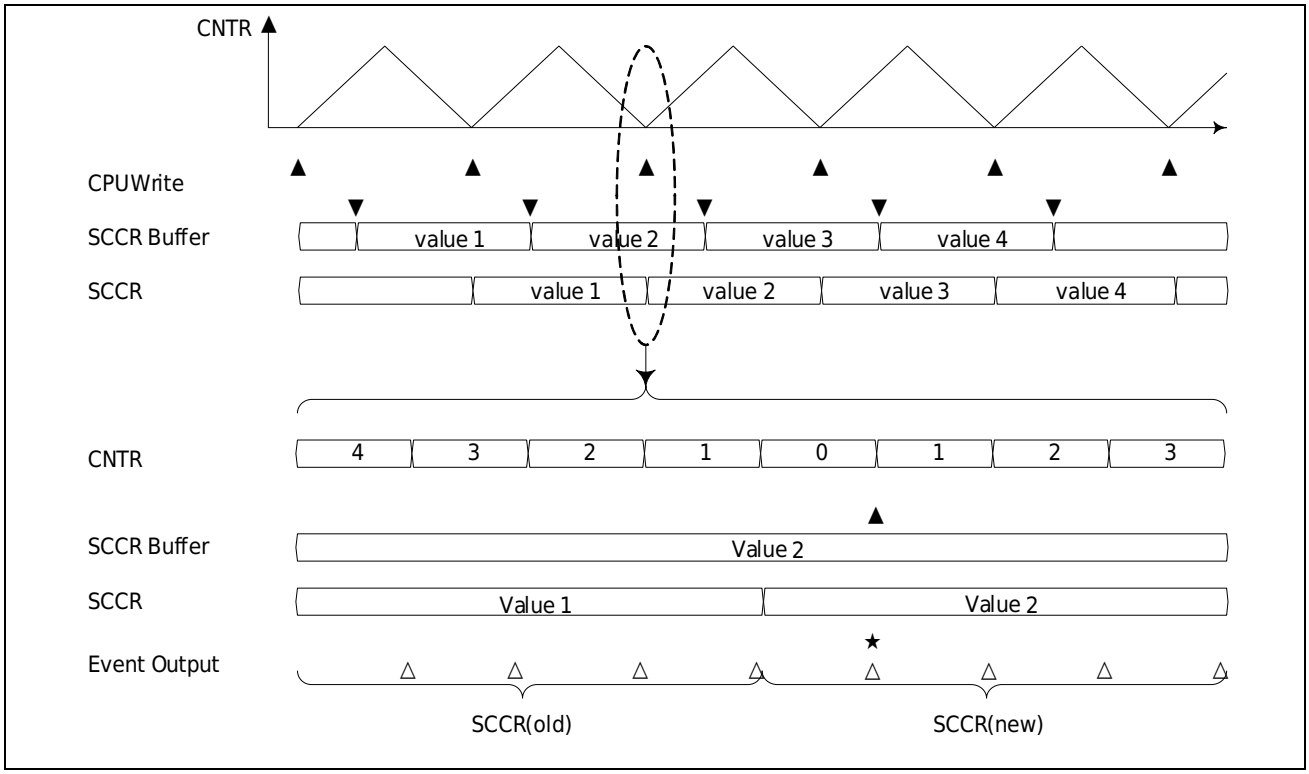


Figure 20-14 SCCR Buffer Transmission Operation (Period Interval Response Link Transmission Disabled)

The upper half of the diagram is a global diagram, and the lower half of the diagram is a partial magnification of the buffer register transfer operation.

The counter is in the triangular wave counting mode, and the underflow interrupt is generated at the moment of the mark ▲ (counting valley point). At ▼ time, the CPU rewrites the SCCR register, and the written data is stored in the SCCR buffer register. After that, when the counter count is 0x0000, the data loading operation is performed, from the buffer register to the SCCRregister, and the interrupt flag IRQZF is generated.

At △ time, the comparison operation is performed according to the set SCCRregister value and the count value. After the time (count = 0x0000), the dedicated event output signal performs the operation based on the new SCCR data. Perform the operation according to the SCCR data before the time.

The diagram shows the SCCR buffer register transfer operation at the counting valley point, and the SCMR buffer register transfer operation is similar to the SCCR buffer register transfer operation. Similarly, the transfer operation at the counting peak is similar. The new data takes effect immediately after the transmission time (the new write data sets the dedicated event output signal and interrupt flag).

b) When the counter interval response link is enabled, the buffer register transfer operation is performed when the buffer value is in the set count state and the period interval counter count value is 0.

As shown in Figure 20-15, SCCR Buffer Transfer Operation (Periodic Interval Response Link Transfer Enabled) shows a schematic diagram of enabling SCCR buffering, loading at zero counter values (SCSR.BUFEN = 01), and enabling transmissions across response links during the counter cycle (SCSR.LMC = 1).

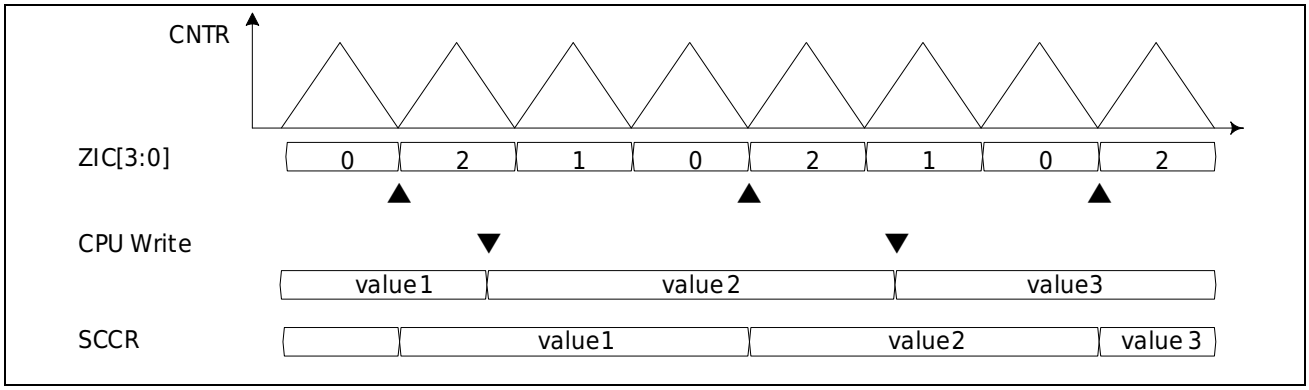


Figure 20-15 SCCR Buffer Transmission Operation (When Periodic Interval Response Link Transmission Is Enabled)

The counter is in the triangular wave counting mode, the cycle interval counter (CVPR.ZIC) counts down from 2 to 0, and the underflow interrupt is generated at the moment of the mark ▲ (counting valley point). At?, the CPU rewrites the SCCRregister and the data written is stored in the SCCR 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 SCCRregister, and the interrupt flag IRQZF is generated.

20.3.4 Universal PWM Output

In the rest of this section, 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, the wildcard "*" represents u, v, w, and x; when internal output signals in_op*h and in_op*l are mentioned, the wildcard "*" represents u, v, w, and x; when ports TMR4_O*H and TMR4_O*L are mentioned, the wildcard "*" represents U, V, W, and X.

20.3.4.1 Independent PWM Output

In the pass-through mode (POCR.PWMMD=00), different PWM outputs can be achieved by setting the reference values of OCCR*h and OCCR*l and the port output states of OCMR*h and OCMR*l respectively. At this point, the PWM output of each port is independently controlled. Before using the independent PWM output mode, you need to set the MOE, OE*H, OE*L bits (register PSCR) to enable the output of Timer4. For detailed description, please refer to [PWM Status Control Register (TMR4_PSCR)]. Figure 20-16 and Figure 20-17 are examples of independent PWM outputs for unit 1's sawtooth and triangle waves, respectively.

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 common comparison reference register (OCCR*h, OCCR*l) is directly output to the corresponding port (TMR4_O*H, TMR4_O*L).*

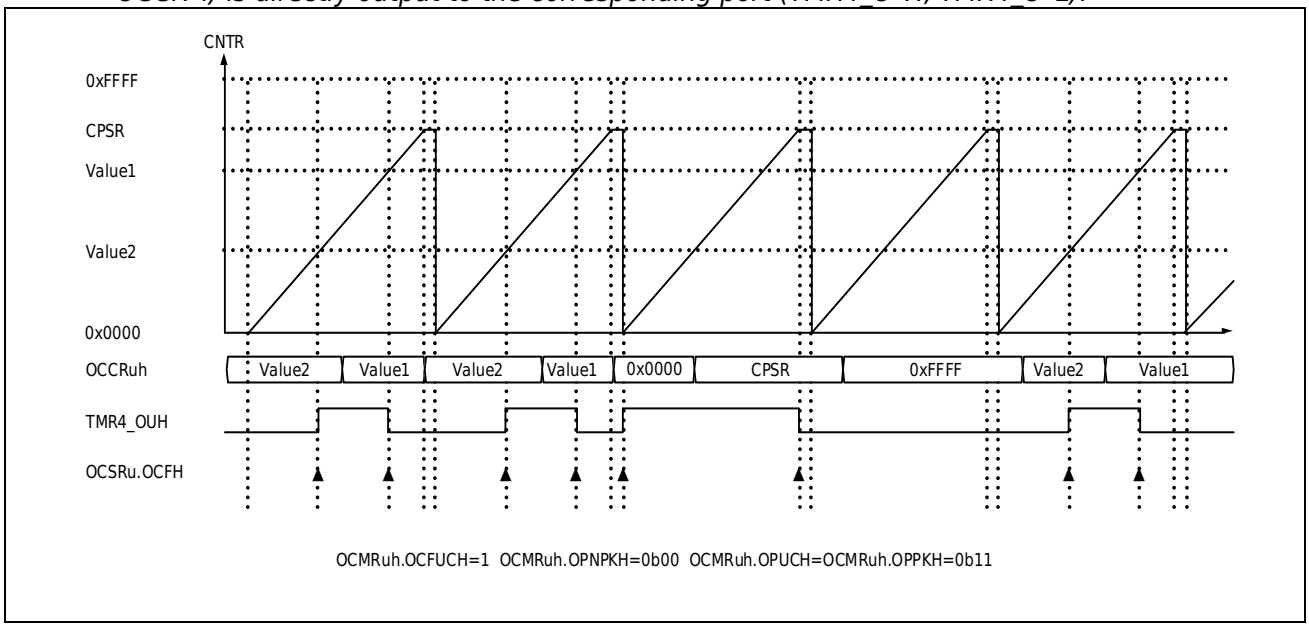


Figure 20-16 Example of Independent PWM Output of Sawtooth Wave

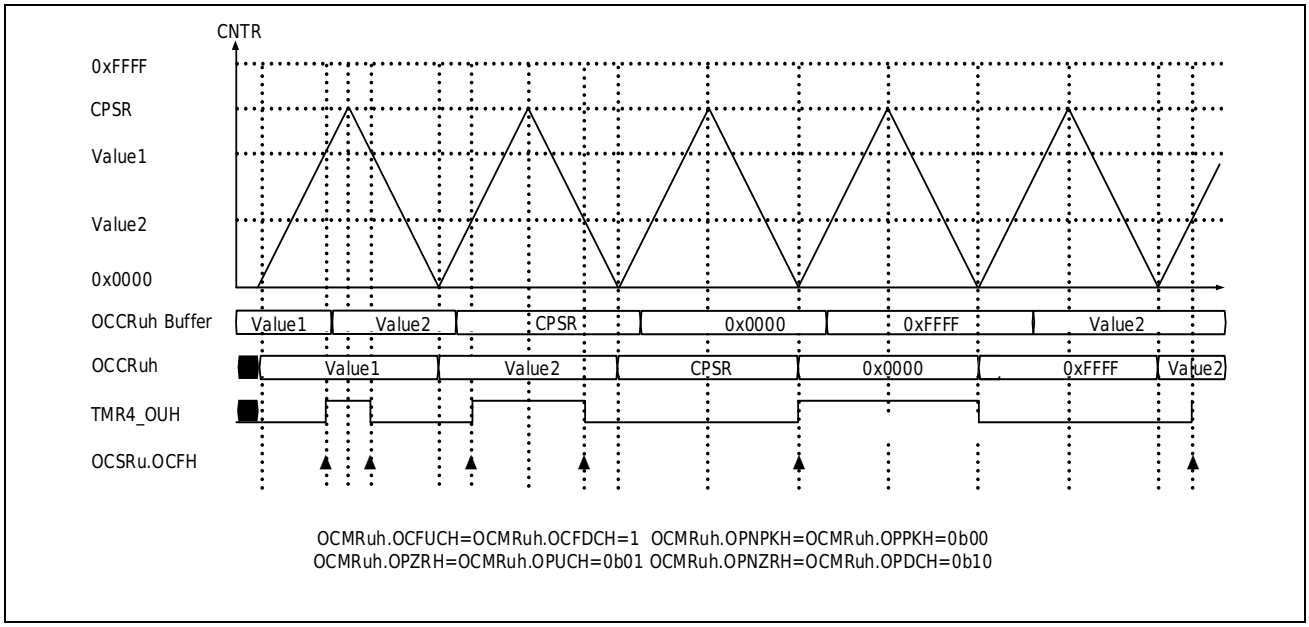


Figure 20-17 Example of Independent PWM Output of Triangle Wave

20.3.4.2 Extended PWM Output

In pass-through mode (POCR.PWMMD = 00), 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 value, 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 (register PSCR) to enable the output of Timer4. For detailed description, please refer to [PWM Status Control Register (TMR4_PSCR)]. Figure 20-18 shown is the PWM output of TMR4_OUH and TMR4_OUL ports in this mode.

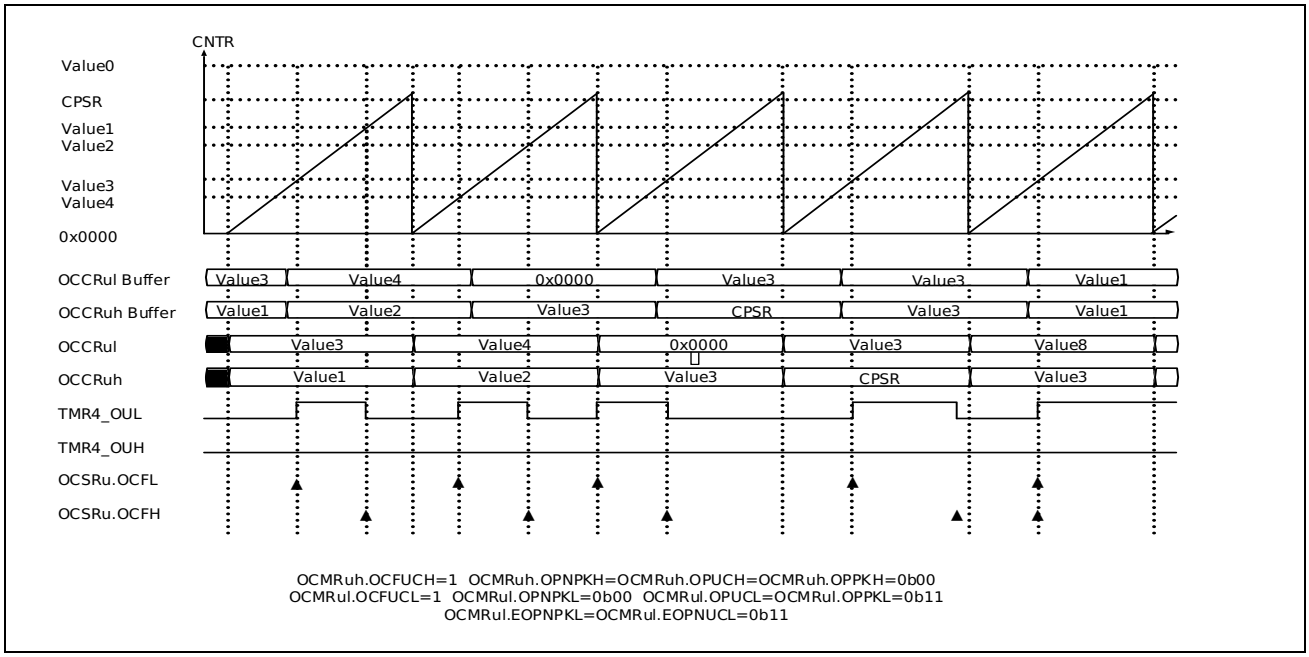


Figure 20-18 Sawtooth Wave Extended PWM Output

Note:

– *In independent PWM output mode, the port status of TMR4_O*L is determined by bit15~bit4 of the OCMR*l register and is only related to the OCCR*l reference value.*

20.3.4.3 Complementary PWM output

Software Setting to Realize Complementary PWM Output

In the pass-through mode (POCR.PWMMD=00), directly set the OCCR*h and OCCR*l reference values to realize a pair of complementary PWM waveforms output to the port. The same setting of 4 groups of ports can realize 4-phase complementary PWM output. As shown in Figure 20-19.

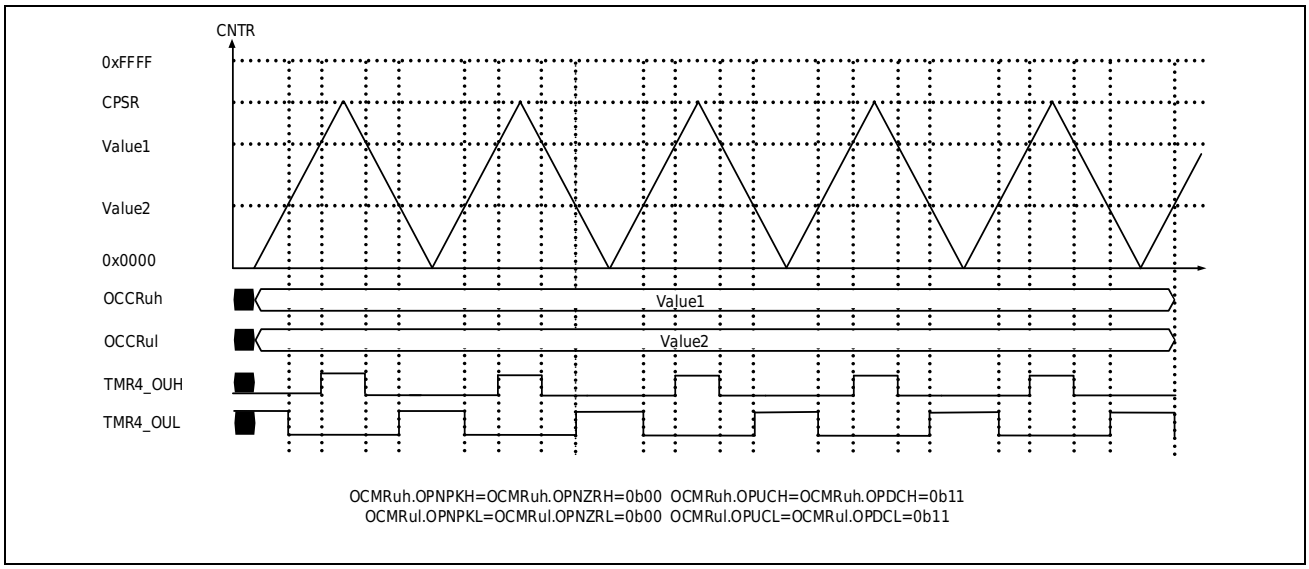


Figure 20-19 Software Complementary PWM Output

Realization of Complementary PWM Output by Hardware Setting

In the dead-band timer mode (POCR.PWMMD=01), the internal output signal (in_op*l) generated by the comparison match of the value of the common comparison reference register (OCCR*l) and the set value of the PWM dead-band control register (PDAR/PDBR) are offset by timing to achieve complementary PWM output in hardware.

In this mode, the polarity of the output of the TMR4_O*H port is the same as in_op*l, and the polarity of the output of the TMR4_O*L port is opposite to in_op*l.

If the rising edge of in_op*l is detected, the output of TMR4_O*L becomes low, 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. If the falling edge of in_op*l is detected, the output of TMR4_O*H becomes low, 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.

By setting PWM dead zone to control registerPDAR and PDBR, output rising and falling dead zone time can be set accordingly.

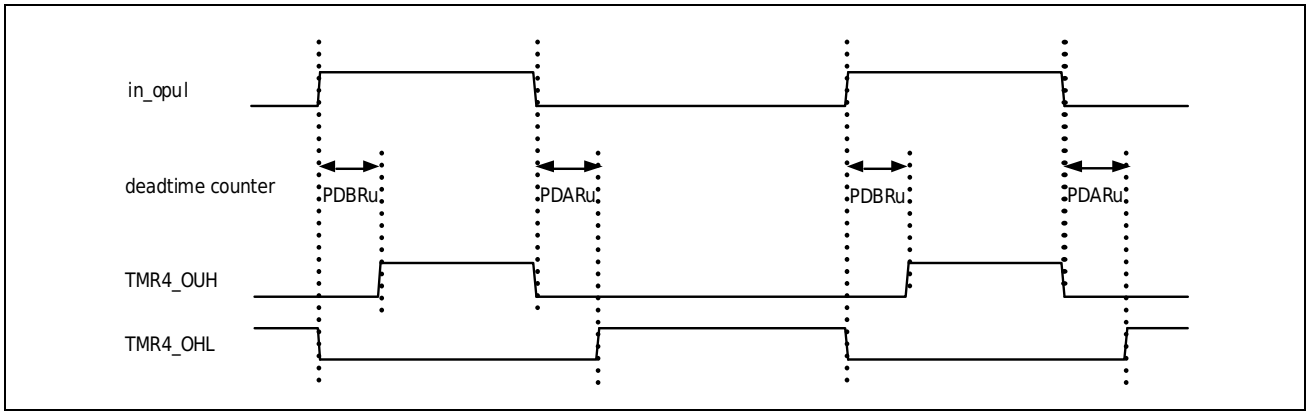


Figure 20-20 Complementary PWM Output In Dead-Band Timer Mode

When the high-level pulse width of in_op*l is less than the dead time set by PDBR, only the TMR4_O*L output becomes low. The condition for the TMR4_O*L output level to become high is when the dead time set by the PDAR register passes after the falling edge of in_op*l. In this case, the TMR4_O*H output will remain low continuously.

When the low-level pulse width of in_op*l is less than the dead time set by PDAR, only the TMR4_O*H output becomes low. The condition for the TMR4_O*H output level to become high is when the dead time set by the PDBR register passes after the rising edge of in_op*l. In this case, the TMR4_O*L output will remain low continuously. As shown in Figure 20-21 below.

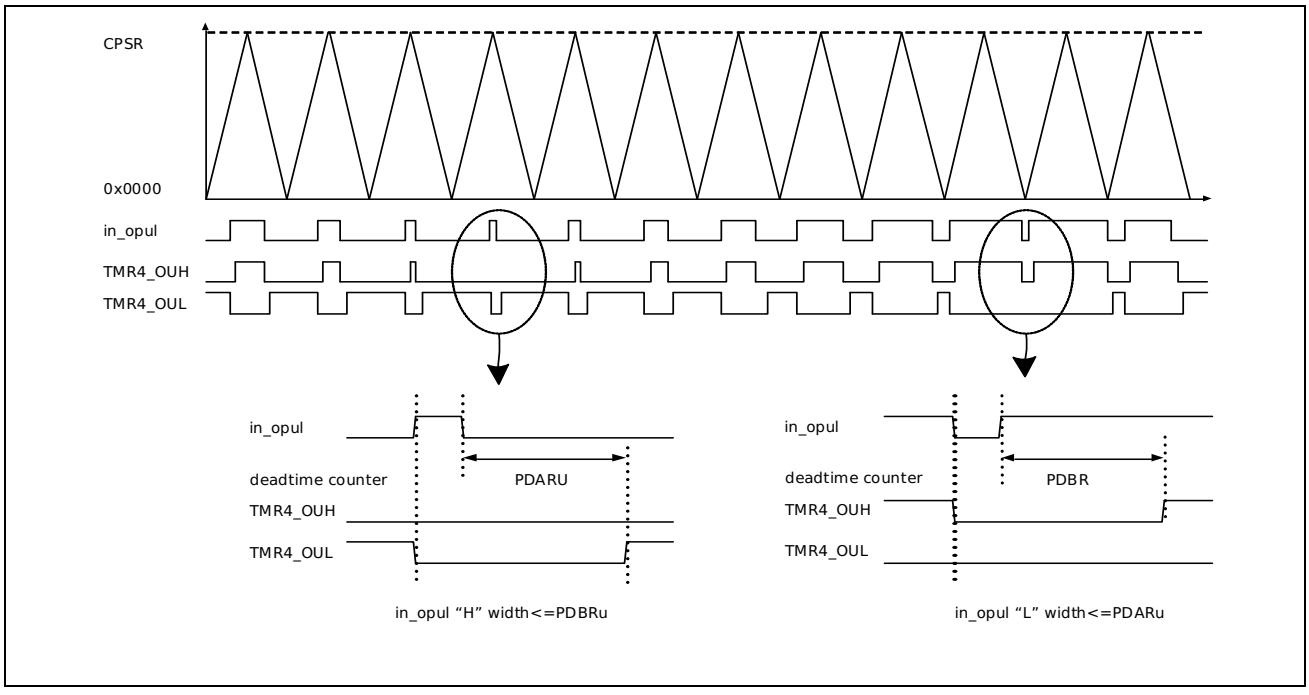


Figure 20-21 Waveform Output In Dead-Band Timer Mode When The Pulse Width Is Abnormal

On the basis of the above hardware implementation of the dead zone output mode, the pulse width of the internal comparison match signal (in_op*l) can also be monitored, thereby realizing the filtering control of the in_op*l signal. This in_op*l dead-band output implementation with pulse width filtering is called dead-band counter filtering mode (POCR.PWMMD=10).

In dead-zone counter filtering mode, the filter width is determined by the set value of PWM filter control register (PFSRn). When the pulse width of in_op*l is greater than the time set by register PFSRn, the filter counter delays the output of the in_op*l signal after the time set by PFSR, and then generates a complementary PWM output in the manner described by the dead-zone timer mode.

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 the PFSR, the TMR4_O*L output becomes low, 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. If the falling edge of the signal in_op*l is detected, the filter counter loads the PFSR register value and starts measuring 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 the time set by the PFSR, the TMR4_O*H output becomes low, 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. When the level pulse width of in_op*l is less than the time set by register PFSR, the outputs TMR4_O*H and TMR4_O*L will remain unchanged.

Figure 20-22 shown is an example of complementary PWM output in dead-band counter filter mode.

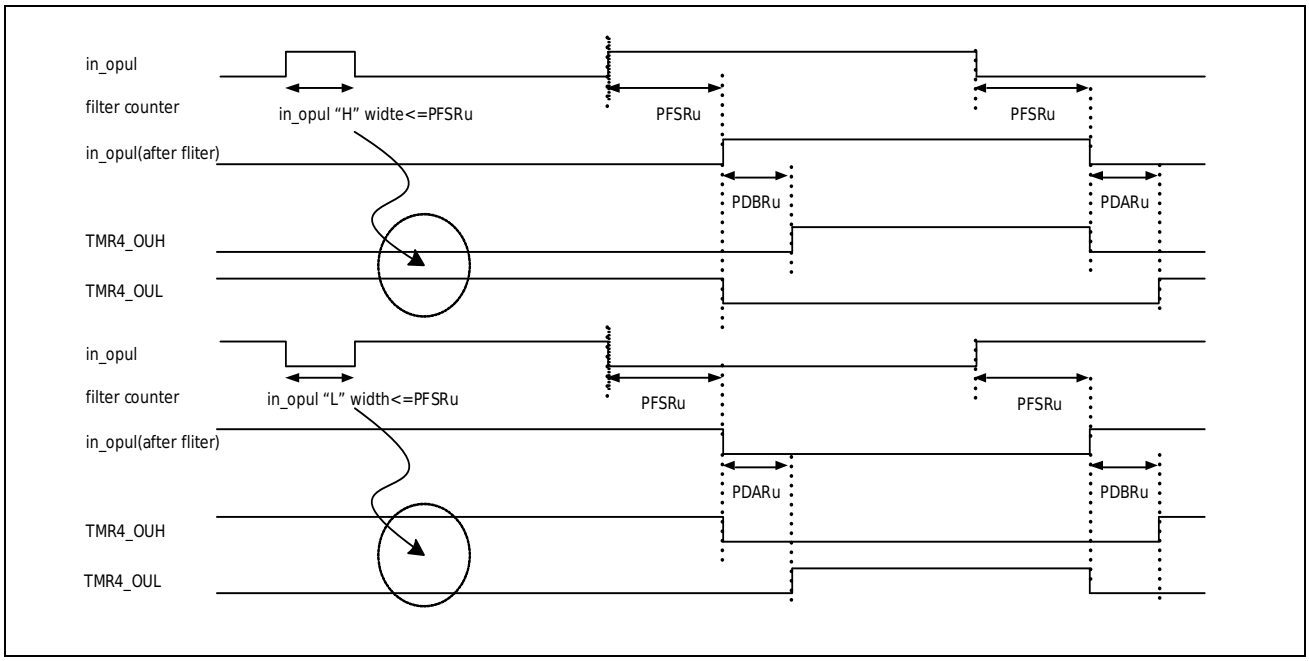


Figure 20-22 Complementary PWM Output in Dead-band Timer Filtering Mode

Note:

– Before using software or hardware complementary PWM output mode, you need to set the MOE, OE*H, OE*L bits (register PSCR) to enable the output of Timer4. For detailed description, please refer to [*PWM Status Control Register* (TMR4_PSCR)].

20.3.5 Periodic interval response

The underflow interrupt mask counter is used to reduce (mask) the number of times the underflow flag bit (CCSR.IRQZF) is set. The underflow interrupt mask counter (CVPR.ZIC[3:0]) operates as a down counter and loads the value set by CVPR.ZIM[3:0] at the beginning, when CVPR.ZIC[3:0]=" When 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 bit (CCSR.IRQPF) is set. The overflow interrupt mask counter (CVPR.PIC[3:0]) operates as a down counter and is loaded with the value set by CVPR.PIM[3:0] at the beginning. When CVPR.PIC[3:0]= 0, the overflow flag (CCSR.IRQPF) is set to 1.

As shownFigure 20-23 as shown below, it is the timing diagram of setting IRQZF and IRQPF at the time of periodic interval response.

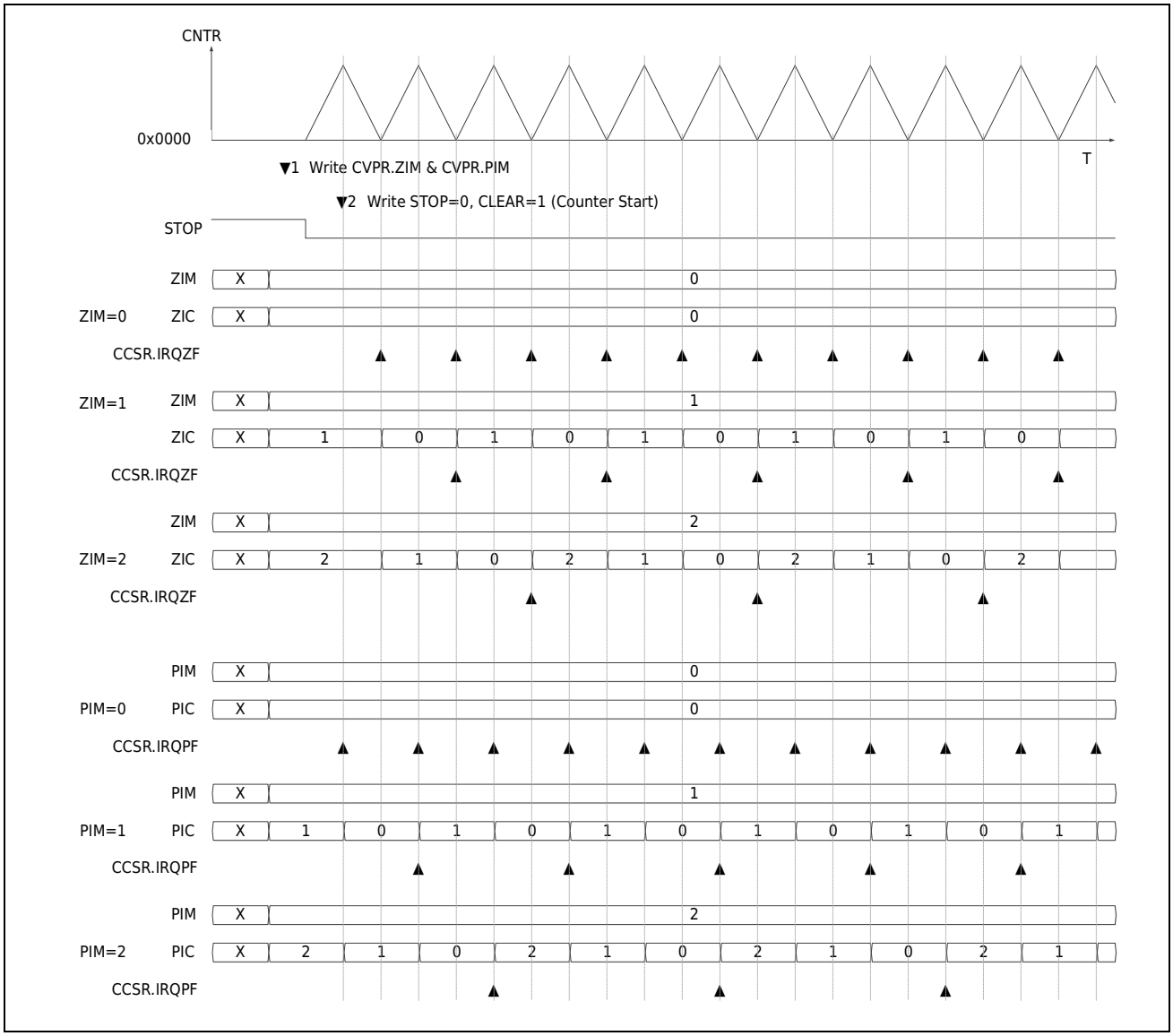


Figure 20-23 Cycle Interval Response Timing Diagram

- When counter stops, it writes CVPR.ZIM and CVPR.PIM initialization values, which are immediately reflected in internal counter (CVPR.ZIC, CVPR.PIC).
- Initialize and initiate counter (STOP = 0 and CLEAR = 1), counter starts from zero after bus reset or after software initialize CLEAR = 1, at which point the CCSR.IRQZF flag is not set immediately, and then whenever the count value of interrupt shielded counter is 0x0000 and the count value of counter is 0x0000 and CPSR, the flag "is the time when CCSR.IRQZF or SR.CCIRQPF is set.

Note:

- In the counter run, CVPR.ZIM and CVPR.PIM are written, and this setting does not immediately react to interrupt shielded counter (CVPR.ZIC and CVPR.PIC). If a soft reset is written (CLEAR = 1), the written CVPR.ZIM and CVPR.PIM values are immediately loaded as the initial value of the interrupt mask counter.

A comparison match event (dedicated event output) for a dedicated baseline register (SCCRm) also has a periodic interval response function. Shown as Figure 20-24 shown below is the periodic interval response output graph of the dedicated event output signal.

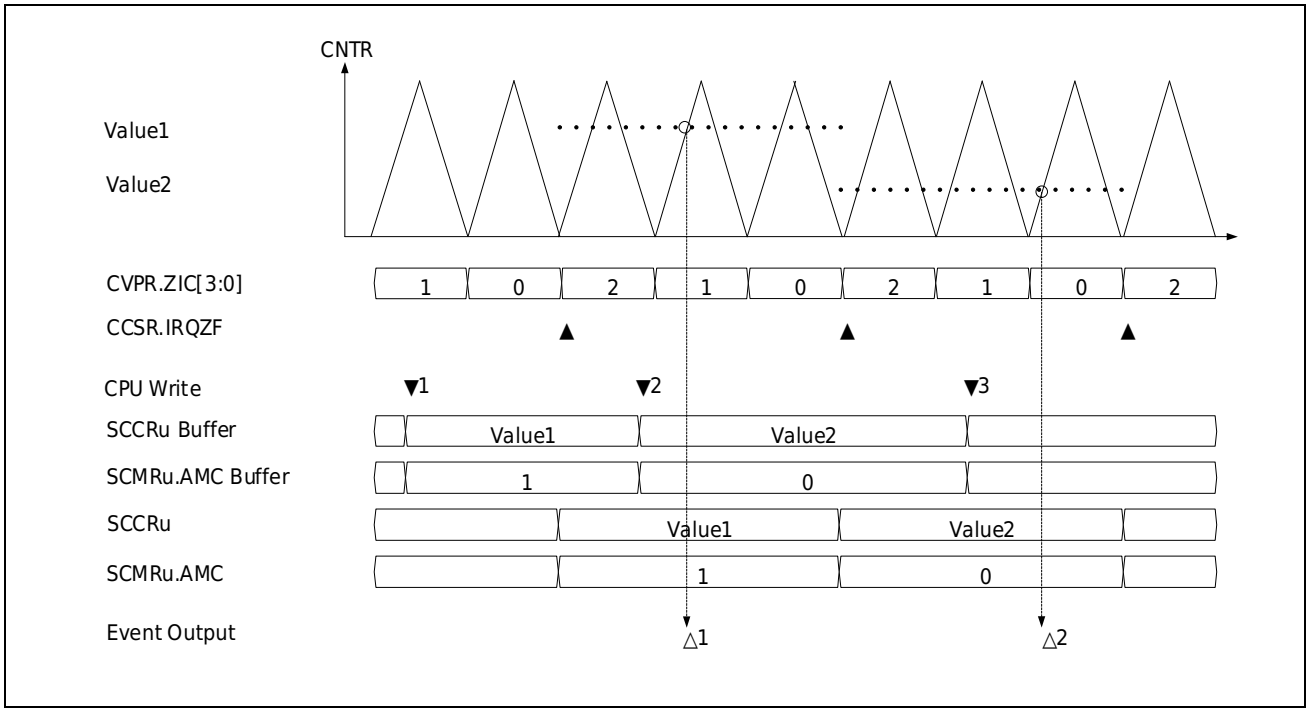


Figure 20-24 Dedicated Event Output Signal Periodic Interval Response Output

Counter is a triangular wave count mode, and the underflow interrupt shielded counter (CVPR.ZIC) is decremented from 2 to 0. Underflow interrupt occurs at the moment▲.

Write value 1 to SCCRu buffer register at time ▼1. Meanwhile, MZCE = 1, MPCE = 0, AMC = 0001 write SCMRu buffer register. After that, the buffer registerSCCRu and SCMRu transfer operations are performed at the moment. Because MZCE = 1, AMC = 1 and SCCRu = Value 1, special event output signal is set at time △ 1.

At time 2, write value 2 to SCCRu buffer register. Meanwhile, MZCE = 1, MPCE = 0, AMC = 0000 write SCMRu buffer register. After that, the buffer registerSCCRu and SCMRu transfer operations are performed at the moment. Because MZCE = 1, AMC = 0 and SCCRu = Value 2, special event output signal is set at time △ 2.

20.3.6 EMB Control

Each Timer 4 unit has an output invalid event interface connected to the EMB event output from the EMB module. The abnormal condition events selected on this interface can be set from the EMB side (see the [Emergency Brake Module (EMB)] section).

During the normal output period of the four PWM ports in each unit, if an 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 becomes 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, and an EMB event occurs during the normal output of the TMR4_OUH port of Timer4, the output on the TMR4_OUH port becomes a high-impedance state.

After the condition that triggers the EMB exception event invalid, the user can clear the EMB exception event by writing the EMB status reset register (EMB_STATCLR0). If PSCR.AOE=1 at this time, the PSCR.MOE bit will be automatically set to 1 by hardware at the next count underflow point (the underflow point here is the actual count zero point of the counter, which is not affected by the set value of the register CVPR), and then the PWM port will resume normal output; if PSCR.AOE=0 at this time, the software needs to set PSCR.MOE to restore the PWM port to normal output.

It should be noted that the hardware controls the PWM output enable using the asynchronous signal of MOE. When the value of MOE changes, it takes several clock cycles to read the actual value on the bus. (After an EMB exception event occurs, it is necessary to wait for 4 Timer4 bus clock cycles, and after the EMB exception event is released, it is necessary to wait for 6 Timer4 bus clock cycles).

For detailed operation methods of EMB, please refer to the description in the [Emergency Brake Module (EMB)] section.

Note:

When the event release mode of EMB is configured as state release (configured by EMB_RLSSEL register), and the event source of EMB is configured as external port or the comparison result of internal voltage comparator valid, it must be ensured that the level width of the comparison result of external port and internal voltage comparator is greater than 3 count clock cycles, otherwise a glitch of up to 2 count clock cycles will appear on PWM output. When the event release mode of EMB is configured as software release (configured through the EMB_RLSSEL register), and the event source of EMB is configured as PWM output port in phase, a glitch of up to 2 count clock cycles will occur before PWM in-phase output occurs and PWM enters the safe state.

20.3.7 Monitoring Output

Each Timer4 unit has a PWM period output monitoring port TMR4_PCT and a dedicated 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 counting status of the counter. In triangle wave mode, each time the count reaches a peak or a trough, a flip occurs on the monitoring port TMR4_PCT. Figure 18-25 is an example of counting direction signal.

The period output monitoring function of the TMR4_PCT port is only used when the count clock is not divided (CCSR.CKDIV = 0x0).

The dedicated event output monitoring port TMR4_ADSM can be used to monitor the dedicated comparison events of the internal counter. The dedicated comparison output channel to be monitored is selected through the SCER.EVTRS bit of the dedicated extended control register (SCER). When the comparison reference value and the counter on this channel 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 (sawtooth wave overflows and triangle wave counts to the trough), the monitoring port TMR4_ADSM will be reset (the output is low).

It should be noted that when the configuration register TMR4_SCSRm enables both up-counting EVT and down-counting EVT (TMR4_SCSRm.UEN=1 and TMR4_SCSRm.DEN=1), the synchronization signal output to the external port is only valid for up-counting EVT.

Figure 20-26 is an example of the synchronization signal output to the port in the comparison start mode (SCSRm.EVTMS=0). TMR4_ADSM is set after the upward counting EVT valid, and TMR4_ADSM is cleared after the counter underflows.

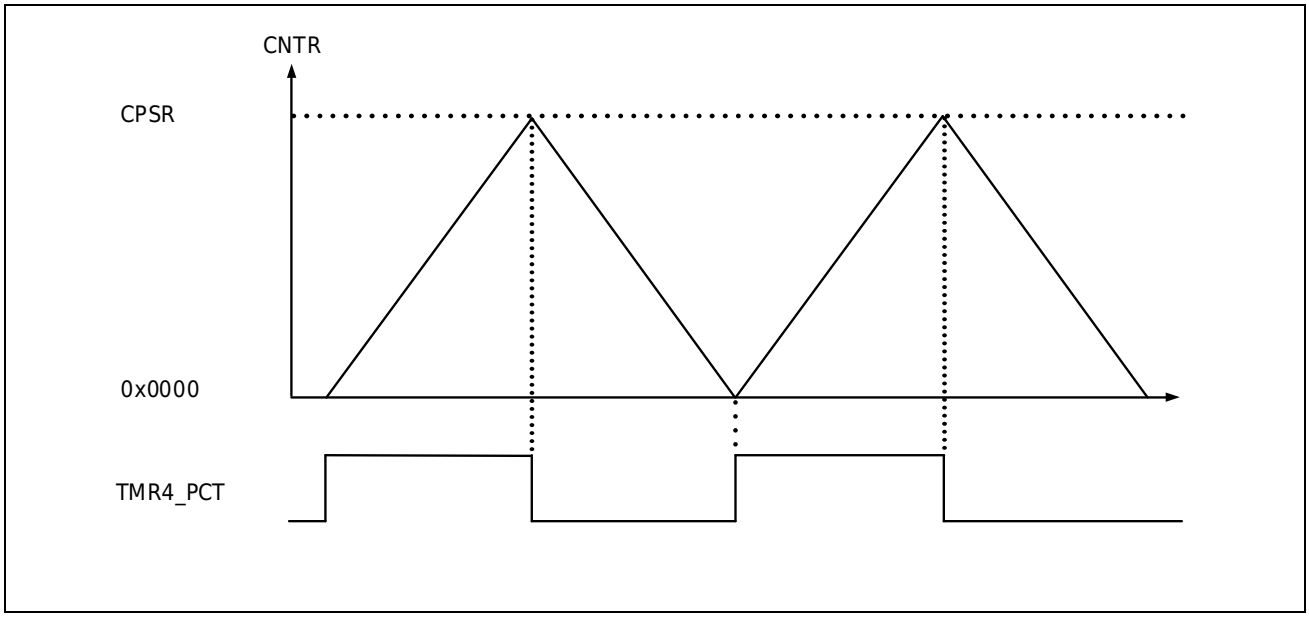


Figure 20-25 PWM Period Output Monitoring Example

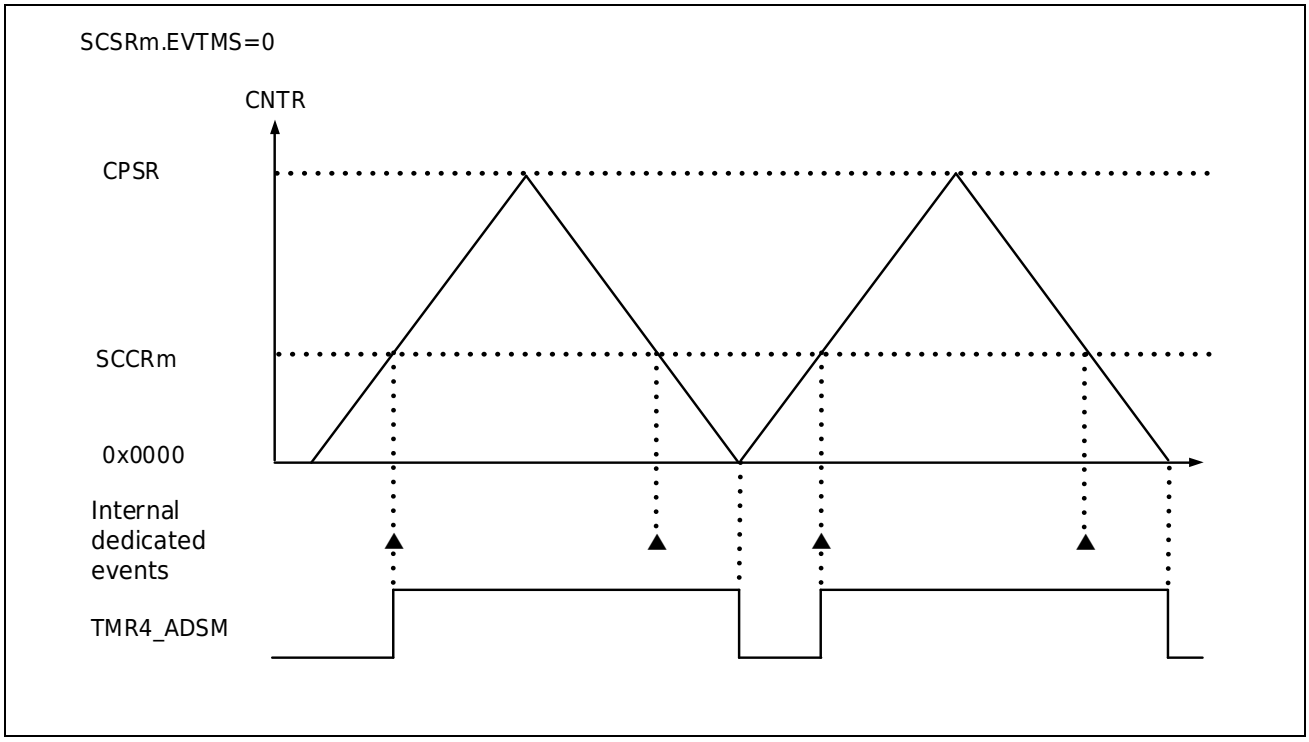


Figure 20-26 Dedicated Event Output TMR4_ADSM Timing in Compare Start Mode

20.3.8 Hardware Startup

Timer4 can start the counter 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 internal trigger event selection register (TMR4_TRGSEL). For details on the register content, see the [Timer4 Hardware Trigger Event Selection Register (TMR4_TRGSEL)] section. For specific event correspondence, please refer to the [Interrupt controller (INTC)] section. When using the internal hardware trigger function, the function clock control register (PWC_FCG0) peripheral circuit trigger function is required to enable position 1.

20.3.9 Synchronous Startup

The product is equipped with a count synchronization enable register (TMR_SYNENR) to synchronously start all timers (including Timer0, Timer4, Timer6, TimerA) carried by the product. for application method and Register Description, please refer to the [Universal Timer (Timer0)] section.

20.4Interrupt and Events

20.4.1 Universal Purpose Count Compare Match Interruptand Event

There are 8 common baseline values register (OCCRm), which can be compared with the count values to produce matching valid signals. When the count comparison matches, the OCSRn. OCFH and OCSRn. OCFL bits in the generic control state register (OCSRn) are set to 1. At this time, if

OCSRn.OCIEH and OCSRn.OCIEL are set to enable interrupts, the corresponding interrupt request (TMR4_GCMmn, m=U, V, W, X; n=H, L) will also be triggered.

The Universal count comparison match event can be used to trigger other modules. For the event number, please refer to the [Interrupt Event List] section.

20.4.2 Counting Period Match Interruptand Events

The CCSR's CCSR IRQPF or CCSR IRQZF position is set to 1 when the zigzag wave is incremented to the upper spillage point, the zigzag wave is decremented to the lower spillage point, the triangular wave is counted to the valley point or the triangular wave is counted to the peak point. At this time, if the CCSR.IRQPEN or CCSR.IRQZEN bit is set to enable interrupt, the count cycle match interrupt (TMR4_GOVF and TMR4_GUDF) can be triggered at the corresponding time point.

The counting cycle match event can be used to trigger other modules. For the event number, please refer to the [Interrupt Event List] section.

20.4.3 Reload Count Match Interruptand Event

When the overload function is effective, the PWM filter control register (PFSRn) compares with the count value to produce the matched signal. When the count compare matches, the RCSR.RTIFU, RCSR.RTIFV, RCSR.RTIFW, and RCSR.RTIFX bits in the reload control status register (RCSR) will be set to 1 respectively. At this time, if the RCSR.RTIDU, RCSR.RTIDV, RCSR.RTIDW, and RCSR.RTIFX interrupt masks are set to invalid, the corresponding reload count match interrupt request (TMR4_RLOm, m=U, V, W, X) will also be triggered.

The reload count match event can be used to trigger other modules. For the event number, please refer to the [Interrupt Event List] section.

20.4.4 Dedicated Compare Match Interrupts and Events

The 8 dedicated compare reference registers (SCCRm) of Timer4 generate 8 dedicated 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, a count comparison match event occurs in the dedicated comparison reference value (SCCRm), generating a corresponding valid request signal. This request signal can be configured to any event EVT output signal (set by the SCSR.EVTOS bit) to trigger other modules. At this time, if SCIR.ITEN0~7 is set to enable interrupt, the corresponding dedicated compare match interrupt request (TMR4_SCM0~7) will also be triggered.

The output of the event request signal can be selected to compare the start mode or delay start mode. After the match event, the dedicated event output signal is output directly; in the delayed start mode (SCSR.EVTMS=1), after the count comparison match event of OCCR*h or OCCR*l (selected by the SCSR.EVTDS bit) is generated, the dedicated event output signal is output after

the reference cycle time set by SCCR. TheFigure 20-27 following shows an example of the request output of the dedicated event output signal in the delayed start mode.

Dedicated comparison match events can be used to trigger other modules. For event numbers, please refer to the [Interrupt Event List] section.

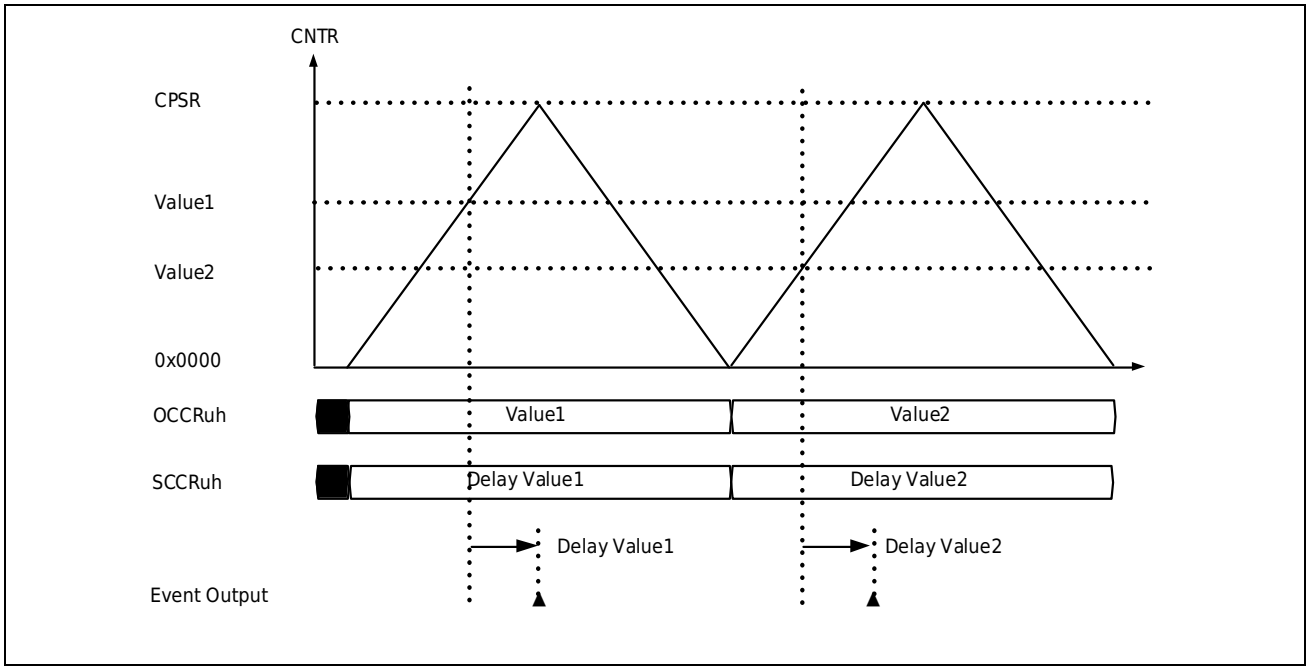


Figure 20-27 Output Timing of Dedicated Event Output Signal in Delayed Start Mode

Note:

– *In a delay count run, if an OCCR and counter match event occurs again, the delay counter reloads the count value and decrements the count again. Therefore, if the OCCR match event time interval is less than the set delay time SCCR, the request signal output by the dedicated event may not always be generated.*

20.5Register Description

Table 20-3 Timer4 Base Address

Name	Base Address	Description
Timer4	0x4003 8000	Timer4_1 base address

Table 20-4 Timer4 Register List

Register	Description	Offset Address	Bit Width	Reset Value
TMR4_CNTR	Register count value	0x0054	16	0x0000
TMR4_CPSR	Cycle reference register	0x0050	16	0xFFFF
TMR4_CCSR	Control state	0x0058	16	0x0040
TMR4_CVPR	Valid period register	0x005A	16	0x0000
TMR4_PSCR	PWM State Control Register	0x005C	32	0x5555 0000
TMR4_OCCRuh	Generic baseline registerUH	0x0000	16	0x0000
TMR4_OCCRul	Generic baseline registerUL	0x0004	16	0x0000
TMR4_OCCRvh	Generic baseline registerVH	0x0008	16	0x0000
TMR4_OCCRvl	Generic baseline registerVL	0x000C	16	0x0000
TMR4_OCCRwh	Generic baseline registerWH	0x0010	16	0x0000
TMR4_OCCRwl	Generic baseline registerWL	0x0014	16	0x0000
TMR4_OCCRxh	Generic baseline register XH	0x0018	16	0x0000
TMR4_OCCRxl	Generic baseline register XL	0x001C	16	0x0000
TMR4_OCSRu	Generic control state register U	0x0020	16	0xFF00
TMR4_OCSRv	Generic control state registerV	0x0024	16	0xFF00
TMR4_OCSRw	Generic control state registerW	0x0028	16	0xFF00
TMR4_OCSRx	Universal Control Status Register X	0x002C	16	0xFF00
TMR4_OCERu	Generic extension control register U	0x0022	16	0x0000
TMR4_OCERv	Generic extension control register V	0x0026	16	0x0000
TMR4_OCERw	Generic Extended Control Register W	0x002A	16	0x0000
TMR4_OCERx	Universal Extended Control Register X	0x002E	16	0x0000
TMR4_OCMRuh	Generic mode control register UH	0x0030	16	0x0000
TMR4_OCMRul	Generic mode control registerUL	0x0034	32	0x0000 0000
TMR4_OCMRvh	Generic mode control registerVH	0x0038	16	0x0000
TMR4_OCMRvl	Generic mode control registerVL	0x003C	32	0x0000 0000
TMR4_OCMRwh	Generic mode control	0x0040	16	0x0000

Register	Description	Offset Address	Bit Width	Reset Value
	registerWH			
TMR4_OCMRwl	Generic mode control registerWL	0x0044	32	0x0000 0000
TMR4_OCMRxh	Generic mode control register XH	0x0048	16	0x0000
TMR4_OCMRxl	Generic mode control register XL	0x004C	32	0x0000 0000
TMR4_SCCRuh	Dedicated baseline register UH	0x00B0	16	0x0000
TMR4_SCCRul	Special baseline registerUL	0x00B4	16	0x0000
TMR4_SCCRvh	Dedicated baseline registerVH	0x00B8	16	0x0000
TMR4_SCCRvl	Dedicated baseline registerVL	0x00BC	16	0x0000
TMR4_SCCRwh	Special baseline registerWH	0x00C0	16	0x0000
TMR4_SCCRwl	Dedicated baseline registerWL	0x00C4	16	0x0000
TMR4_SCCRxh	Dedicated baseline register XH	0x00C8	16	0x0000
TMR4_SCCRxl	Dedicated baseline register XL	0x00CC	16	0x0000
TMR4_SCSRuh	Special control state register UH	0x00D0	16	0x0000
TMR4_SCSRul	Special control status registerUL	0x00D4	16	0x0000
TMR4_SCSRvh	Special control state registerVH	0x00D8	16	0x0000
TMR4_SCSRvl	Special control state registerVL	0x00DC	16	0x0000
TMR4_SCSRwh	Special control state registerWH	0x00E0	16	0x0000
TMR4_SCSRwl	Special control state registerWL	0x00E4	16	0x0000
TMR4_SCSRxh	Special control state register XH	0x00E8	16	0x0000
TMR4_SCSRxl	Special control state register XL	0x00EC	16	0x0000
TMR4_SCMRuh	Special mode control register UH	0x00D2	16	0xFF00
TMR4_SCMRul	Special mode control registerUL	0x00D6	16	0xFF00
TMR4_SCMRvh	Special mode control registerVH	0x00DA	16	0xFF00
TMR4_SCMRvl	Special mode control registerVL	0x00DE	16	0xFF00
TMR4_SCMRwh	Special mode control registerWH	0x00E2	16	0xFF00
TMR4_SCMRwl	Special mode control registerWL	0x00E6	16	0xFF00
TMR4_SCMRxh	Special mode control register XH	0x00EA	16	0xFF00
TMR4_SCMRxl	Special mode control register XL	0x00EE	16	0xFF00
TMR4_SCER	Dedicated Extended Control Register	0x00F0	16	0x0000
TMR4_POCRu	PWM basic control register U	0x00A0	16	0xFF00
TMR4_POCRv	PWM basic control register V	0x00A4	16	0xFF00

Register	Description	Offset Address	Bit Width	Reset Value
TMR4_POCRw	PWM basic control register W	0x00A8	16	0xFF00
TMR4_POCRx	PWM Basic Control Register X	0x00AC	16	0xFF00
TMR4_PFSRu	PWM filtering control register U	0x0082	16	0x0000
TMR4_PFSRv	PWM filtering control register V	0x008A	16	0x0000
TMR4_PFSRw	PWM filtering control register W	0x0092	16	0x0000
TMR4_PFSRx	PWM Filter Control Register X	0x009A	16	0x0000
TMR4_PDARu	PWM dead zone control register AU	0x0084	16	0x0000
TMR4_PDBRu	PWM dead zone control registerBU	0x0086	16	0x0000
TMR4_PDARv	PWM dead zone control registerAV	0x008C	16	0x0000
TMR4_PDBRv	PWM dead zone control registerBV	0x008E	16	0x0000
TMR4_PDARw	PWM dead zone control registerAW	0x0094	16	0x0000
TMR4_PDBRw	PWM dead zone control registerBW	0x0096	16	0x0000
TMR4_PDARx	PWM dead zone control register AX	0x009C	16	0x0000
TMR4_PDBRx	PWM dead zone control register BX	0x009E	16	0x0000
TMR4_RCSR	Overload control status register	0x00F4	32	0x0000 0000
TMR4_SCIR	Dedicated interrupt control register	0x00F8	16	0xFF00
TMR4_SCFR	Dedicated flag control register	0x00FC	16	0xFF00

Note:

– *In the following register detailed descriptions, m=uh, ul, vh, vl, wh, wl, xh, xl, and n=u, v, w, x. The registers pointed to by m correspond to the output controls of ports TMR4_OUH, TMR4_OUL, TMR4_OVH, TMR4_OVL, TMR4_OWH, TMR4_OWL, TMR4_OXH, and TMR4_OXL respectively when the function is implemented; the registers pointed to by n correspond to the output controls of ports TMR4_OUx, TMR4_OVx, TMR4_OWx, and TMR4_OXx respectively when the function is implemented, where x=H or L, and the specific controls of H or L have corresponding symmetrical bits in these registers.*

20.5.1 Count Value Register (TMR4 _ CNTR)

Reset value: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
CNTR[15:0]															

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b15~b0	CNTR[15:0]	Counter current value	When the count stops, the counter count value can be initialized by writing a value to the register In a count, the bit indicates the current counter count value **Note:** You cannot write a value to the register in a count	RW

20.5.2 Periodic Reference Register (TMR4 _ CPSR)

Reset Value: 0xFFFF

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
CPSR[15:0]															

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b15~b0	CPSR[15:0]	Universal cycle reference	Counter count cycle value **Note:** When reading data from the local area, it is not the value of the buffer register, but the value of the CPS rregister	RW

20.5.3 Control State Register (TMR4 _ CCSR)

Reset Value: 0x0040

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
ECK EN	Res	HST	Res	IRQ ZF	IRQ ZEN	IRQ PF	IRQ PEN	BUF EN	STOP	MODE	CLEAR	CKDIV[3:0]			

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b15	ECKEN	Timer selection	0: Internal PCLK0 clock 1: External TMR4_CLK port input clock **Note:** 1. This bit is set when the counter is stopped. 2. When operating with an external input clock, the edge of the original external input clock, whether rising or falling, is ignored after the counter operation is started, and the counting operation begins with the latter edge	RW
b14	Res	Reserved Bit	Read at “0”, write at “0”	RW
b13	HST	Hardware enable	Condition: The event specified in the TMR4_TRGSEL register occurs 0: Hardware startup is invalid when conditions match 1: When the condition is matched, the hardware start is effective and the counter starts counting	
b14	Res	Reserved Bit	Read at “0”, write at “0”	RW
b11	IRQZF	Underflow state	0: No count underflow 1: Occurrence count underflow **Note:** 1. When the periodic interval response function is used, the setting condition of the bit is set by the periodic interval counter set by CVPR 2. When the counter is reset by the bus or CLEAR=1 is written, the IRQZF bit will not be set.	RW
B10	IRQZEN	Underflow interrupt enable	0: Disable IRQZF from producing interrupt to CPU 1: Allow IRQZF to generate interrupt to CPU	RW
b9	IRQPF	Overflow state	0: No count overflow 1: Occurrence count overflow **Note:** 1. When the periodic interval response function is used, the setting condition of the bit is set by the periodic interval counter set by CVPR 2. When the counter is reset by the bus or CLEAR=1 is written, the IRQZF bit will not be set.	RW
b8	IRQPEN	Overflow interrupt enable	0: Disable IRQPF from producing interrupt to CPU 1: Allow IRQPF to generate interrupt to CPU	RW
b7	BUFEN	Cache enable	0: Disable CPSR caching 1: Enable CPSR caching	RW
b6	STOP	Counter enable	0: Counter Startup 1: Counter stopped	RW
b5	MODE	Waveform mode	0: Sawtooth mode (counting up only) 1: Triangular wave mode	RW
b4	CLEAR	Counter reset	0: No operation 1: Counter zeroing **Note:** This bit is always read as 0	RW
b3~b0	CKDIV	Counting clock division frequency	This bit indicates the counting clock division of the basic counter 0000: The counting clock is PCLK0 0001: Counting clock is PCLK0/2 0010: Counting clock is PCLK0/4 0011: Counting clock is PCLK0/8 0100: Counting clock is PCLK0/16 0101: Counting clock is PCLK0/32 0110: Counting clock is PCLK0/64 0111: Counting clock is PCLK0/128 1000: Counting clock is PCLK0/256 1001: Counting clock is PCLK0/512 1010: Counting clock is PCLK0/1024 Other: prohibitions **Note:** When the clock source is internal PCLK0, the frequency division is effective	RW

20.5.4 Valid Period Register (TMR4 _ CVPR)

Reset value: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
PIC[3:0]				ZIC[3:0]				PIM[3:0]				ZIM[3:0]			

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b15~b12	PIC[3:0]	Overflow interrupt shielding state	PIC is a fixed down counter. When an overflow interrupt occurs, the value of PIC decreases. 1. Reading PIC can obtain the number of overflow interrupts that currently need to be masked. Write condition: counter stopped state (CCSR.STOP=1) The value of PIM is loaded into the PIC register if one of the following is met: 1. Counter clearing occurs (CCSR.CLEAR writes 1) 2. PIC=0 and CNTR counter overflow interrupt occurs	R
b11~b8	ZIC[3:0]	Underflow interrupt shielding state	ZIC is a fixed down-counting counter. When an underflow interrupt occurs, the value of ZIC is Subtract 1 and read ZIC to 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 the ZIC register if one of the following conditions is met: 1. Counter clearing occurs (CCSR.CLEAR writes 1) 2. ZIC=0 and CNTR counter underflow interrupt occurs	R
b7~b4	PIM[3:0]	Overflow interrupt shielding setting	Set the number of overflow interrupts that need to be masked The value of PIM is loaded into the PIC register if one of the following is met: 1. Counter clearing occurs (CCSR.CLEAR writes 1) 2. PIC=0 and CNTR counter overflow interrupt occurs	RW
b3~b0	ZIM[3:0]	Underflow interrupt mask setting	Set the number of underflow interrupts that need to be masked The value of ZIM is loaded into the ZIC register if one of the following conditions is met: 1. Counter clearing occurs (CCSR.CLEAR writes 1) 2. ZIC=0 and CNTR counter underflow interrupt occurs	RW

20.5.5 Generic Baseline Register (TMR4 _ OCCRm)

Reset value: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
OCCR[15:0]															

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b15~b0	OCCR[15:0]	Common baseline value	Common baseline value **Note:** When reading data from the local area, it is not the value of the buffer register but the value of the OCCR register	RW

20.5.6 Universal Control State Register (TMR4 _ OCSRn)

Reset Value: 0xFF00

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res								OCFL	OCFH	OCIEL	OCIEH	OCPL	OCPH	OCEL	OCEH

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b15~b8	Res	Reserved Bit	All bits are "1" when reading, and all bits are written "1" when writing	RW
b7	OCFL	Count match flag OCFL	0: Counter count value is not equal to OCCR*l set value 1: Counter count is equal to OCCR*l set **Note:** Note: The bit must be valid when OCEL=1	RW
b6	OCFH	Count match flag OCFH	0: Counter count value is not equal to OCCR*h set value 1: Counter count is equal to OCCR*h set **Note:** Note: The bit must be valid when OCEH=1	RW
b5	OCIEL	Count match interrupt enable L	0: OCFL does not interrupt when set 1: Interrupt occurs when OCFL is set	RW
b4	OCIEH	Count match interrupt enable H	0: No interrupt occurs when OCFH is set 1: Interrupt occurs when OCFH is set	RW
b3	OCPL	Port state L when comparison output is invalid	0: When OCEL=0, in_op*l outputs low level 1: When OCEL=0, in_op*l outputs high level	RW
b2	OCPH	Port state H when the comparison output is invalid	0: When OCEH=0, in_op*h outputs low level 1: When OCEH=0, in_op*h outputs high level	RW
b1	OCEL	Port Output Selection L	0: Comparison output invalid, in_op*l status is determined by OCPL 1: Comparison output valid, the in_op*l state is determined by the OCMR*l setting and OCFL state	RW
b0	OCEH	Port Output Selection H	0: Comparison output invalid, in_op*h state is determined by OCPH 1: Comparison output valid, the in_op*h state is determined by the setting of OCMR*h and the state of OCFH	RW

20.5.7 Generic Extended Control Register (TMR4 _ OCERn)

Reset value: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res		MCECL	MCECH	LMML	LMMH	LMCL	LMCH	MLBUF EN[1:0]		MHBUF EN[1:0]		CLBUF EN[1:0]		CHBUF EN[1:0]	

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b15~b14	Res	Reserved Bit	Read at “0”, write at “0”	RW
b13	MCECL	Count matching increment condition enable L	Add conditions: count overflow (CNTR=CPSR) and OCCR*l > CPSR as count match conditions 0: Disable this condition 1: Enable this condition. When the condition is met, OCCR*l count matches **Note:** 1. This bit can only be modified if the output comparison function is invalid (OCSR.OCEL = 0)	RW
b12	MCECH	Count matching increment condition enable H	Add conditions: count overflow (CNTR=CPSR) and OCCR*h > CPSR as count match conditions 0: Disable this condition 1: Enable this condition. When the condition is met, the OCCR*h count matches **Note:** 1. This bit can only be modified if the output comparison function is invalid (OCSR.OCEH = 0)	RW
b11	LMML	OCMR*l Cycle Interval Response Function Link L	0: Interval response function link is invalid, OCMR*l cache transfer is determined by MLBUFEN setting 1: The periodic interval response function link is valid, and the buffer transmission of OCMR*l must also satisfy CVPR.PIC[3:0]=0000 (when the count overflows) or CVPR.ZIC[3:0]= on the basis of the MLBUFEN setting. 0000 (count underflow)	RW
b10	LMMH	OCMR*h Cycle Interval Response Function Lin H	0: The periodic interval response function link invalid, and the buffer transfer of OCMR*h is determined by the MHBUFEN setting 1: The periodic interval response function link is valid. The buffer transfer of OCMR*h must satisfy CVPR.PIC[3:0]=0000 (when the count overflows) or CVPR.ZIC[3:0]=0000 (when the count underflows) based on the MHBUFEN setting.	RW
b9	LMCL	OCCR*l Cycle interval response function link L	0: The periodic interval response function link invalid, and the buffer transfer of OCCR*l is determined by the CLBUFEN setting 1: The periodic interval response function link is valid. The buffer transfer of OCCR*l must satisfy CVPR.PIC[3:0]=0000 (when the count overflows) or CVPR.ZIC[3:0]=0000 (when the count underflows) based on the CLBUFEN setting.	RW
b8	LMCH	OCCR*h cycle interval response function link H	0: The periodic interval response function link invalid, and the buffer transfer of OCCR*h is determined by the CHBUFEN setting 1: The periodic interval response function link is valid. The buffer transfer of OCCR*h must satisfy CVPR.PIC[3:0]=0000 (when the count overflows) or CVPR.ZIC[3:0]=0000 (when the count underflows) based on the CHBUFEN setting.	RW
b7~b6	MLBUFEN[1:0]	OCMRxl Cache Transport	00: The value of the OCMR*l cache register is directly written into OCMR*l 01: The value of the OCMR*l cache register is written to OCMR*l when the count underflows 10: The value of the OCMR*l cache register is written to OCMR*l when the count overflows. 11: The value of the OCMR*l cache register is written to OCMR*l when the count underflows or overflows.	RW
b5~b4	MHBUFEN[1:0]	OCMRxh Cache Transfer	00: The value of the OCMR*h cache register is directly written into OCMR*h 01: The value of the OCMR*h buffer register is written to OCMR*h when the count underflows 10: The value of the OCMR*h cache register is written to OCMR*h when the count overflows. 11: The value of the OCMR*h cache register is written to OCMR*h when the count underflows or overflows.	RW
b3~b2	CLBUFEN[1:0]	OCCRxl Cache Transport	00: The value of the OCCR*l cache register is directly written into OCCR*l 01: The value of the OCCR*l cache register is written to OCCR*l when the count underflows 10: The value of the OCCR*l cache register is written to	RW

			OCCR*l when the count overflows. 11: The value of the OCCR*l cache register is written to OCCR*l when the count underflows or overflows.	
b1~b0	CHBUFEN[1:0]	OCCRxh Cache Transport	00: The value of the OCCR*h cache register is directly written into OCCR*h 01: The value of the OCCR*h buffer register is written to OCCR*h when the count underflows 10: The value of the OCCR*h cache register is written to OCCR*h when the count overflows. 11: The value of the OCCR*h cache register is written to OCCR*h when the count underflows or overflows.	RW

20.5.8 Universal Mode Control Register (TMR4 _ OCMRm)

Reset value: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
OPN ZRH[1:0]		OPN PKH[1:0]		OP ZRH[1:0]		OP UCH[1:0]		OP PKH[1:0]		OP DCH[1:0]		OCF ZRH	OCF UCH	OCF PKH	OCF DCH

Table 20-5 OCCR*h Count Match List

OCCR*h count match	Condition Number
0	1
1	2

Note: 0: UnMatch, 1: Match

Note: This register bit description is used for OCMRuh, OCMRvh, OCMRwh, OCMRxh

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b15~b14	OPNZRH[1:0]	Underflow condition 1 port status H	Underflow condition 1 (satisfied at the same time): count underflow, OCCR*h count does not match 00: When the condition is met, in_op*h remains unchanged 01: When the conditions are met, in_op*h outputs a high level 10: When the conditions are met, in_op*h outputs a low level 11: When the condition is met, the in_op*h output is reversed	RW
b13~b12	OPNPKH[1:0]	Overflow condition 1 port status H	Overflow condition 1 (satisfied at the same time): count overflow, OCCR*h count does not match 00: When the condition is met, in_op*h remains unchanged 01: When the conditions are met, in_op*h outputs a high level 10: When the conditions are met, in_op*h outputs a low level 11: When the condition is met, the in_op*h output is reversed	RW
b11~b10	OPZRH[1:0]	Underflow condition 2 port status H	Underflow condition 2 (satisfied at the same time): count underflow, OCCR*h count match 00: When the condition is met, in_op*h remains unchanged 01: When the conditions are met, in_op*h outputs a high level 10: When the conditions are met, in_op*h outputs a low level 11: When the condition is met, the in_op*h output is reversed	RW
b9~b8	OPUCH[1:0]	Up counting condition 2 port status H	Up counting condition 2 (satisfied at the same time): the counter counts up, and the OCCR*h count matches 00: When the condition is met, in_op*h remains unchanged 01: When the conditions are met, in_op*h outputs a high level 10: When the conditions are met, in_op*h outputs a low level 11: When the condition is met, the in_op*h output is reversed	RW
b7~b6	OPPKH[1:0]	Overflow condition 2 port status H	Overflow condition 2 (satisfied at the same time): count overflow, OCCR*h count match 00: When the condition is met, in_op*h remains unchanged 01: When the conditions are met, in_op*h outputs a high level 10: When the conditions are met, in_op*h outputs a low level 11: When the condition is met, the in_op*h output is reversed	RW
b5~b4	OPDCH[1:0]	Down counting condition 2 port status H	Down counting condition 2 (satisfied at the same time): the counter counts down, and the OCCR*h count matches 00: When the condition is met, in_op*h remains unchanged 01: When the conditions are met, in_op*h outputs a high level 10: When the conditions are met, in_op*h outputs a low level 11: When the condition is met, the in_op*h output is reversed	RW
b3	OCFZRH	Underflow point OCFH state H	Conditions (satisfied simultaneously): count underflow, OCCR*h count match 0: OCSR.OCFH bit remains unchanged when the condition is satisfied 1: OCSR.OCFH position bit when conditions are met	RW
b2	OCFUCH	Up count OCFH state H	Conditions (satisfied simultaneously): counter counts up, OCCR*h count matches 0: OCSR.OCFH bit remains unchanged when the condition is satisfied 1: OCSR.OCFH position bit when conditions are met	RW
b1	OCFPKH	Overflow point OCFH state H	Conditions (satisfied simultaneously): count overflow, OCCR*h count match 0: OCSR.OCFH bit remains unchanged when the condition is	RW

			satisfied 1: OCSR.OCFH position bit when conditions are met	
b0	OCFDCH	Down count OCFH state H	Conditions (satisfied simultaneously): counter counts down, OCCR*h count matches 0: OCSR.OCFH bit remains unchanged when the condition is satisfied 1: OCSR.OCFH position bit when conditions are met	RW

Reset value: 0x0000 0000

b31 b30	b29 b28	b27 b26	b25 b24	b23 b22	b21 b20	b19 b18	b17 b16
EOPN ZRL[1:0]	EOPN PKL[1:0]	EOP ZRL[1:0]	EOP UCL[1:0]	EOP PKL[1:0]	EOP DCL[1:0]	EOPN UCL[1:0]	EOPN DCL[1:0]

b15 b14	b13 b12	b11 b10	b9 b8	b7 b6	b5 b4	b3	b2	b1	b0
OPN ZRL[1:0]	OPN PKL[1:0]	OP ZRL[1:0]	OP UCL[1:0]	OP PKL[1:0]	OP DCL[1:0]	OCF ZRL	OCF UCL	OCF PKL	OCF DCL

Table 20-6 OCCR*h / Count Match List

OCCR*h Count Match	OCCR*l Count Match	Condition Number
0	0	1
0	1	2
1	0	3
1	1	4

Note: 0: UnMatch, 1: Match

Note: This register bit description is used for OCMRul, OCMRvl, OCMRwl, OCMRxl

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b30	EOPNZRL[1:0]	Underflow condition 3 port status L	Underflow point condition 3 (satisfied at the same time): count underflow, OCCR*l count does not match, OCCR*h count matches 00: When the condition is met, in_op*l remains unchanged 01: When the conditions are met, in_op*l outputs a high level 10: When the conditions are met, in_op*l outputs a low level 11: When the condition is met, the output of in_op*l is reversed	RW
b29~b28	EOPNPKL[1:0]	Overflow condition 3 port status L	Overflow condition 3 (satisfied at the same time): count overflow, OCCR*l count does not match, OCCR*h count matches 00: When the condition is met, in_op*l remains unchanged 01: When the conditions are met, in_op*l outputs a high level 10: When the conditions are met, in_op*l outputs a low level 11: When the condition is met, the output of in_op*l is reversed	RW
b27~b26	EOPZRL[1:0]	Underflow condition 4 port status L	Underflow condition 4 (satisfied at the same time): count underflow, OCCR*l count match, OCCR*h count match 00: When the condition is met, in_op*l remains unchanged 01: When the conditions are met, in_op*l outputs a high level 10: When the conditions are met, in_op*l outputs a low level 11: When the condition is met, the output of in_op*l is reversed	RW
b25~b24	EOPUCL[1:0]	Up counting condition 4 port status L	Up counting condition 4 (satisfied at the same time): counter counts up, OCCR*l count matches, OCCR*h count matches 00: When the condition is met, in_op*l remains unchanged 01: When the conditions are met, in_op*l outputs a high level 10: When the conditions are met, in_op*l outputs a low level 11: When the condition is met, the output of in_op*l is reversed	RW
b23~b22	EOPPKL[1:0]	Overflow condition 4 port status L	Overflow condition 4 (satisfied at the same time): count overflow, OCCR*l count match, OCCR*h count match 00: When the condition is met, in_op*l remains unchanged 01: When the conditions are met, in_op*l outputs a high level 10: When the conditions are met, in_op*l outputs a low level 11: When the condition is met, the output of in_op*l is reversed	RW
b21~b20	EOPDCL[1:0]	Down count condition 4 port status L	Down counting condition 4 (satisfied at the same time): counter counts down, OCCR*l count matches, OCCR*h count matches 00: When the condition is met, in_op*l remains unchanged 01: When the conditions are met, in_op*l outputs a high level 10: When the conditions are met, in_op*l outputs a low level 11: When the condition is met, the output of in_op*l is reversed	RW

b19~b18	EOPNUCL[1:0]	Up counting condition 3 port status L	Up counting condition 3 (satisfied at the same time): the counter counts up, the OCCR*l count does not match, and the OCCR*h count matches 00: When the condition is met, in_op*l remains unchanged 01: When the conditions are met, in_op*l outputs a high level 10: When the conditions are met, in_op*l outputs a low level 11: When the condition is met, the output of in_op*l is reversed	RW
b17~b16	EOPNDCL[1:0]	Down count condition 3 port status L	Down counting condition 3 (satisfied at the same time): the counter counts down, the OCCR*l count does not match, and the OCCR*h count matches 00: When the condition is met, in_op*l remains unchanged 01: When the conditions are met, in_op*l outputs a high level 10: When the conditions are met, in_op*l outputs a low level 11: When the condition is met, the output of in_op*l is reversed	RW
b15~b14	OPNZRL[1:0]	Underflow condition 1 port status L	Underflow condition 1 (satisfied at the same time): count underflow, OCCR*l count does not match, OCCR*h count does not match 00: When the condition is met, in_op*l remains unchanged 01: When the conditions are met, in_op*l outputs a high level 10: When the conditions are met, in_op*l outputs a low level 11: When the condition is met, the output of in_op*l is reversed	RW
b13~b12	OPNPKL[1:0]	Overflow condition 1 port status L	Overflow condition 1 (satisfied at the same time): count overflow, OCCR*l count mismatch, OCCR*h count mismatch 00: When the condition is met, in_op*l remains unchanged 01: When the conditions are met, in_op*l outputs a high level 10: When the conditions are met, in_op*l outputs a low level 11: When the condition is met, the output of in_op*l is reversed	RW
b11~b10	OPZRL[1:0]	Underflow condition 2 port status L	Underflow condition 2 (satisfied at the same time): count underflow, OCCR*l count matches, OCCR*h count does not match 00: When the condition is met, in_op*l remains unchanged 01: When the conditions are met, in_op*l outputs a high level 10: When the conditions are met, in_op*l outputs a low level 11: When the condition is met, the output of in_op*l is reversed	RW
b9~b8	OPUCL[1:0]	Up counting condition 2 port status L	Up counting condition 2 (satisfied at the same time): the counter counts up, the OCCR*l count matches, and the OCCR*h count does not match 00: When the condition is met, in_op*l remains unchanged 01: When the conditions are met, in_op*l outputs a high level 10: When the conditions are met, in_op*l outputs a low level 11: When the condition is met, the output of in_op*l is reversed	RW
b7~b6	OPPKL[1:0]	Overflow condition 2 port status L	Overflow condition 2 (satisfied at the same time): count overflow, OCCR*l count matches, OCCR*h count does not match 00: When the condition is met, in_op*l remains unchanged 01: When the conditions are met, in_op*l outputs a high level 10: When the conditions are met, in_op*l outputs a low level 11: When the condition is met, the output of in_op*l is reversed	RW
b5~b4	OPDCL[1:0]	Down count port status L	Conditions (satisfied simultaneously): counter counts down, OCCR*l count matches, OCCR*h count does not match 00: When the condition is met, in_op*l remains unchanged 01: When the conditions are met, in_op*l outputs a high level 10: When the conditions are met, in_op*l outputs a low level 11: When the condition is met, the output of in_op*l is reversed	RW
b3	OCFZRL	Underflow point OCFL state L	Conditions (satisfied simultaneously): count underflow, OCCR*l count match 0: OCSR.OCFL bit remains unchanged when the condition is satisfied 1: OCSR.OCFL position bits when conditions are met	RW
b2	OCFUCL	Count up OCFL status L	Conditions (satisfied simultaneously): counter counts up, OCCR*l count matches 0: OCSR.OCFL bit remains unchanged when the condition is satisfied 1: OCSR.OCFL position bits when conditions are met	RW
b1	OCFPKL	Overflow point OCFL state L	Conditions (satisfied simultaneously): count overflow, OCCR*l count match 0: OCSR.OCFL bit remains unchanged when the condition is satisfied 1: OCSR.OCFL position bits when conditions are met	RW
b0	OCFDCL	Count down OCFL status L	Conditions (satisfied simultaneously): counter counts down, OCCR*l count matches 0: OCSR.OCFL bit remains unchanged when the condition is satisfied 1: OCSR.OCFL position bits when conditions are met	RW

Note:

- *When reading data from the local area, it is not the value of the buffer register, but the value of the OCMRregister.*
- *TMR4_O*L can be determined by OCCR*l and the count value of the counter (independent operation mode), or by OCCR*h and the count value of the counter and OCCR*l 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*l, and write "0b0000" to OCMR*l[19:16]. At this time, the output of in_op*l will not be affected by OCCR*h, but only determined by OCCR*l. Independent operation mode: Channel *H is determined by OCCR*h, and channel *L is configured by OCCR*l. If the independent operation mode does not meet the above conditions, it is the linked operation mode: the channel*L output is affected by both OCCR*h and OCCR*l.*

- *When the following two conditions are met, it is also considered to meet the conditions set by OCMR*h. OPPKH[7:6]:*

  *1） CCSR.MODE=1 && count overflow && OCCR*h=0xFFFF;*

  2） 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*1. OPPKL[7:6]:*

  *1） CCSR.MODE=1 && count overflow && OCCR*l=0xFFFF;*

  2） OCER.MCECL=1 && count overflow && OCCR*l≥CNTR.*

20.5.9 Special Reference Register (TMR4 _ SCCRm)

Reset value: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
SCCR[15:0]															

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b15~b0	SCCR[15:0]	Dedicated baseline value	Dedicated baseline (compare startup mode) or delay baseline (delay startup mode) **Note:** When reading data from the local area, it is not the value of the buffer register but the value of the SCCR register	RW

20.5.10 Special Control State Register (TMR4 _ SCSRm)

Reset value: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
ZEN	UEN	PEN	DEN	Res		EVTDS	EVTMS	Res		LMC	EVTOS[2:0]			BUFEN[1:0]	

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b15	ZEN	Underflow EVT enable	0: EVT does not operate when counting underflow 1: When counting underflow: EVTMS = 0 & SCCR Compare Match & SCMR Set Match, EVT Startup Output EVTMS = 1 & OCCR Compare Match & SCMR Set Match, EVT Delay Mode Startup	RW
b14	UEN	Up count EVT enable	0: EVT does not operate when counting up 1: When counting up: EVTMS = 0 & SCCR Compare Match & SCMR Set Match, EVT Startup Output EVTMS = 1 & OCCR Compare Match & SCMR Set Match, EVT Delay Mode Startup	RW
b13	PEN	Overflow point EVT enable	0: EVT does not operate when counting overflow 1: When counting overflow: EVTMS = 0 & SCCR Compare Match & SCMR Set Match, EVT Startup Output EVTMS = 1 & OCCR Compare Match & SCMR Set Match, EVT Delay Mode Startup	RW
b12	DEN	Down count EVT enable	0: EVT does not operate when counting down 1: Down count: EVTMS = 0 & SCCR Compare Match & SCMR Set Match, EVT Startup Output EVTMS = 1 & OCCR Compare Match & SCMR Set Match, EVT Delay Mode Startup	RW
b11~b10	Res	Reserved Bit	Read at “0”, write at “0”	RW
b9	EVTDS	EVT Delay Object Selection	0: OCCR*h matches as a delay comparison object in delay start mode 1: OCCR*l matches as a delay comparison object in delay start mode **Note:** This bit invalid when EVTMS=0.	RW
b8	EVTMS	EVT mode selection	0: Compare startup mode (CNTR and SCCR comparison triggers) 1: Delay start mode (compare matching events triggered after SCCR delay)	RW
b7~b6	Res	Reserved Bit	Read at “0”, write at “0”	RW
b5	LMC	Periodic interval response link	0: Interval response function link invalid, SCCR cache transfer is determined by BUFEN setting 1: The periodic interval response function link is valid, and the cache transmission of SCCR is based on the setting of BUFEN, and must also satisfy CVPR.PIC[3:0]=0000 (when the count overflows) or CVPR.ZIC[3:0]= 0000 (count underflow)	RW
b4~b2	EVTOS[2:0]	EVT Output Selection	000: EVT output of Special Event 0 (TMR4_SCM0) valid 001: EVT output of Special Event 1 (TMR4_SCM1) valid 010: EVT output of Special Event 2 (TMR4_SCM2) valid 011: EVT output of Special Event 3 (TMR4_SCM3) valid 100: EVT output of Special Event 4 (TMR4_SCM4) valid 101: EVT output of Special Event 5 (TMR4_SCM5) valid 110: EVT output of Special Event 6 (TMR4_SCM6) valid 111: EVT output of Special Event 7 (TMR4_SCM7) valid	RW
b1~b0	BUFEN[1:0]	SCCR & SCMR Cache Transport	00: SCCR, SCMR Cache register values are written directly to SCCR, SCMR 01: SCCR, SCMR Cache register values are written to SCCR, SCMR when counting underflow 10: SCCR, SCMR Cache register values are written to SCCR, SCMR when count overflows 11: SCCR, SCMR Cache register values are written to SCCR, SCMR when counting underflow or overflow	RW

20.5.11 Dedicated Extended Control Register (TMR4_SCER)

Reset value: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res							PCTS	Res				EVTRS[3:0]			

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b15~b9	Res	Reserved Bit	Read at “0”, write at “0”	RW
b8	PCTS	Counting Direction Output enable	When the counter works in triangle wave mode, the counting direction is controlled and output to the port 0: Disable the counting direction signal output to the TMR4_PCT port 1: Allow the counting direction signal to be output to the TMR4_PCT port	RW
b7~b4	Res	Reserved Bit	Read at “0”, write at “0”	RW
b3~b0	EVTRS[3:0]	Dedicated event output selection	0000: Disable dedicated event output to the port 0001: EVT of Special Event 0 (TMR4_SCM0) is output to TMR4_ADSM port 0010: EVT of Special Event 1 (TMR4_SCM1) is output to TMR4_ADSM port 0011: EVT of Special Event 2 (TMR4_SCM2) is output to TMR4_ADSM port 0100: EVT of Special Event 3 (TMR4_SCM3) is output to TMR4_ADSM port 0101: EVT of Special Event 4 (TMR4_SCM4) is output to TMR4_ADSM port 0110: EVT of Special Event 5 (TMR4_SCM5) is output to TMR4_ADSM port 1000: EVT of Special Event 6 (TMR4_SCM6) is output to TMR4_ADSM port 1001: EVT of Special Event 7 (TMR4_SCM7) is output to TMR4_ADSM port Other values: Disabled	RW

20.5.12 Special Mode Control Register (TMR4 _ SCMRm)

Reset Value: 0xFF00

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res								MPCE	MZCE	Res		AMC[3:0]			

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b15~b8	Res	Reserved Bit	All bits are "1" when reading, and all bits are written "1" when writing	RW
b7	MPCE	Periodic interval response enable	0: No comparison between AMC and CVPR.PIC 1: Enable comparison between AMC and CVPR.PIC	RW
b6	MZCE	Periodic interval response enable	0: No comparison between AMC and CVPR.ZIC 1: Enable comparison between AMC and CVPR.ZIC	RW
b5~b4	Res	Reserved Bit	Read at “0”, write at “0”	RW
b3~b0	AMC[3:0]	Special event output interval value	This bit sets the interval value of the special event output function. When AMC and CVPR.PIC or CVPR.ZIC are equal, the special event output function is valid.	RW

Note:

– *When reading data from the local area, it is not the value of the buffer register, but the value of the SCMRregister.*

20.5.13 PWM Basic Control Register (TMR4 _ POCRn)

Reset Value: 0xFF00

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res								LVLS[1:0]		PWMMD[1:0]		-	DIVCK[2:0]		

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b15~b8	Res	Reserved Bit	All bits are "1" when reading, and all bits are written "1" when writing	RW
b7~b6	LVLS[1:0]	PWM output polarity control	00: Both TMR4_O*H and TMR4_O*L outputs are not inverted 01: The outputs of TMR4_O*H and TMR4_O*L are both inverted 10: The output of TMR4_O*H is inverted, and the output of TMR4_O*L is not inverted. 11: The output of TMR4_O*H is not inverted, and the output of TMR4_O*L is inverted	RW
b5~b4	PWMMD[1:0]	PWM output mode	00: Pass-through mode 01: Dead-zone timer mode 10: Dead-zone timer filtering mode 11: Setting prohibited	RW
b3	Res	Reserved Bit	Read at “0”, write at “0”	RW
b2~b0	DIVCK[2:0]	Counting clock division frequency	This bit indicates the frequency division of the counting clock between the filter counter and the dead-zone counter. 000: The counting clock is PCLK0 001: Counting clock is PCLK0/2 010: Counting clock is PCLK0/4 011: Counting clock is PCLK0/8 100: Counting clock is PCLK0/16 101: Counting clock is PCLK0/32 110: Counting clock is PCLK0/64 111: Counting clock is PCLK0/128	RW

20.5.14 PWM Status Control Register (TMR4_PSCR)

Reset value: 0x5555 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
OSXL[1:0]		OSXH[1:0]		OSWL[1:0]		OSWH[1:0]		OSVL[1:0]		OSVH[1:0]		OSUL[1:0]		OSUH[1:0]	
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res				ODT[1:0]		AOE	MOE	OEXL	OEXH	OEWL	OEWH	OEVL	OEVH	OEUL	OEUH

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b30	OSXL[1:0]	TMR4_OXL output disable status	Conditions: 1. EMB event occurs; 2. Software writes MOE=0; 3. MOE=1&OEXL=0 00: When one of the above conditions is met, the TMR4_OXL port outputs normally 01: When one of the above conditions is met, the TMR4_OXL port output is Hi-z 10: When one of the above conditions is met, the TMR4_OXL port output is fixed at a low level 11: When one of the above conditions is met, the TMR4_OXL port output is fixed at a high level **Note:** This bit can only be modified when Timer4 is stopped (CCSR.STOP=1).	RW
b29~b28	OSXH[1:0]	TMR4_OXH output disable status	Conditions: 1. EMB event occurs; 2. Software writes MOE=0; 3. MOE=1&OEXH=0 00: When one of the above conditions is met, the TMR4_OXH port outputs normally 01: When one of the above conditions is met, the TMR4_OXH port output is Hi-z 10: When one of the above conditions is met, the TMR4_OXH port output is fixed at a low level 11: When one of the above conditions is met, the TMR4_OXH port output is fixed at a high level **Note:** This bit can only be modified when Timer4 is stopped (CCSR.STOP=1).	RW
b27~b26	OSWL[1:0]	TMR4_OWL output disable status	Conditions: 1. EMB event occurs; 2. Software writes MOE=0; 3. MOE=1&OEWL=0 00: When one of the above conditions is met, the TMR4_OWL port outputs normally 01: When one of the above conditions is met, the TMR4_OWL port output is Hi-z 10: When one of the above conditions is met, the TMR4_OWL port output is fixed at a low level 11: 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
b25~b24	OSWH[1:0]	TMR4_OWH output disable status	Conditions: 1. EMB event occurs; 2. Software writes MOE=0; 3. MOE=1&OEWH=0 00: When one of the above conditions is met, the TMR4_OWH port outputs normally 01: When one of the above conditions is met, the TMR4_OWH port output is Hi-z 10: When one of the above conditions is met, the TMR4_OWH port output is fixed at a low level 11: 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
b23~b22	OSVL[1:0]	TMR4_OVL output disable status	Conditions: 1. EMB event occurs; 2. Software writes MOE=0; 3. MOE=1&OEVL=0 00: When one of the above conditions is met, the TMR4_OVL port outputs normally 01: When one of the above conditions is met, the TMR4_OVL port output is Hi-z 10: When one of the above conditions is met, the TMR4_OVL port output is fixed at a low level 11: 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

b21~b20	OSVH[1:0]	TMR4_OVH output disable status	Conditions: 1. EMB event occurs; 2. Software writes MOE=0; 3. MOE=1&OEVH=0 00: When one of the above conditions is met, the TMR4_OVH port outputs normally 01: When one of the above conditions is met, the TMR4_OVH port output is Hi-z 10: When one of the above conditions is met, the TMR4_OVH port output is fixed at a low level 11: 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
b19~b18	OSUL[1:0]	TMR4_OUL output disable status	Conditions: 1. EMB event occurs; 2. Software writes MOE=0; 3. MOE=1&OEUL=0 00: When one of the above conditions is met, the TMR4_OUL port outputs normally 01: When one of the above conditions is met, the TMR4_OUL port output is Hi-z 10: When one of the above conditions is met, the TMR4_OUL port output is fixed at a low level 11: 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
b17~b16	OSUH[1:0]	TMR4_OUH output disable status	Conditions: 1. EMB event occurs; 2. Software writes MOE=0; 3. MOE=1 and OEUH=0 00: When one of the above conditions is met, the TMR4_OUH port outputs normally 01: When one of the above conditions is met, the TMR4_OUH port output is Hi-z 10: When one of the above conditions is met, the TMR4_OUH port output is fixed at a low level 11: 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
b15~b12	Res	Reserved Bit	Read at “0”, write at “0”	RW
b11~b10	ODT[1:0]	Port enable bit effective time	Condition: After the value of port enable bit OE*L, OE*H is changed 0X: When the conditions are met, it takes effect immediately 10: When the condition is met, the counter will take effect when it underflows next time 11: When the condition is met, the counter will take effect the next time it overflows.	RW
b9	AOE	Automatic output enable	0: After the EMB event is cleared, the MOE bit can only be set to 1 by software to restore normal PWM output 1: After the EMB event is cleared, the MOE bit is automatically set to 1 by hardware at the next count underflow point, and the PWM output returns to normal **Note:** Setting EMB_STATCLR or clearing EMB_SOE can clear the EMB event.	RW
b8	MOE	Master Output Enable	0: TMR4_O*H/L port outputs the status specified by the OS*H/L bit 1: When OE*H, OE*L bits are set to enable, TMR4_O*H, TMR4_O*L port output is enabled **Note:** This bit is cleared by hardware immediately after an EMB event occurs. After the EMB event is cleared, this bit can be set to 1 by software or automatically set to 1 by hardware according to the setting of the AOE bit.	RW
b7	OEXL	TMR4_OXL output enable	0: TMR4_OXL port output state is set by OSXL bit 1: When MOE=1, TMR4_OWL port outputs normal PWM waveform	RW
b6	OEXH	TMR4_OXH output enable	0: TMR4_OXH port output state is set by OSXH bit 1: When MOE=1, TMR4_OXH port outputs normal PWM waveform	RW
b5	OEWL	TMR4_OWL output enable	0: TMR4_OWL port output state is set by OSWL bit 1: When MOE=1, TMR4_OWL port outputs normal PWM waveform	RW
b4	OEWH	TMR4_OWH output enable	0: TMR4_OWH port output state is set by OSWH bit 1: When MOE=1, TMR4_OWH port outputs normal PWM waveform	RW
b3	OEVL	TMR4_OVL output enable	0: TMR4_OVL port output state is set by OSVL bit 1: When MOE=1, TMR4_OVL port outputs normal PWM waveform	RW
b2	OEVH	TMR4_OVH output enable	0: TMR4_OVH port output state is set by OSVH bit 1: When MOE=1, TMR4_OVH port outputs normal PWM waveform	RW
b1	OEUL	TMR4_OUL output enable	0: TMR4_OUL port output state is set by OSUL bit 1: When MOE=1, TMR4_OUL port outputs normal PWM waveform	RW
b0	OEUH	TMR4_OUH output enable	0: TMR4_OUH port output state is set by OSUH bit 1: 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.

Table 20-7 PWM port output status and register setting value

MOE bit	OE*L/H position	OS*L/H position	TMR4_O*L/H output status
1	0	00	Normal Output (Timer4 driven)
		01	Hi-Z
		10	Low level
		11	High level
	1	XX	Normal Output (Timer4 driven)
0	X	00	Normal Output (Timer4 driven)
		01	Hi-Z
		10	Low level
		11	High level

20.5.15 PWM Filtering Control Register (TMR4 _ PFSRn)

Reset value: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
PFSR[15:0]															

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b15~b0	PFSR[15:0]	Filtering initial value	Filter count initial value **Note:** When the dead-zone timer filter mode is not selected for the PWM waveform output mode, the 16-bit filter counter is used as the 16-bit heavy-duty counter. At this time, the 16-bit filter counter can periodically generate interrupt output, which is independent of the PWM waveform generator function.	RW

20.5.16 PWM Dead Zone Control Register (TMR4 _ PDA < B > Rn)

Reset value: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
PDA/BR[15:0]															

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b15~b0	PDA/BR[15:0]	Dead zone initial value	Dead zone count initial value	RW

20.5.17 Overload Control Status Register (TMR4 _ RCSR)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res												RTSX	RTEX	RTICX	RTIFX
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
RTSW	RTEW	RTICW	RTIFW	RTSV	RTEV	RTICV	RTIFV	RTSU	RTEU	RTICU	RTIFU	RTIDX	RTIDW	RTIDV	RTIDU

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b20	Res	Reserved Bit	Read at “0”, write at “0”	RW
b19	RTSX	Reload Counter Stop X	0: No operation 1: Stop heavy-duty counterW and clear RTIFX **Note:** The bit is always 0 when read	RW
b18	RTEX	Reload counter start X	0: Writing 0 invalid, reading 0 means the reload counter X has stopped 1: Write 1 to start reload counter X, read 1 means reload counter X has been started	RW
b17	RTICX	Clear count match status X	0: No operation 1: Clear RTIFX flag bit **Note:** The bit is always 0 when read	RW
b16	RTIFX	Count matching status X	0: Overloaded counter count does not match PFSRx 1: Overloaded counter counts match PFSRx	R
b15	RTSW	Heavy-duty counter stops	0: No operation 1: Stop heavy-duty counterW and clear RTIFW **Note:** The bit is always 0 when read	RW
b14	RTEW	Heavy-duty counter starting W	0: Writing 0 invalid, reading 0 means the reload counter W has stopped 1: Write 1 to start the reload counter W. Reading 1 indicates that the reload counter W has been started.	RW
b13	RTICW	Zero count matching state W	0: No operation 1: Clear RTIFW flag bit **Note:** The bit is always 0 when read	RW
b12	RTIFW	Count Matching Status W	0: Overloaded counter count does not match PFSRw 1: Overloaded counter counts match PFSRw	R
b11	RTSV	Heavy-duty counter stop V	0: No operation 1: Stop heavy-duty counterV and clear RTIFV **Note:** The bit is always 0 when read	RW
b10	RTEV	Heavy-duty counter starting V	0: Writing 0 invalid, and reading 0 means that the reload counter V has stopped. 1: Write 1 to start the reload counter V, and read 1 to indicate that the reload counter V has been started.	RW
b9	RTICV	Zero count matching state V	0: No operation 1: Clear RTIFV flag bit **Note:** The bit is always 0 when read	RW
b8	RTIFV	Count matching state V	0: Overloaded counter count does not match PFSRv 1: Overloaded counter counts match PFSRv	R
b7	RTSU	Heavy-duty counter stop U	0: No operation 1: Stop heavy-duty counterU and clear RTIFU **Note:** The bit is always 0 when read	RW
b6	RTEU	Heavy-duty counter startup U	0: Writing 0 invalid, and reading 0 means that the reload counter U has stopped. 1: Write 1 to start the reload counter U. Reading 1 indicates that the reload counter U has been started.	RW
b5	RTICU	Zero count matching state U	0: No operation 1: Clear RTIFU flag bit **Note:** The bit is always 0 when read	RW
b4	RTIFU	Count matching state U	0: Overloaded counter count does not match PFSRu 1: Overloaded counter counts match PFSRu	R
b3	RTIDX	Reload interrupt mask X	0: When the reload function valid, the reload interrupt X output valid 1: When the reload function valid, the reload interrupt X output invalid	RW

b2	RTIDW	Heavy-duty interrupt shielding W	0: When the overload function is valid, the overload interruptW output is valid. 1: When the overload function is active, the overload interruptW output is invalid	RW
b1	RTIDV	Heavy-duty interrupt shielding V	0: When the overload function is valid, the overload terminal V output is valid. 1: When the overload function is active, the overload interruptV output is invalid	RW
b0	RTIDU	Heavy-duty interrupt shielding U	0: When the overload function is active, the overload interruptU output is active 1: When the overload function is active, the overload interruptU output is invalid	RW

20.5.18 Dedicated Interrupt Control Register (TMR4_SCIR)

Reset Value: 0xFF00

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res								ITEN7	ITEN6	ITEN5	ITEN4	ITEN3	ITEN2	ITEN1	ITEN0

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b15~b8	Res	Reserved Bit	All bits are "1" when reading, and all bits are written "1" when writing	RW
b7	ITEN7	Dedicated match interrupt 7 enable	0: Dedicated count compare match event 7 disable 1: Dedicated count compare match event 7 enable	RW
b6	ITEN6	Dedicated match interrupt 6 enable	0: Dedicated count compare match event 6 disable 1: Dedicated count compare match event 6 enable	RW
b5	ITEN5	Dedicated match interrupt 5 enable	0: Dedicated count compare match event 5 disable 1: Dedicated count compare match event 5 enable	RW
b4	ITEN4	Dedicated match interrupt 4 enable	0: Dedicated count compare match event 4 disable 1: Dedicated count compare match event 4 enable	RW
b3	ITEN3	Dedicated match interrupt 3 enable	0: Dedicated count compare match event 3 disable 1: Dedicated count compare match event 3 enable	RW
b2	ITEN2	Dedicated match interrupt 2 enable	0: Dedicated count compare match event 2 disable 1: Dedicated count compare match event 2 enable	RW
b1	ITEN1	Dedicated match interrupt 1 enable	0: Dedicated count compare match event 1 disable 1: Dedicated count compare match event 1 enable	RW
b0	ITEN0	Dedicated match interrupt 0 enable	0: Dedicated count compare match event 0 disable 1: Dedicated count compare match event 0 enable	RW

20.5.19 Dedicated Flag Control Register (TMR4_SCFR)

Reset Value: 0xFF00

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res								SF7	SF6	SF5	SF4	SF3	SF2	SF1	SF0

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b15~b8	Res	Reserved Bit	Read at “1”, write at “1”	RW
b7	SF7	Dedicated count match flag bit 7	0: No dedicated count compare match event occurred 7 1: Dedicated count compare match event 7 occurs	RW
b6	SF6	Dedicated count match flag bit 6	0: No dedicated count compare match event occurred 6 1: Dedicated count compare match event 6 occurs	RW
b5	SF5	Dedicated count match flag bit 5	0: No dedicated count compare match event occurred 5 1: Dedicated count compare match event 5 occurs	RW
b4	SF4	Dedicated count match flag bit 4	0: No dedicated count compare match event occurred 4 1: Dedicated count compare match event 4 occurs	RW
b3	SF3	Dedicated count match flag bit 3	0: No dedicated count compare match event occurred 3 1: Dedicated count compare match event 3 occurs	RW
b2	SF2	Dedicated count match flag bit 2	0: No dedicated count compare match event occurred 2 1: Dedicated count compare match event 2 occurs	RW
b1	SF1	Dedicated count match flag bit 1	0: No dedicated count compare match event occurred 1 1: Dedicated count compare match event 1 occurs	RW
b0	SF0	Dedicated count match flag bit 0	0: No dedicated count compare match event occurred 0 1: Dedicated count compare match event 0 occurs	RW

21 Emergency Brake Module (EMB)

21.1 Overview

The emergency brake module is a functional module that generates a control event and outputs it to the timer when certain conditions are met, so as to control the timer to stop or change the PWM signal output to the external motor. The following sources can be used to generate control events:

- Change of input level of external port
- Level of PWM output port occurs in phase (same high or same low)
- Voltage comparator comparison results
- System error occurred
- Write register software control

21.2 Functional Description

EMB is used to output a control event signal to the timer module with PWM function (Timer4, Timer6, HRPWM) when certain conditions are met, notifying the timer module to turn off the current PWM output. The EMB module has 9 groups, among which group0~group5 are used to control HRPWM, group6~group7 are used to control Timer6, and group8 is used to control Timer4, each corresponding to one unit of Timer4. The EMB structure block diagram is shown in the Figure 21-1.

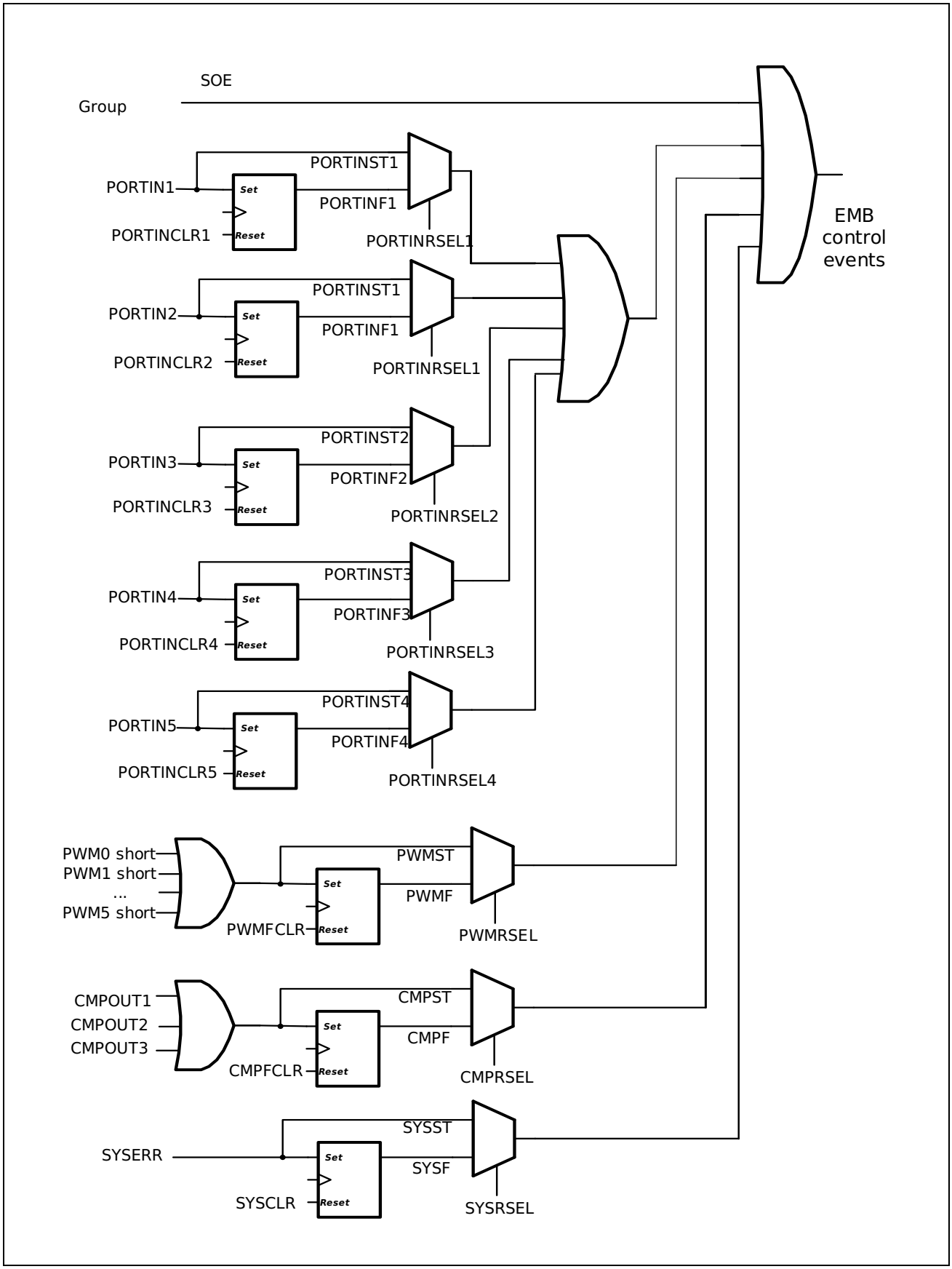


Figure 21-1 EMB Channel Functional Block Diagram

21.2.1 EMB Structure Block Diagram

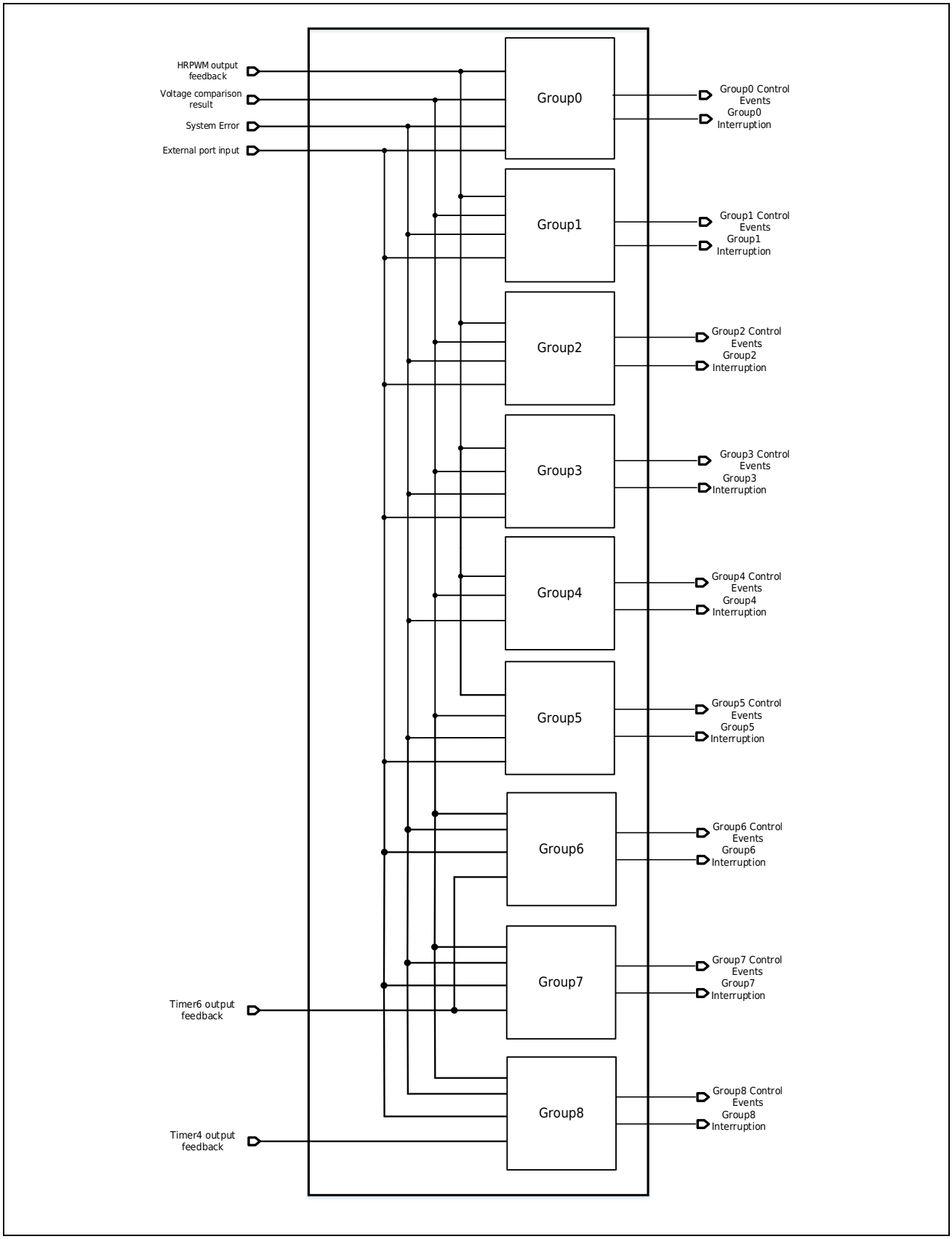


Figure 21-2 EMB Block Diagram

21.2.2 Control PWM Signal Output When External Port Input Level Changes

EMB has a total of 5 external ports for controlling PWM signal output when the input level changes. Each group can independently set one or more of the five external ports to be valid. The port allocation of group0~5 is shown in Table 21-1, and the port allocation of group6~8 is shown in Table 21-2.

Table 21-1 Port Allocation Table

port	PIN Name
EMB_PORT1_IN1	PA6, PB12, PA12, PB7
EMB_PORT1_IN2	PA15, PA11, PA6, PC3, PF3
EMB_PORT1_IN3	PA0, PA9, PB10, PC5, PB8
EMB_PORT1_IN4	PB11, PB9, PA1, PF2
EMB_PORT1_IN5	PC7, PA10, PA13

Table 21-2 Port Allocation Table

port	PIN Name
EMB_PORT2_IN1	PA6, PB12, PA14, PA15, PB8
EMB_PORT2_IN2	PA6, PA11, PC3
EMB_PORT2_IN3	PA0, PA9, PC5
EMB_PORT2_IN4	PB5, PA1, PF2
EMB_PORT2_IN5	PD2, PA10, PB4

When using the external port input level change to control the PWM signal output, first set the enable bit, that is, the EMB control register EMB_CTL1_x. PORTINENy (x=0~8, y=1~5) to be valid, and at the same time, set EMB_CTL1_x.INVSEL y (x=0~8, y=1~5) to generate a control signal when the port level is high (INVSEL=0) or low (INVSEL=1). Enable the filter function and set the filter clock as needed through EMB_CTL2_x.NFENy (x=0~8, y=1~5), EMB_CTL2_x.NFSELy (x=0~8, y=1~5), and EMB_CTL2_x.NFSELy_2 (x=0~8, y=1~5).

After the filter samples the input signal according to the filter clock, it performs filtering in a way that the number of comparisons is consistent N times. The number of comparisons is configured by EMB_CTL2_x.NFMD. When the filter clock samples the same level on the port N times, the level is regarded as a valid level and sent to the module; the level that is less than N times consistent will be filtered out as external interference and not sent to the module.

After configuring the EMB_CTL2_x and EMB_CTL1_x registers, write 1 to EMB_STATCLRx.PORTINFCLRy, confirm that EMB_STATx.PORTINFy is 0, and then configure the EMB_INTENx.PORTINENy bit to enable interrupts.

When a valid level that meets the conditions is generated on the port, the port input status EMB_STATx.PORTINSTy (x=0~8, y=1~5) of the EMB status register is set, and the port input control

flag EMB_STATx.PORTINFy (x=0~8, y=1~5) is set. An interrupt will be generated when the interrupt enable register EMB_INTENx.PORTINEN (x=0~8)=1.

The control events of EMB are output to HRPWM, Timer6 and Timer4 through EMB_STATx.PORTINSTy (x=0~8, y=1~5) or EMB_STATx.PORTINFy (x=0~8, y=1~5) via the EMB release mode selection register EMB_RLSSELx.PORTINRSELy (x=0~8, y=1~5). After receiving a control event, HRPWM, Timer6 and Timer4 can set the output port to high level, low level or high impedance state according to the register settings.

When releasing control, if EMB_RLSSELx. PORTINRSELy (x=0~8, y=1~5)=1, when the port input level becomes invalid, EMB_STATx.PORTINSTy (x=0~8, y=1~5) will be automatically cleared and the control event will be released immediately. If EMB_RLSSELx.PORTINRSELy (x=0~8, y=1~5)=0, when the port input level becomes invalid, it is necessary to write the EMB status reset register EMB_STATCLRx.PORTINFCLR (x=0~8) to clear EMB_STATx.PORTINF (x=0~8) to release the control event.

21.2.3 Stop PWM Signal Output When The Level Of PWM Output Port Is In The Same Phase (Same High Or Same Low)

When using, first set the enable bit, that is, the EMB control register EMB_CTL1_x.PWMSEN y (x=0~5, y=0~5 or x=6~7, y=0~3 or x=8, y=0~3) to be valid, and at the same time select the valid level to be monitored through the EMB control register EMB_CTL2_x.PWMLVy (x=0~5, y=0~5 or x=6~7, y=0~3 or x=8, y=0~3).

After the setup is completed, EMB monitors the complementary PWM output signals of Timer6/HRPWM and Timer4. When the output signals are at the same high or low level, the PWM output status EMB_STATx.PWMST (x=0~8) of the EMB status register is set, and the PWM output same-phase control flag EMB_STATx.PWMSF (x=0~8) is set. When the interrupt enable register EMB_INTENx. PWMEN (x=0~8)=1, an interrupt will be generated.

The control events of EMB are output to Timer6 and Timer4 through EMB_STATx. PWMST (x=0~8) or EMB_STATx. PWMSF (x=0~8) through the EMB release mode selection register EMB_RLSSELx.PWMRSEL (x=0~8). After receiving a control event, Timer6/HRPWM and Timer4 can set the output port to high level, low level or high impedance state according to the register settings.

When releasing control, if EMB_RLSSELx. PWMRSEL (x = 0 ~ 8) = 1, when the PWM output is released from the high or low state, EMB_STATx.PWMST (x = 0 ~ 8) will be automatically cleared and the control event will be released immediately. If EMB_RLSSELx. PWMST (x=0~8) = 0, when the PWM output is released from the high or low state, it is necessary to write the EMB status reset register EMB_STATCLRx. PWMSFCLR (x=0~8) to clear EMB_STATxPWMSF (x=0~8) to 0 and then release the control event.

Table 21-3 EMB Group Comparison Table

Port Name	Function	Corresponding Group	EMB_CTL1 Control bits	EMB_CTL2 Control bits
HRPWM_m_PWMA (m=1~6)	Complementary PWM output signal of HRPWM	Group0~5	PWMSEN[5:0]	PWMLV[5:0]
HRPWM_m_PWMB (m=1~6)				
TMR6_m_PWMA (m=1~4)	Complementary PWM Output Signal of Timer6	Group6~7	PWMSEN[3:0]	PWMLV[3:0]
TMR6_m_PWMB (m=1~4)				
TMR4_OXH	Complementary PWM Output Signal of Timer4_1	Group8	PWMSEN[3]	PWMLV[3]
TMR4_OXL				
TMR4_OUH			PWMSEN[2]	PWMLV[2]
TMR4_OUL				
TMR4_OVH			PWMSEN[1]	PWMLV[1]
TMR4_OVL				
TMR4_OWH			PWMSEN[0]	PWMLV[0]
TMR4_OWL				

21.2.4 Stop PWM Signal Output According To Voltage Comparator Comparison Result

Each Group of EMB can send control event signals to Timer6/HRPWM and Timer4 according to the three comparison results of the voltage comparator. For setting the voltage comparator output result, please refer to the [Voltage Comparator (CMP)] section.

When using, first set the enable bit, that is, the EMB control register EMB_CTL1_x. CMPEN y (x=0~8) to valid. When the voltage comparator comparison result flag is set, the voltage comparator status EMB_STATx.CMPST (x=0~8) of the EMB status register is set, and the EMB voltage comparator control flag EMB_STATx.CMPF (x=0~8) is set. When the interrupt enable register EMB_INTENx. CMPINTEN (x=0~8)=1, an interrupt will be generated. When multiple voltage comparators are valid and the EMB voltage comparator control flag EMB_STATx.CMPF (x=0~8) is set, the source of the comparator is confirmed by the EMB voltage comparator flag EMB_STATx.CMP<y>F (x=0~8).

The control events of EMB are output to Timer6 and Timer4 through EMB_STATx.CMPST (x=0~8) or EMB_STATx.CMPF (x=0~8) via the EMB release mode selection register EMB_RLSSELx.CMPRSELy (x=0~8). After receiving the control event signal, Timer6 and Timer4 can set the output port to high level, low level or high impedance state according to the register setting.

When releasing control, if EMB_RLSSELx. CMPRSEL (x = 0 ~ 8) = 1, when the voltage comparator result becomes invalid, EMB_STATx.CMPST (x = 0 ~ 8) will be automatically cleared and the control event will be released immediately. If EMB_RLSSELx.CMPRSELy (x=0~8)=0, when the voltage comparator result becomes invalid, you need to write the EMB status reset register EMB_STATCLRx.CMPFCLR (x=0~8) to clear EMB_STATx.CMPF (x=0~8) to 0, or clear all EMB voltage comparator flags EMB_STATx.CMP<y>F to release the control event.

After configuring the EMB_CTL2_x and EMB_CTL1_x registers, write 1 to EMB_STATCLRx.CMPFCLR, confirm that EMB_STATx.CMPF is 0, and then configure the EMB_INTENx.CMPINTEN bit to enable interrupts.

21.2.5 Stop PWM Signal Output When System Error Occurs

Each Group of EMB can send notification signals to Timer6 and Timer4 when a system error occurs. System error signals come from external oscillator stop, RAM (CACHERAM) parity check error and SRAM ECC check error, Cortex-M4 lock signal and PVD detection.

When using, first set the enable bit, that is, the EMB control register EMB_CTL1_x. SYSEN (x=0~8) to be valid, and set the EMB_CTL1_x. PVDEN, LOCKUPEN, SRAMPYERREN, SRAMECCERREN, and OSCSTPEN bits to enable the need to detect system error sources. When the selected system error source flag is set, the system error status EMB_STATx.SYSST (x=0~8) of the EMB status register is set, and the EMB system error control flag EMB_STATx.SYSF (x=0~8) is set. An interrupt will also be generated when the interrupt enable register EMB_INTENx. SYSINTEN (x = 0 ~ 8) = 1. When multiple system error sources are valid and the EMB system error control flag EMB_STATx.SYSF (x=0~8) is set, the cause of the system error is confirmed through the PVD detection flag, Cortex®-M4 LOCKUP flag, RAM (CACHERAM) parity check error flag, SRAM ECC check error flag and oscillator stop flag of EMB_STATx (x=0~8).

The control events of EMB are output to Timer6, HRPWM and Timer4 through EMB_STATx.SYSST (x=0~8) or EMB_STATx.SYSF (x=0~8) via the EMB release mode selection register EMB_RLSSELx.SYSRSELy (x=0~8). After receiving the control event signal, Timer6 and Timer4 can set the output port to high level, low level or high impedance state according to the register setting.

When releasing control, if EMB_RLSSELx.SYSRSEL (x=0~8)=1, when all enabled system error source flag bits become invalid (refer to the corresponding module for clearing the corresponding flag bits), EMB_STATx.SYSST (x=0~8) will be automatically cleared and the control event will be released immediately. If EMB_RLSSELx.SYSRSELy (x=0~8)=0, when all enabled system error source flag bits become invalid (refer to the corresponding module for clearing the corresponding flag bits), it is necessary to write the EMB status reset register EMB_STATCLRx.SYSFCLR (x=0~8) to clear EMB_STATx.SYSF (x=0~8) to 0, or write bits b19 to b23 of the EMB status reset register EMB_STATCLRx to clear each system error flag bit to release the control event.

Note:

When RAM (CACHERAM) parity check error, SRAM ECC error and PVD detection are enabled (EMB_CTL1_x. PVDEN, RAMPYERREN, SRAMECCERREN bits are 1), no matter EMB_RLSSELx. SYSRSEL (x=0~8) is set to 0 or 1, the corresponding flag bits in the relevant modules need to be cleared so that EMB_STATx.SYSST (x=0~8) can become invalid. When a RAM (CACHERAM) parity check error occurs, the CACHE_PYERR bit of SRAMC_CKSR needs to be cleared; when an SRAM ECC check error occurs, the SRAMB_2ERR, SRAMB_1ERR, SRAM0_2ERR, SRAM0_1ERR and

SRAMH_PYERR bits of SRAMC_CKSR need to be cleared; when PVD detection occurs, the PVD1DETFLG and PVD2DETFLG bits of PWC_PVDDSR need to be cleared.

Before RAM (CACHERAM) parity check error, SRAM ECC error and PVD detection are enabled, the flag bits in the above corresponding modules need to be cleared to prevent EMB from generating erroneous control event signals and sendting them to HRPWM, Timer6 and Timer4.

21.2.6 Write register software control PWM signal output

The software output enable control register (EMB_SOE) of EMB allows the user to send control signals to HRPWM, Timer6 and Timer4 by directly setting and resetting the software. No interrupt request will be generated when the software controls the PWM output.

21.3Register Description

Table 21-4 EMB Base Address

Name	Base Address	Description
EMB0	0x4003 8600	EMB0 base address
EMB1	0x4003 8620	EMB1 base address
EMB2	0x4003 8640	EMB2 base address
EMB3	0x4003 8660	EMB3 base address
EMB4	0x4003 8680	EMB4 base address
EMB5	0x4003 86A0	EMB5 base address
EMB6	0x4003 86C0	EMB6 base address
EMB7	0x4003 86E0	EMB7 base address
EMB8	0x4003 8700	EMB8 base address

Table 21-5 EMB Register List

Register	Description	Offset Address	Bit Width	Reset Value	Description
EMB_CTL1	EMB control register 1	0x00	32	0x0000 0000	Enable of each PWM output control event
EMB_CTL2	EMB control register 2	0x04	32	0x0000 0000	Filtering of port input control events and selection of effective level of PWM feedback signal
EMB_SOE	EMB software output enable control register	0x08	32	0x0000 0000	Software generates PWM output control events
EMB_STAT	EMB status register	0x0C	32	0x0000 0000	Indicates the status of PWM output control
EMB_STATCLR	EMB state reset register	0x10	32	0x0000 0000	Clear the status of PWM output control
EMB_INTEN	EMBinterrupt enable register	0x14	32	0x0000 0000	Interrupt enable
EMB_RLSSEL	EMB control PWM output release mode selection register	0x18	32	0x0000 0000	Select the release method for each PWM output control event

21.3.1 EMB Control Register 1 (EMB_CTL1, Group0~5)

The register is a single write register that can only be written once after a reset

Address: 0x4003 8600, 0x4003 8620, 0x4003 8640, 0x4003 8660, 0x4003 8680, 0x4003 86A0

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
PVDEN	LOCKUPEN	RAMPYERREN	SRAMECCERREN	OSCSTPEN	INVSEL5	INVSEL4	INVSEL3	INVSEL2	INVSEL1	Res	PORTINEN5	PORTINEN4	PORTINEN3	PORTINEN2	PORTINEN1
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res					PWMSEN5	PWMSEN4	PWMSEN3	PWMSEN2	PWMSEN1	PWMSEN0	SYSEN	Res	CMPEN3	CMPEN2	CMPEN1

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31	PVDEN	PVD detection control enable	0: PVD test result output control invalid 1: PVD test result output control valid	RW
b30	LOCKUPEN	Cortex®-M4 Lockup control enable	0: Output control invalid when Lockup occurs 1: Output control valid when Lockup occurs	RW
b29	RAMPYERREN	RAM parity check error control enable	0: Output control invalid when RAM (CACHERAM) parity check error occurs 1: Output control valid when RAM (CACHERAM) parity check error occurs	RW
b28	SRAMECCERREN	SRAM ECC check error control enable	0: Output control invalid when SRAM ECC check error occurs 1: Output control valid when SRAM ECC check error occurs	RW
b27	OSCSTPEN	Oscillator stop control enable	0: Output control invalid when oscillator stops 1: Output control valid when oscillator stops	RW
b26	INVSEL5	Port 5 input active level selection	0: High level valid 1: Low level valid	RW
b25	INVSEL4	Port 4 input active level selection	0: High level valid 1: Low level valid	RW
b24	INVSEL3	Port 3 input active level selection	0: High level valid 1: Low level valid	RW
b23	INVSEL2	Port 2 input active level selection	0: High level valid 1: Low level valid	RW
b22	INVSEL1	Port 1 input active level selection	0: High level valid 1: Low level valid	RW
b21	Res	Reserved Bit	Read “0” when reading, write“0” when writing	RW
b20	PORTINEN5	Port 5 input control enable	0: Invalid port input control 1: Port input control is valid	RW
b19	PORTINEN4	Port 4 input control enable	0: Invalid port input control 1: Port input control is valid	RW
b18	PORTINEN3	Port 3 input control enable	0: Invalid port input control 1: Port input control is valid	RW
b17	PORTINEN2	Port 2 input control enable	0: Invalid port input control 1: Port input control is valid	RW
b16	PORTINEN1	Port 1 input control enable	0: Invalid port input control 1: Port input control is valid	RW
b15~b11	Res	Reserved Bit	Read “0” when reading, write“0” when writing	RW
b10	PWMSEN5	HRPWM_6_PWMA/B short circuit output control enable	0: Invalid output control when short circuit 1: Output control effective when short circuit	RW
b9	PWMSEN4	HRPWM_5_PWMA/B short circuit output control enable	0: Invalid output control when short circuit 1: Output control effective when short circuit	RW
b8	PWMSEN3	HRPWM_4_PWMA/B short-circuit output control enable	0: Invalid output control when short circuit 1: Output control effective when short circuit	RW
b7	PWMSEN2	HRPWM_3_PWMA/B short circuit output control enable	0: Invalid output control when short circuit 1: Output control effective when short circuit	RW
b6	PWMSEN1	HRPWM_2_PWMA/B short circuit output control enable	0: Invalid output control when short circuit 1: Output control effective when short circuit	RW
b5	PWMSEN0	HRPWM_1_PWMA/B short circuit output control enable	0: Invalid output control when short circuit 1: Output control effective when short circuit	RW
b4	SYSEN	System error control enable	0: Output control invalid when a system error occurs 1: Output control valid when a system error occurs	RW
b3	Res	Reserved Bit	Read “0” when reading, write“0” when writing	RW

b2	CMPEN3	CMP3 Voltage comparator comparison result control enable	0: Invalid voltage comparator output control 1: Voltage comparator output control is effective	RW
b1	CMPEN2	CMP2 Voltage comparator comparison result control enable	0: Invalid voltage comparator output control 1: Voltage comparator output control is effective	RW
b0	CMPEN1	CMP1 Voltage comparator comparison result control enable	0: Invalid voltage comparator output control 1: Voltage comparator output control is effective	RW

21.3.2 EMB Control Register 1 (EMB_CTL1, Group 6~7)

The register is a single write register that can only be written once after a reset

Address: 0x4003 86C0, 0x4003 86E0

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
PVDEN	LOCKUPEN	RAMPYERREN	SRAMECCERREN	OSCSTPEN	INVSEL5	INVSEL4	INVSEL3	INVSEL2	INVSEL1	Res	PORTINEN5	PORTINEN4	PORTINEN3	PORTINEN2	PORTINEN1
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res							PWMSEN3	PWMSEN2	PWMSEN1	PWMSEN0	SYSEN	Res	CMPEN3	CMPEN2	CMPEN1

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31	PVDEN	PVD detection control enable	0: PVD test result output control invalid 1: PVD test result output control valid	RW
b30	LOCKUPEN	Cortex®-M4 Lockup control enable	0: Output control invalid when Lockup occurs 1: Output control valid when Lockup occurs	RW
b29	RAMPYERREN	RAM parity check error control enable	0: Output control invalid when RAM (CACHERAM) parity check error occurs 1: Output control valid when RAM (CACHERAM) parity check error occurs	RW
b28	SRAMECCERREN	SRAM ECC check error control enable	0: Output control invalid when SRAM ECC check error occurs 1: Output control valid when SRAM ECC check error occurs	RW
b27	OSCSTPEN	Oscillator stop control enable	0: Output control invalid when oscillator stops 1: Output control valid when oscillator stops	RW
b26	INVSEL5	Port 5 input active level selection	0: High level valid 1: Low level valid	RW
b25	INVSEL4	Port 4 input active level selection	0: High level valid 1: Low level valid	RW
b24	INVSEL3	Port 3 input active level selection	0: High level valid 1: Low level valid	RW
b23	INVSEL2	Port 2 input active level selection	0: High level valid 1: Low level valid	RW
b22	INVSEL1	Port 1 input active level selection	0: High level valid 1: Low level valid	RW
b21	Res	Reserved Bit	Read “0” when reading, write“0” when writing	RW
b20	PORTINEN5	Port 5 input control enable	0: Invalid port input control 1: Port input control is valid	RW
b19	PORTINEN4	Port 4 input control enable	0: Invalid port input control 1: Port input control is valid	RW
b18	PORTINEN3	Port 3 input control enable	0: Invalid port input control 1: Port input control is valid	RW
b17	PORTINEN2	Port 2 input control enable	0: Invalid port input control 1: Port input control is valid	RW
b16	PORTINEN1	Port 1 input control enable	0: Invalid port input control 1: Port input control is valid	RW
b15~b9	Res	Reserved Bit	Read “0” when reading, write“0” when writing	RW
b8	PWMSEN3	TMR6_4_PWMA/B short-circuit output control enable	0: Invalid output control when short circuit 1: Output control effective when short circuit	RW
b7	PWMSEN2	TMR6_3_PWMA/B short-circuit output control enable	0: Invalid output control when short circuit 1: Output control effective when short circuit	RW
b6	PWMSEN1	TMR6_2_PWMA/B short-circuit output control enable	0: Invalid output control when short circuit 1: Output control effective when short circuit	RW
b5	PWMSEN0	TMR6_1_PWMA/B short-circuit output control enable	0: Invalid output control when short circuit 1: Output control effective when short circuit	RW
b4	SYSEN	System error control enable	0: Output control invalid when a system error occurs 1: Output control valid when a system error occurs	RW
b3	Res	Reserved Bit	Read “0” when reading, write“0” when writing	RW

b2	CMPEN3	CMP3 Voltage comparator comparison result control enable	0: Invalid voltage comparator output control 1: Voltage comparator output control is effective	RW
b1	CMPEN2	CMP2 Voltage comparator comparison result control enable	0: Invalid voltage comparator output control 1: Voltage comparator output control is effective	RW
b0	CMPEN1	CMP1 Voltage comparator comparison result control enable	0: Invalid voltage comparator output control 1: Voltage comparator output control is effective	RW

21.3.3 EMB Control Register 1 (EMB_CTL1, Group 8)

The register is a single write register that can only be written once after a reset

Address: 0x4003 8700

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
PVDEN	LOCKUPEN	RAMPYERREN	SRAMECCERREN	OSCSTPEN	INVSEL5	INVSEL4	INVSEL3	INVSEL2	INVSEL1	Res	PORTINEN5	PORTINEN4	PORTINEN3	PORTINEN2	PORTINEN1
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res							PWMSEN3	PWMSEN2	PWMSEN1	PWMSEN0	SYSEN	Res	CMPEN3	CMPEN2	CMPEN1

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31	PVDEN	PVD detection control enable	0: PVD test result output control invalid 1: PVD test result output control valid	RW
b30	LOCKUPEN	Cortex®-M4 Lockup control enable	0: Output control invalid when Lockup occurs 1: Output control valid when Lockup occurs	RW
b29	RAMPYERREN	RAM parity check error control enable	0: Output control invalid when RAM (CACHERAM) parity check error occurs 1: Output control valid when RAM (CACHERAM) parity check error occurs	RW
b28	SRAMECCERREN	SRAM ECC check error control enable	0: Output control invalid when SRAM ECC check error occurs 1: Output control valid when SRAM ECC check error occurs	RW
b27	OSCSTPEN	Oscillator stop control enable	0: Output control invalid when oscillator stops 1: Output control valid when oscillator stops	RW
b26	INVSEL5	Port 5 input active level selection	0: High level valid 1: Low level valid	RW
b25	INVSEL4	Port 4 input active level selection	0: High level valid 1: Low level valid	RW
b24	INVSEL3	Port 3 input active level selection	0: High level valid 1: Low level valid	RW
b23	INVSEL2	Port 2 input active level selection	0: High level valid 1: Low level valid	RW
b22	INVSEL1	Port 1 input active level selection	0: High level valid 1: Low level valid	RW
b21	Res	Reserved Bit	Read “0” when reading, write“0” when writing	RW
b20	PORTINEN5	Port 5 input control enable	0: Invalid port input control 1: Port input control is valid	RW
b19	PORTINEN4	Port 4 input control enable	0: Invalid port input control 1: Port input control is valid	RW
b18	PORTINEN3	Port 3 input control enable	0: Invalid port input control 1: Port input control is valid	RW
b17	PORTINEN2	Port 2 input control enable	0: Invalid port input control 1: Port input control is valid	RW
b16	PORTINEN1	Port 1 input control enable	0: Invalid port input control 1: Port input control is valid	RW
b15~b9	Res	Reserved Bit	Read “0” when reading, write“0” when writing	RW
b8	PWMSEN3	TMR4_OXH/L Short-circuit output control enable	0: Invalid output control when short circuit 1: Output control effective when short circuit	RW
b7	PWMSEN2	TMR4_OUH/L short-circuit output control enable	0: Invalid output control when short circuit 1: Output control effective when short circuit	RW
b6	PWMSEN1	TMR4_OVH/L short-circuit output control enable	0: Invalid output control when short circuit 1: Output control effective when short circuit	RW
b5	PWMSEN0	TMR4_OWH/L short-circuit output control enable	0: Invalid output control when short circuit 1: Output control effective when short circuit	RW
b4	SYSEN	System error control enable	0: Output control invalid when a system error occurs 1: Output control valid when a system error occurs	RW
b3	Res	Reserved Bit	Read “0” when reading, write“0” when writing	RW
b2	CMPEN2	CMP3 Voltage comparator comparison result control enable	0: Invalid voltage comparator output control 1: Voltage comparator output control is effective	RW

b1	CMPEN1	CMP2 Voltage comparator comparison result control enable	0: Invalid voltage comparator output control 1: Voltage comparator output control is effective	RW
b0	CMPEN0	CMP1 Voltage comparator comparison result control enable	0: Invalid voltage comparator output control 1: Voltage comparator output control is effective	RW

21.3.4 EMB Control Register 2 (EMB_CTL2, Group0~5)

The register is a single write register that can only be written once after a reset

Address: 0x4003 8604, 0x4003 8624, 0x4003 8644, 0x4003 8664, 0x4003 8684, 0x4003 86A4

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res	NFEN5	NFSEL5[1:0]		NFEN4	NFSEL4[1:0]		NFEN3	NFSEL3[1:0]		NFEN2	NFSEL2[1:0]		NFEN1	NFSEL1[1:0]	

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
NFMD	NFSEL 5_2	NFSEL 4_2	NFSEL 3_2	NFSEL 2_2	NFSEL 1_2	Res				PWML V5	PWML V4	PWML V3	PWML V2	PWML V1	PWML V0

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31	Res	Reserved Bit	Read “0” when reading, write“0” when writing	RW
b30	NFEN5	Port 5 input digital filter enable	0: Invalid filter 1: Filter is valid	
b29~b28	NFSEL5[1:0]	Port 5 digital filter clock selection	This bit is used in combination with NFSEL5_2 When NFSEL5_2=0 00: Use PCLK0 filtering 01: Use the 8-frequency filter of PCLK0 10: Use 32-division filtering of PCLK0 11: Use 128-division filtering of PCLK0 When NFSEL5_2=1 00: Use the 2-way filter of PCLK0 01: Use the 4-division filter of PCLK0 10: Use the 16-frequency filter of PCLK0 11: Use the 64-division filter of PCLK0	
b27	NFEN4	Port 4 input digital filter enable	0: Invalid filter 1: Filter is valid	RW
b26~b25	NFSEL4[1:0]	Port 4 digital filter clock selection	This bit is used in combination with NFSEL4_2 When NFSEL4_2=0 00: Use PCLK0 filtering 01: Use the 8-frequency filter of PCLK0 10: Use 32-division filtering of PCLK0 11: Use 128-division filtering of PCLK0 When NFSEL4_2 = 1 00: Use the 2-way filter of PCLK0 01: Use the 4-division filter of PCLK0 10: Use the 16-frequency filter of PCLK0 11: Use the 64-division filter of PCLK0	RW
b24	NFEN3	Port 3 input digital filter enable	0: Invalid filter 1: Filter is valid	RW
b23~b22	NFSEL3[1:0]	Port 3 digital filter clock selection	This bit is used in combination with NFSEL3_2 When NFSEL3_2=0 00: Use PCLK0 filtering 01: Use the 8-frequency filter of PCLK0 10: Use 32-division filtering of PCLK0 11: Use 128-division filtering of PCLK0 When NFSEL3_2=1 00: Use the 2-way filter of PCLK0 01: Use the 4-division filter of PCLK0 10: Use the 16-frequency filter of PCLK0 11: Use the 64-division filter of PCLK0	RW
b21	NFEN2	Port 2 input digital filter enable	0: Invalid filter 1: Filter is valid	RW
b20~b19	NFSEL2[1:0]	Port 2 digital filter clock selection	This bit is used in combination with NFSEL2_2 When NFSEL2_2=0 00: Use PCLK0 filtering 01: Use the 8-frequency filter of PCLK0 10: Use 32-division filtering of PCLK0 11: Use 128-division filtering of PCLK0 When NFSEL2_2=1 00: Use the 2-way filter of PCLK0 01: Use the 4-division filter of PCLK0 10: Use the 16-frequency filter of PCLK0 11: Use the 64-division filter of PCLK0	RW

b18	NFEN1	Port 1 input digital filter enable	0: Invalid filter 1: Filter is valid	RW
b17~b16	NFSEL1[1:0]	Port 1 digital filter clock selection	This bit is used in combination with NFSEL1_2 When NFSEL1_2=0 00: Use PCLK0 filtering 01: Use the 8-frequency filter of PCLK0 10: Use 32-division filtering of PCLK0 11: Use 128-division filtering of PCLK0 When NFSEL1_2=1 00: Use the 2-way filter of PCLK0 01: Use the 4-division filter of PCLK0 10: Use the 16-frequency filter of PCLK0 11: Use the 64-division filter of PCLK0	RW
b15	NFMD	Filter times selection	0: The number of filtering times is 3 1: The number of filtering times is 2	RW
b14	NFSEL5_2	Port 5 digital filter filter clock selection 2	This bit is used in combination with NFSEL5[1:0]	RW
b13	NFSEL4_2	Port 4 digital filter filter clock selection 2	This bit is used in combination with NFSEL4[1:0]	RW
b12	NFSEL3_2	Port 3 digital filter filter clock selection 2	This bit is used in combination with NFSEL3[1:0]	RW
b11	NFSEL2_2	Port 2 digital filter filter clock selection 2	This bit is used in combination with NFSEL2[1:0]	RW
b10	NFSEL1_2	Port 1 digital filter filter clock selection 2	This bit is used in combination with NFSEL1[1:0]	RW
b9~b6	Res	Reserved Bit	Read “0” when reading, write“0” when writing	RW
b5	PWMLV5	HRPWM_6_PWMA/B output valid level selection	0: Low to active 1: High to active	RW
b4	PWMLV4	HRPWM_5_PWMA/B output valid level selection	0: Low to active 1: High to active	RW
b3	PWMLV3	HRPWM_4_PWMA/B output valid level selection	0: Low to active 1: High to active	RW
b2	PWMLV2	HRPWM_3_PWMA/B output valid level selection	0: Low to active 1: High to active	RW
b1	PWMLV1	HRPWM_2_PWMA/B output valid level selection	0: Low to active 1: High to active	RW
b0	PWMLV0	HRPWM_1_PWMA/B output valid level selection	0: Low to active 1: High to active	RW

21.3.5 EMB Control Register 2 (EMB_CTL2, Group 6~7)

The register is a single write register that can only be written once after a reset

Address: 0x4003 86C4, 0x4003 86E4

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res	NFEN5	NFSEL5[1:0]		NFEN4	NFSEL4[1:0]		NFEN3	NFSEL3[1:0]		NFEN2	NFSEL2[1:0]		NFEN1	NFSEL1[1:0]	

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
NFMD	NFSEL 5_2	NFSEL 4_2	NFSEL 3_2	NFSEL 2_2	NFSEL 1_2	Res						PWML V3	PWML V2	PWML V1	PWML V0

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31	Res	Reserved Bit	Read “0” when reading, write“0” when writing	RW
b30	NFEN5	Port 5 input digital filter enable	0: Invalid filter 1: Filter is valid	
b29~b28	NFSEL5[1:0]	Port 5 digital filter clock selection	This bit is used in combination with NFSEL5_2 When NFSEL5_2=0 00: Use PCLK0 filtering 01: Use the 8-frequency filter of PCLK0 10: Use the 2-way filter of PCLK0 11: Use 128-division filtering of PCLK0 When NFSEL5_2=1 00: Use the 2-way filter of PCLK0 01: Use the 4-division filter of PCLK0 10: Use the 16-frequency filter of PCLK0 11: Use the 64-division filter of PCLK0	
b27	NFEN4	Port 4 input digital filter enable	0: Invalid filter 1: Filter is valid	RW
b26~b25	NFSEL4[1:0]	Port 4 digital filter clock selection	This bit is used in combination with NFSEL4_2 When NFSEL4_2=0 00: Use PCLK0 filtering 01: Use the 8-frequency filter of PCLK0 10: Use 32-division filtering of PCLK0 11: Use 128-division filtering of PCLK0 When NFSEL4_2 = 1 00: Use the 2-way filter of PCLK0 01: Use the 4-division filter of PCLK0 10: Use the 16-frequency filter of PCLK0 11: Use the 64-division filter of PCLK0	RW
b24	NFEN3	Port 3 input digital filter enable	0: Invalid filter 1: Filter is valid	RW
b23~b22	NFSEL3[1:0]	Port 3 digital filter clock selection	This bit is used in combination with NFSEL3_2 When NFSEL3_2=0 00: Use PCLK0 filtering 01: Use the 8-frequency filter of PCLK0 10: Use 32-division filtering of PCLK0 11: Use 128-division filtering of PCLK0 When NFSEL3_2=1 00: Use the 2-way filter of PCLK0 01: Use the 4-division filter of PCLK0 10: Use the 16-frequency filter of PCLK0 11: Use the 64-division filter of PCLK0	RW
b21	NFEN2	Port 2 input digital filter enable	0: Invalid filter 1: Filter is valid	RW
b20~b19	NFSEL2[1:0]	Port 2 digital filter clock selection	This bit is used in combination with NFSEL2_2 When NFSEL2_2=0 00: Use PCLK0 filtering 01: Use the 8-frequency filter of PCLK0 10: Use 32-division filtering of PCLK0 11: Use 128-division filtering of PCLK0 When NFSEL2_2=1 00: Use the 2-way filter of PCLK0 01: Use the 4-division filter of PCLK0 10: Use the 16-frequency filter of PCLK0 11: Use the 64-division filter of PCLK0	RW

b18	NFEN1	Port 1 input digital filter enable	0: Invalid filter 1: Filter is valid	RW
b17~b16	NFSEL1[1:0]	Port 1 digital filter clock selection	This bit is used in combination with NFSEL1_2 When NFSEL1_2=0 00: Use PCLK0 filtering 01: Use the 8-frequency filter of PCLK0 10: Use 32-division filtering of PCLK0 11: Use 128-division filtering of PCLK0 When NFSEL1_2=1 00: Use the 2-way filter of PCLK0 01: Use the 4-division filter of PCLK0 10: Use the 16-frequency filter of PCLK0 11: Use the 64-division filter of PCLK0	RW
b15	NFMD	Filter times selection	0: The number of filtering times is 3 1: The number of filtering times is 2	RW
b14	NFSEL5_2	Port 5 digital filter filter clock selection 2	This bit is used in combination with NFSEL5[1:0]	RW
b13	NFSEL4_2	Port 4 digital filter filter clock selection 2	This bit is used in combination with NFSEL4[1:0]	RW
b12	NFSEL3_2	Port 3 digital filter filter clock selection 2	This bit is used in combination with NFSEL3[1:0]	RW
b11	NFSEL2_2	Port 2 digital filter filter clock selection 2	This bit is used in combination with NFSEL2[1:0]	RW
b10	NFSEL1_2	Port 1 digital filter filter clock selection 2	This bit is used in combination with NFSEL1[1:0]	RW
b9~b4	Res	Reserved Bit	Read “0” when reading, write“0” when writing	RW
b3	PWMLV3	TMR6_4_PWMA/B output valid level selection	0: Low to active 1: High to active	RW
b2	PWMLV2	TMR6_3_PWMA/B output valid level selection	0: Low to active 1: High to active	RW
b1	PWMLV1	TMR6_2_PWMA/B output valid level selection	0: Low to active 1: High to active	RW
b0	PWMLV0	TMR6_1_PWMA/B output valid level selection	0: Low to active 1: High to active	RW

21.3.6 EMB Control Register 2 (EMB_CTL2, Group 8)

The register is a single write register that can only be written once after a reset

Address: 0x4003 8704

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res	NFEN5	NFSEL5[1:0]		NFEN4	NFSEL4[1:0]		NFEN3	NFSEL3[1:0]		NFEN2	NFSEL2[1:0]		NFEN1	NFSEL1[1:0]	

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
NFMD	NFSEL5_2	NFSEL4_2	NFSEL3_2	NFSEL2_2	NFSEL1_2	Res						PWMLV3	PWMLV2	PWMLV1	PWMLV0

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31	Res	Reserved Bit	Read “0” when reading, write“0” when writing	RW
b30	NFEN5	Port 5 input digital filter enable	0: Invalid filter 1: Filter is valid	-
b29~b28	NFSEL5[1:0]	Port 5 digital filter clock selection	This bit is used in combination with NFSEL5_2 When NFSEL5_2=0 00: Use PCLK0 filtering 01: Use the 8-frequency filter of PCLK0 10: Use 32-division filtering of PCLK0 11: Use 128-division filtering of PCLK0 When NFSEL5_2=1 00: Use the 2-way filter of PCLK0 01: Use the 4-division filter of PCLK0 10: Use the 16-frequency filter of PCLK0 11: Use the 64-division filter of PCLK0	-
b27	NFEN4	Port 4 input digital filter enable	0: Invalid filter 1: Filter is valid	RW
b26~b25	NFSEL4[1:0]	Port 4 digital filter clock selection	This bit is used in combination with NFSEL4_2 When NFSEL4_2=0 00: Use PCLK0 filtering 01: Use the 8-frequency filter of PCLK0 10: Use 32-division filtering of PCLK0 11: Use 128-division filtering of PCLK0 When NFSEL4_2 = 1 00: Use the 2-way filter of PCLK0 01: Use the 4-division filter of PCLK0 10: Use the 16-frequency filter of PCLK0 11: Use the 64-division filter of PCLK0	RW
b24	NFEN3	Port 3 input digital filter enable	0: Invalid filter 1: Filter is valid	RW
b23~b22	NFSEL3[1:0]	Port 3 digital filter clock selection	This bit is used in combination with NFSEL3_2 When NFSEL3_2=0 00: Use PCLK0 filtering 01: Use the 8-frequency filter of PCLK0 10: Use 32-division filtering of PCLK0 11: Use 128-division filtering of PCLK0 When NFSEL3_2=1 00: Use the 2-way filter of PCLK0 01: Use the 4-division filter of PCLK0 10: Use the 16-frequency filter of PCLK0 11: Use the 64-division filter of PCLK0	RW
b21	NFEN2	Port 2 input digital filter enable	0: Invalid filter 1: Filter is valid	RW
b20~b19	NFSEL2[1:0]	Port 2 digital filter clock selection	This bit is used in combination with NFSEL2_2 When NFSEL2_2=0 00: Use PCLK0 filtering 01: Use the 8-frequency filter of PCLK0 10: Use 32-division filtering of PCLK0 11: Use 128-division filtering of PCLK0 When NFSEL2_2=1 00: Use the 2-way filter of PCLK0 01: Use the 4-division filter of PCLK0 10: Use the 16-frequency filter of PCLK0 11: Use the 64-division filter of PCLK0	RW

b18	NFEN1	Port 1 input digital filter enable	0: Invalid filter 1: Filter is valid	RW
b17~b16	NFSEL1[1:0]	Port 1 digital filter clock selection	This bit is used in combination with NFSEL1_2 When NFSEL1_2=0 00: Using bus clock filtering 01: Use the 8-frequency filter of PCLK0 10: Use 32-division filtering of PCLK0 11: Use 128-division filtering of PCLK0 When NFSEL1_2=1 00: Use the 2-way filter of PCLK0 01: Use the 4-division filter of PCLK0 10: Use the 16-frequency filter of PCLK0 11: Use the 64-division filter of PCLK0	RW
b15	NFMD	Filter times selection	0: The number of filtering times is 3 1: The number of filtering times is 2	RW
b14	NFSEL5_2	Port 5 digital filter filter clock selection 2	This bit is used in combination with NFSEL5[1:0]	RW
b13	NFSEL4_2	Port 4 digital filter filter clock selection 2	This bit is used in combination with NFSEL4[1:0]	RW
b12	NFSEL3_2	Port 3 digital filter filter clock selection 2	This bit is used in combination with NFSEL3[1:0]	RW
b11	NFSEL2_2	Port 2 digital filter filter clock selection 2	This bit is used in combination with NFSEL2[1:0]	RW
b10	NFSEL1_2	Port 1 digital filter filter clock selection 2	This bit is used in combination with NFSEL1[1:0]	RW
b9~b4	Res	Reserved Bit	Read “0” when reading, write“0” when writing	RW
b3	PWMLV3	TMR4_OXH/L output valid level selection	0: Low to active 1: High to active	RW
b2	PWMLV2	TMR4_OUH/L output valid level selection	0: Low to active 1: High to active	RW
b1	PWMLV1	TMR4_OVH/L output valid level selection	0: Low to active 1: High to active	RW
b0	PWMLV0	TMR4_OWH/L output valid level selection	0: Low to active 1: High to active	RW

21.3.7 EMB Software Output Enable Control Register (EMB _ SOE)

Address: 0x4003 8608, 0x4003 8628, 0x4003 8648, 0x4003 8668, 0x4003 8688, 0x4003 86A8, 0x4003 86C8, 0x4003 86E8, 0x4003 8708

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res															SOE

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b1	Res	Reserved Bit	Read “0” when reading, write“0” when writing	RW
b0	SOE	Software control output	0: PWM normal output 1: PWM stop output	RW

21.3.8 EMB Status Register (EMB _ STAT)

Address: 0x4003 860C, 0x4003 862C, 0x4003 864C, 0x4003 866C, 0x4003 868C, 0x4003 86AC, 0x4003 86CC, 0x4003 86EC, 0x4003 870C

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res					CMP3F	CMP2F	CMP1F	PVDF	LOCKUPF	RAMPYERRF	SRAMECCERRF	OSCF	PORTINST5	PORTINST4	PORTINST3
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
PORTINST2	PORTINST1	Res	PORTINF5	PORTINF4	PORTINF3	PORTINF2	PORTINF1	SYSST	CMPST	PWMST	Res	SYSF	CMPF	PWMSF	Res

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b27	Res	Reserved Bit	Read “0” when reading, write“0” when writing	RW
b26	CMP3F	Voltage comparator 3 flag bit	Set when the voltage comparator 3 result is valid EMB_STATCLR.CMP3FCLR is cleared to 0 when writing 1 Or clear to 0 when EMB_STATCLR.CMPFCLR is written to 1	R
b25	CMP2F	Voltage comparator 2 flag bit	Set when the voltage comparator 2 result is valid EMB_STATCLR.CMP2FCLR is cleared to 0 when writing 1 Or clear to 0 when EMB_STATCLR.CMPFCLR is written to 1	R
b24	CMP1F	Voltage comparator 1 flag bit	Set when the voltage comparator 1 result is valid EMB_STATCLR.CMP1FCLR is cleared to 0 when writing 1 Or clear to 0 when EMB_STATCLR.CMPFCLR is written to 1	R
b23	PVDF	PVD detection flag bit	Set when the PVD test result is valid EMB_STATCLR.PVDFCLR is cleared to 0 when writing 1 Or EMB_STATCLR.SYSFCLR is cleared to 0 when writing 1	R
b22	LOCKUPF	Cortex®-M4 LOCKUP flag bit	Set when Cortex®-M4 LOCKUP occurs EMB_STATCLR.LOCKUPFCLR is cleared to 0 when writing 1 Or EMB_STATCLRx.SYSFCLR is cleared to 0 when writing 1	R
b21	RAMPYERRF	Cache RAM parity check error flag bit	Set when a cache RAM parity check error occurs EMB_STATCLR.RAMPYERRFCLR is cleared to 0 when writing 1 Or EMB_STATCLR.SYSFCLR is cleared to 0 when writing 1	R
b20	SRAMECCERRF	SRAM ECC check error flag bit	Set when an SRAM ECC check error occurs EMB_STATCLR.SRAMECCERRFCLR is cleared to 0 when writing 1 Or EMB_STATCLR.SYSFCLR is cleared to 0 when writing 1	R
b19	OSCF	Oscillator stop flag	Set when the oscillator stops EMB_STATCLRx. OSCFCLR is cleared to 0 when writing 1 Or EMB_STATCLRx.SYSFCLR is cleared to 0 when writing 1	R
b18	PORTINST5	Port 5 input control status	0: Port 5 input control is in invalid level state 1: Port 5 input control is in valid level state	R
b17	PORTINST4	Port 4 input control status	0: Port 4 input control is in invalid level state 1: Port 4 input control is in valid level state	R
b16	PORTINST3	Port 3 input control status	0: Port 3 input control is in invalid level state 1: Port 3 input control is in valid level state	R
b15	PORTINST2	Port 2 input control status	0: Port 2 input control is in invalid level state 1: Port 2 input control is in valid level state	R
b14	PORTINST1	Port 1 input control status	0: Port 1 input control is in invalid level state 1: Port 1 input control is in valid level state	R
b13	Res	Reserved Bit	Read “0” when reading, write“0” when writing	RW

b12	PORTINF5	Port 5 input control flag bit	Set when EMB_STAT.PORTINST5 is 1 EMB_STATCLR.PORTINFCLR5 is cleared to 0 when writing 1	R
b11	PORTINF4	Port 4 input control flag bit	Set when EMB_STAT.PORTINST4 is 1 EMB_STATCLR.PORTINFCLR4 is cleared to 0 when writing 1	R
b10	PORTINF3	Port 3 input control flag bit	Set when EMB_STAT.PORTINST3 is 1 EMB_STATCLR.PORTINFCLR3 is cleared to 0 when writing 1	R
b9	PORTINF2	Port 2 input control flag bit	Set when EMB_STAT.PORTINST2 is 1 EMB_STATCLR.PORTINFCLR2 is cleared to 0 when writing 1	R
b8	PORTINF1	Port 1 input control flag bit	Set when EMB_STAT.PORTINST1 is 1 EMB_STATCLR.PORTINFCLR1 is cleared to 0 when writing 1	R
b7	SYSST	System error status	0: No system error occurred 1: System error occurred	R
b6	CMPST	Voltage comparator status	Voltage comparator comparison results	R
b5	PWMST	PWM output state	0: No PWM output in phase 1: PWM output in phase	R
b4	Res	Reserved Bit	Read “0” when reading, write“0” when writing	RW
b3	SYSF	System error flag bit	Set when EMB_STAT.SYSST is 1 EMB_STATCLR.OSFCLR is cleared to 0 when writing 1 This bit is automatically cleared when bits b23-b19 are all cleared to 0.	R
b2	CMPF	Voltage comparator control flag	Set when EMB_STAT.CMPST is 1 EMB_STATCLR.CMPFCLR is cleared to 0 when writing 1 This bit is automatically cleared when bits b26-b24 are all cleared to 0.	R
b1	PWMSF	PWM output same phase control flag bit	Set when EMB_STAT.PWMST is 1 EMB_STATCLR.PWMSCLR is cleared to 0 when writing 1	R
b0	Res	Reserved Bit	Read “0” when reading, write“0” when writing	RW

21.3.9 EMB State Reset Register (EMB _ STATCLR)

Address: 0x4003 8610, 0x4003 8630, 0x4003 8650, 0x4003 8670, 0x4003 8690, 0x4003 86B0, 0x4003 86D0, 0x4003 86F0, 0x4003 8710

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res					CMP3FCLR	CMP2FCLR	CMP1FCLR	PVDFCLR	LOCKUPFCLR	RAMPYERRFCLR	SRAMECCERRFCLR	OSCFCLR	Res		

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res			PORTINFCLR5	PORTINFCLR4	PORTINFCLR3	PORTINFCLR2	PORTINFCLR1	Res				SYSFCLR	CMPFCLR	PWMSFCLR	Res

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b27	Res	Reserved Bit	Read “0” when reading, write“0” when writing	W
b26	CMP3FCLR	Reduction EMB_STAT.CMP3F	0: Nothing 1: Clear EMB_STAT.CMP3F to 0 Reading this register bit has no effect	W
b25	CMP2FCLR	Reduction EMB_STAT.CMP2F	0: Nothing 1: Clear EMB_STAT.CMP2F to 0 Reading this register bit has no effect	W
b24	CMP1FCLR	Reduction EMB_STAT.CMP1F	0: Nothing 1: Clear EMB_STAT.CMP1F to 0 Reading this register bit has no effect	W
b23	PVDFCLR	Reduction EMB_STAT.PVDF	0: Nothing 1: Clear EMB_STAT.PVDF to 0 Reading this register bit has no effect	W
b22	LOCKUPFCLR	Reduction EMB_STAT.LOCKUPFCLR	0: Nothing 1: Clear EMB_STAT.LOCKUPFCLR to 0 Reading this register bit has no effect	W
b21	RAMPYERRFCLR	Reduction EMB_STAT.RAMPYERRF	0: Nothing 1: Clear EMB_STAT. RAMPYERRF to 0 Reading this register bit has no effect	W
b20	SRAMECCERRFCLR	Reduction EMB_STAT.SRAMECCERRF	0: Nothing 1: Clear EMB_STAT.SRAMECCERRF to 0 Reading this register bit has no effect	W
b19	OSCFCLR	Reduction EMB_STAT.OSCF	0: Nothing 1: Clear EMB_STAT.OSCF to 0 Reading this register bit has no effect	W
b18~b13	Reserved	-	Read “0” when reading, write“0” when writing	W
b12	PORTINFCLR5	Reduction EMB_STAT.PORTINF5	0: Nothing 1: When EMB_STAT.PORTINST5=0, clear EMB_STAT.PORTINF5 to 0 Reading this register bit has no effect	W
b11	PORTINFCLR4	Reduction EMB_STAT.PORTINF4	0: Nothing 1: When EMB_STAT.PORTINST4=0, clear EMB_STAT.PORTINF4 to 0 Reading this register bit has no effect	W
b10	PORTINFCLR3	Reduction EMB_STAT.PORTINF3	0: Nothing 1: When EMB_STAT.PORTINST3=0, clear EMB_STAT.PORTINF3 to 0 Reading this register bit has no effect	W

b9	PORTINFCLR2	Reduction EMB_STAT.PORTINF2	0: Nothing 1: When EMB_STAT.PORTINST2=0, clear EMB_STAT.PORTINF2 to 0 Reading this register bit has no effect	W
b8	PORTINFCLR1	Reduction EMB_STAT.PORTINF1	0: Nothing 1: When EMB_STAT.PORTINST1=0, clear EMB_STAT.PORTINF1 to 0 Reading this register bit has no effect	W
b7~b4	Res	Reserved Bit	Read “0” when reading, write“0” when writing	W
b3	SYSFCLR	Reduction EMB_STAT.SYSF	0: Nothing 1: When EMB_STAT.SYSST=0, clear EMB_STAT.SYSF to 0 Reading this register bit has no effect	W
b2	CMPFCLR	Reduction EMB_STAT.CMPF	0: Nothing 1: When EMB_STAT.CMPST=0, clear EMB_STAT.CMPF to 0 Reading this register bit has no effect	W
b1	PWMSCLR	Reduction EMB_STAT.PWMSF	0: Nothing 1: When EMB_STAT.PWMST=0, clear EMB_STAT. PWMSF to 0 Reading this register bit has no effect	W
b0	Res	Reserved Bit	Read “0” when reading, write“0” when writing	W

21.3.10 EMB Interrupt License Register (EMB _ INTEN)

Address: 0x4003 8614, 0x4003 8634, 0x4003 8654, 0x4003 8674, 0x4003 8694, 0x4003 86B04

0x4003 86D4, 0x4003 86F4, 0x4003 8714

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res			PORTINTEN5	PORTINTEN4	PORTINTEN3	PORTINTEN2	PORTINTEN1	Res				SYSINTEN	CMPINTEN	PWMSINTEN	Res

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b13	Res	Reserved Bit	Read “0” when reading, write“0” when writing	RW
b12	PORTININTEN5	Port 5 input control interrupt enable	0: Port 5 input control does not generate interrupt 1: Port 5 input control generates interrupt	RW
b11	PORTININTEN4	Port 4 input control interrupt enable	0: Port 4 input control does not generate interrupt 1: Port 4 input control generates interrupt	RW
b10	PORTININTEN3	Port 3 input control interrupt enable	0: Port 3 input control does not generate interrupt 1: Port 3 input control generates interrupt	RW
b9	PORTININTEN2	Port 2 input control interrupt enable	0: Port 2 input control does not generate interrupt 1: Port 2 input control generates interrupt	RW
b8	PORTININTEN1	Port 1 input control interrupt enable	0: Port 1 input control does not generate interrupt 1: Port 1 input control generates interrupt	RW
b7~b4	Res	Reserved Bit	Read “0” when reading, write“0” when writing	RW
b3	SYSINTEN	System error control interrupt enable	0: System error control does not generate interrupt 1: System error control generates interrupt	RW
b2	CMPINTEN	Voltage comparator comparison result control interrupt enable	0: The voltage comparator comparison result control does not generate an interrupt 1: The voltage comparator comparison result controls the generation of an interrupt	RW
b1	PWMSINTEN	PWM output same phase control interrupt enable	0: PWM output same phase control does not generate interrupt 1: PWM output same phase control generates interrupt	RW
b0	Res	Reserved Bit	Read “0” when reading, write“0” when writing	RW

21.3.11 EMB Control PWM Output Release Mode Selection Register (EMB_RLSSEL)

Address: 0x4003 8618, 0x4003 8638, 0x4003 8658, 0x4003 8678, 0x4003 8698, 0x4003 86B8, 0x4003 86D8, 0x4003 86F8, 0x4003 8718

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res			PORTINRSEL5	PORTINRSEL4	PORTINRSEL3	PORTINRSEL2	PORTINRSEL1	Res				SYSRSEL	CMPRSEL	PWMRSEL	Res

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b13	Res	Reserved Bit	Read “0” when reading, write“0” when writing	RW
b12	PORTINRSEL5	Port input 5 control release mode selection	0: Release PWM output control when EMB_STAT.PORTINF5=0 1: Release PWM output control when EMB_STAT.PORTINST5=0	RW
b11	PORTINRSEL4	Port input 4 control release mode selection	0: Release PWM output control when EMB_STAT.PORTINF4=0 1: Release PWM output control when EMB_STAT.PORTINST4=0	RW
b10	PORTINRSEL3	Port input 3 control release mode selection	0: Release PWM output control when EMB_STAT.PORTINF3=0 1: Release PWM output control when EMB_STAT.PORTINST3=0	RW
b9	PORTINRSEL2	Port input 2 control release mode selection	0: Release PWM output control when EMB_STAT.PORTINF2=0 1: Release PWM output control when EMB_STAT.PORTINST2=0	RW
b8	PORTINRSEL1	Port input 1 control release mode selection	0: Release PWM output control when EMB_STAT.PORTINF1=0 1: Release PWM output control when EMB_STAT.PORTINST1=0	RW
b7~b4	Res	Reserved Bit	Read “0” when reading, write“0” when writing	RW
b3	SYSRSEL	System error control release mode selection	0: Release PWM output control when EMB_STAT.SYSF=0 1: Release PWM output control when EMB_STAT.SYSYST=0	RW
b2	CMPRSEL	Comparator comparison result controls the release mode selection	0: Release PWM output control when EMB_STAT.CMPF=0 1: Release PWM output control when EMB_STAT.CMPST=0	RW
b1	PWMRSEL	PWM output control release mode selection	0: Release PWM output control when EMB_STAT.PWMSF=0 1: Release PWM output control when EMB_STAT.PWMST=0	RW
b0	Res	Reserved Bit	Read “0” when reading, write“0” when writing	RW

22 General Timer (TimerA)

22.1 Overview

General timer A (TimerA) is a timer with 16/32-bit count width and 4-channel PWM output. The timer supports two waveform modes, triangle wave and sawtooth wave, and can generate various PWM waveforms (single-side aligned PWM, double-side symmetrical PWM); supports synchronous start of counter; the Comparison Reference Value Register supports cache function; supports inter-unit cascade counting; supports 2-phase orthogonal encoding counting and 3-phase orthogonal encoding counting. This series of products is equipped with 5 TimerA units, of which unit 1 (TimerA1) is a 32-bit timer; unit 2 (TimerA2) ~ unit 5 (TimerA5) are 16-bit timers; 5 units can achieve up to 20 PWM outputs.

22.2 Main Features

The basic functions and features of TimerA are as Table 22-1 shown.

Table 22-1 Basic functions and features of TimerA

Waveform mode	**Sawtooth wave, triangular wave**
Basic functions	Direction of increments and decrements
	Synchronous start counter
	Benchmark caching function
	32-bit cascade count
	Orthogonal coding count
	4-Channel PWM Output
	Compare Matched Event Output
Interrupt type	Compare match interrupt
	Periodic matching interrupt

22.3Functional Description

22.3.1 TimerA Block Diagram

The basic block diagram of TimerA is as Figure 22-1 shown. In the figure, "< t >" is the unit number, that is, "< t >" is 1 to 5. In this section, when "< t >" is mentioned later, it refers to the unit number and is not repeated

This series of products carries 5 units of TimerA, among which the bus clock and count clock of units 1 to 4 are both PCLK0 (the PCLK shown in the block diagram refers to PCLK0 at this time); the bus clock and count clock of unit 5 are PCLK1 (the PCLK shown in the block diagram refers to PCLK1 at this time). The counting frequency division clock source of each unit and the sampling reference clock source of the port digital filter are the same as its bus clock and counting clock.

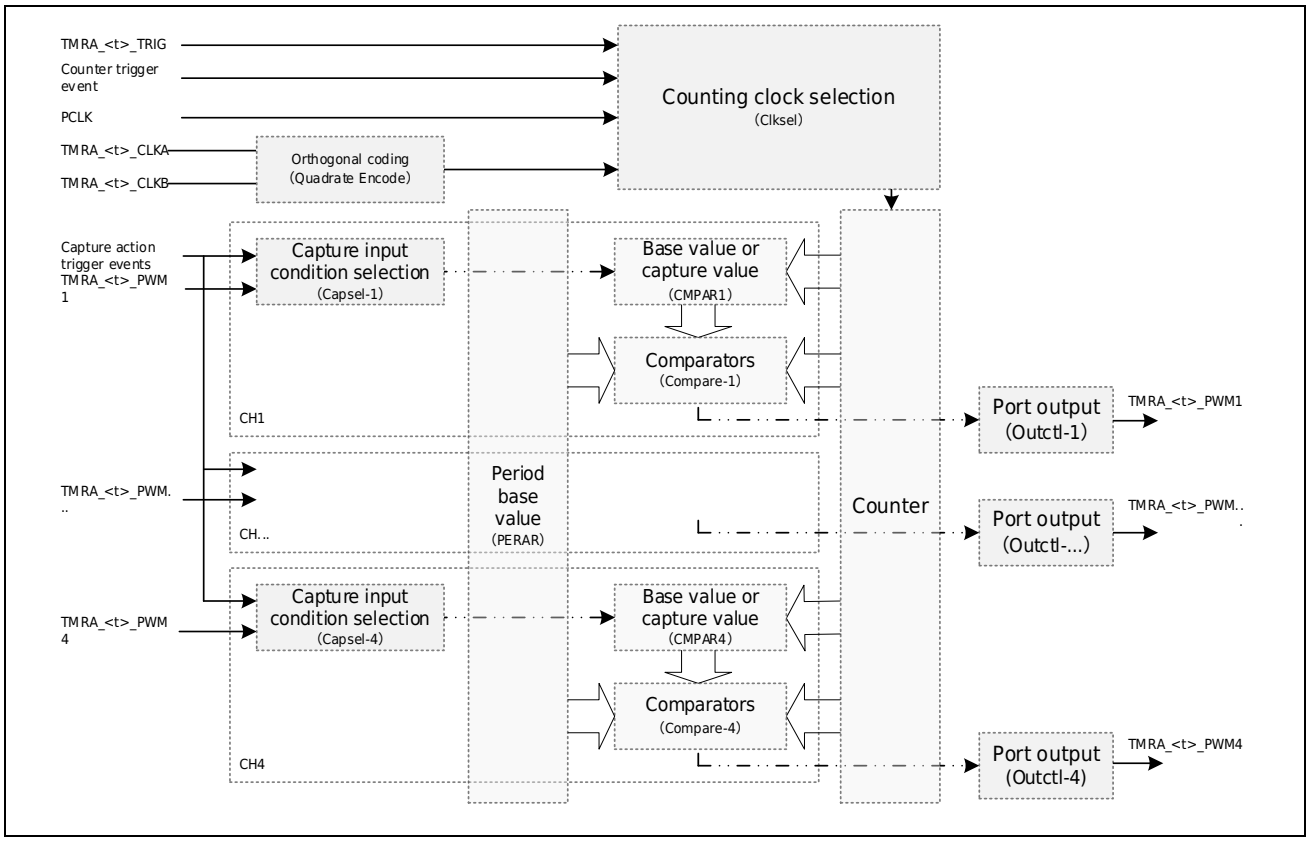


Figure 22-1 TimerA Basic Block Diagram

Table 22-2 shown is the list of input and output ports of TimerA.

Table 22-2 TimerA Port List

Port Name	Direction	Function
TMRA_<t>_PWMm	Input output	Capture input event port or PWM output port (m = 1 ~ 4)
TMRA_<t>_CLKA	Input	Orthogonal Coded Count Event Input Port
TMRA_<t>_CLKB		
TMRA_<t>_TRIG	Input	Hardware Trigger Startup, Stop, Zero Event Input Port

22.3.2 Symmetrical Unit

Unit 1 and unit 2 of TimerA are symmetrical units to each other, unit 3 and unit 4 are symmetrical units to each other, and so on. Symmetrical units can be linked to achieve cascade counting, synchronous start, etc., as described in the following chapters.

22.3.3 Waveform Mode

TimerA has 2 basic counting wave modes: Serrated wave mode and triangular wave mode. The basic waveforms of the two waveform modes are shown in Figure 22-2 and Figure 22-3.

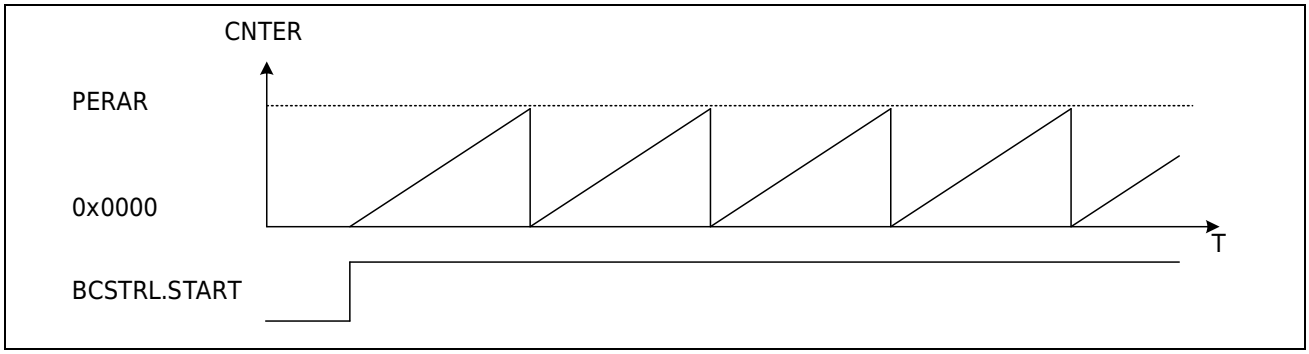


Figure 22-2 Sawtooth Waveform (Counting Up)

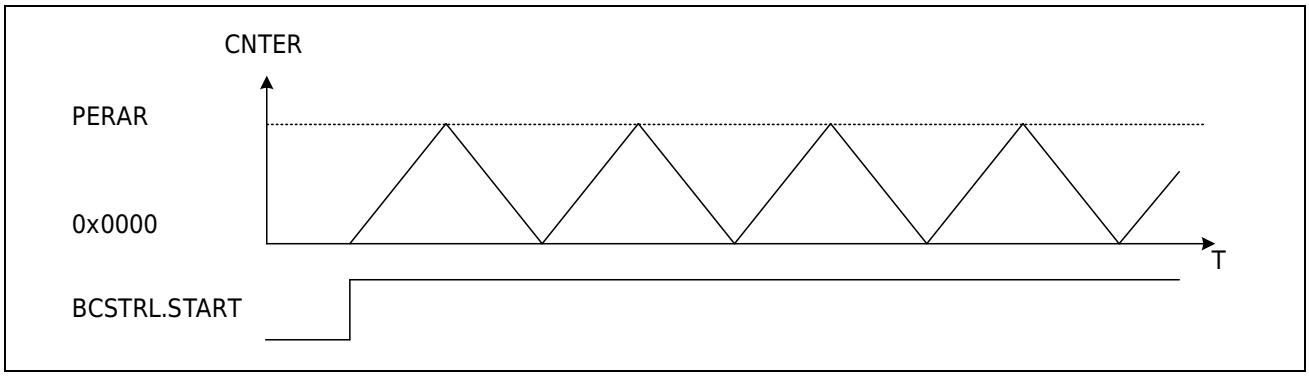


Figure 22-3 Triangle Waveform

22.3.4 Timer Selection

TimerA's counting clock has the following options:

a) 1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024 division of PCLK (BCSTRL.CKDIV[3:0] setting)

b) TMRA_<t>_TRIG port event input (HCUPR[9:8] or HCDOR[9:8] setting)

c) Internal counter trigger event input (HCUPR [10] or HCDOR [10] settings)

d) Count overflow or count underflow event input of the symmetric unit (set by HCUPR[12:11] or HCDOR[12:11])

e) TMRA_<t>_CLKA, TMRA_<t>_CLKB port quadrature encoding input (HCUPR[7:0] or HCDOR[7:0] setting)

The counting timer is a software counting mode, and the counting timer is b, c, d, e hardware counting mode. When the counting clock is selected as d, it is mostly used in the revolution counting mode of three-phase orthogonal encoding counting (see [Position Overflow Counting] and [Mixed Counting] sections), and can also be used for cascade counting. The above description shows that the b, c, d and e clocks are independent of each other, can be set to be valid or invalid respectively, and the a clock is automatically invalid when b, c, d and e clocks are selected.

22.3.5 Comparison Output

Each TimerA unit contains 4 channels of comparison output (TMRA_<t>_PWMm), which can output the specified level when the count value matches the comparison reference value. The TMRA_CMPARm registers correspond to the count comparison reference values of the TMRA_<t>_PWMm output ports. When the timer count value is equal to TMRA_CMPARm, the TMRA_<t>_PWMm port outputs the specified level (m=1~4).

The count start level, count stop level, count compare match level, count cycle match level, etc. of the TMRA_<t>_PWMm port can be controlled by setting the STAC, STPC, CMPC, PERC, and FORC bits of the port control register (PCONRm) (m=1~4). Figure 22-4 is an example of the comparison output operation of Unit 1.

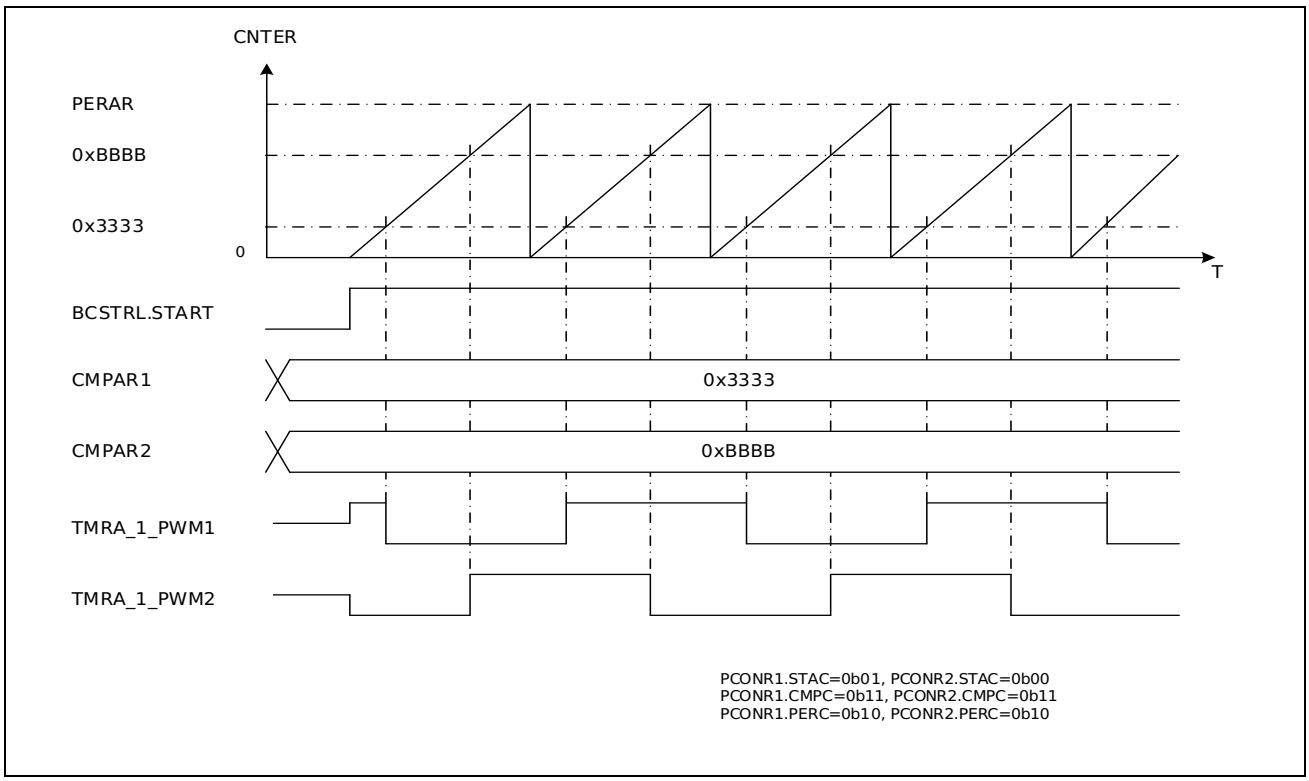


Figure 22-4 Compare Output Action

22.3.6 Capture Input

Each PWM output channel of each Timer A unit has a capture input function to store the captured count values. Set the CCONR. CAPMD bit of the capture control register (CCONRm) to 1 to make the capture input function valid. When the corresponding capture input condition is selected and is valid, the current count value is saved in the corresponding register (CMPARm) (n = 1 ~ 4).

The capture input condition can select the internal capture action trigger event (through TMRA_TRGSELx, x=0~3) register selection, refer to the [TimerA Hardware Trigger Event Selection Register (TMRA_TRGSELx, x = 0 ~ 3)] section for details), TMRA_<t>_PWMm port input, etc. The specific condition selection can be set through the HICP bit of the capture control register (CCONRm) (m=1~4).

The internal trigger event selection register is shared by all five units of TimerA. The specific allocation is shown in the following table.

Table 22-3 Internal Trigger Event TRGSEL Selection Relationship Correspondence Table

TimerA Unit	Event Classification	The corresponding selection register
TimerA1	Counter trigger event: TRGSEL selection event indicated by HCONR, HCUPR, HCDOR	TMRA_TRGSEL0
TimerA2		TMRA_TRGSEL1
TimerA3		TMRA_TRGSEL2
TimerA4		TMRA_TRGSEL3
TimerA5		TMRA_TRGSEL0
TimerA1	Capture action trigger events: TRGSEL selection event referred to in CCONR	TMRA_TRGSEL1
TimerA2		TMRA_TRGSEL0
TimerA3		TMRA_TRGSEL3
TimerA4		TMRA_TRGSEL2
TimerA5		TMRA_TRGSEL1

Figure 22-5 is an example of capture input action.

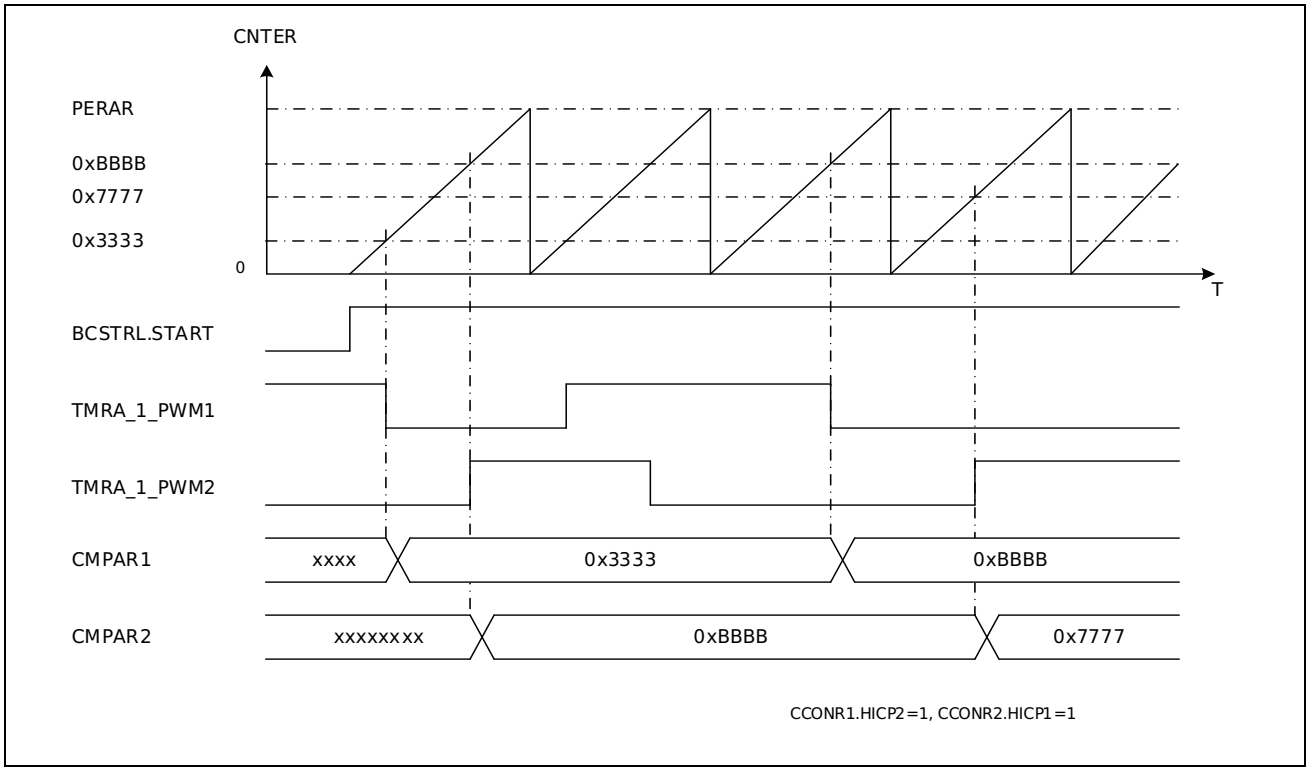


Figure 22-5 Capturing Input Actions

22.3.7 Synchronous Startup

The symmetric units of TimerA can be started synchronously by software or hardware, and the even units can be started synchronously by the odd units.

When the BCSTRL.SYNST bit of the even-numbered unit in the symmetrical unit is set to 1, the synchronous start function of the even-numbered unit and the odd-numbered unit valid. At this time, if the software sets the BCSTRL.START bit of the odd unit to 1, the counter of the synchronized even unit starts software synchronous counting; if the hardware sets any bit in HCONR.HSTA1~0 of the odd unit to 1, and the corresponding hardware event occurs, the counter of the synchronized even unit starts hardware synchronous counting. When the hardware synchronous count start function is selected, the corresponding bits of HCONR.HSTA1~0 of the synchronized even-numbered units must also be set to valid.

Figure 22-6 is an example of software synchronous startup when BCSTRL.SYNST = 1 in unit 2 is set.

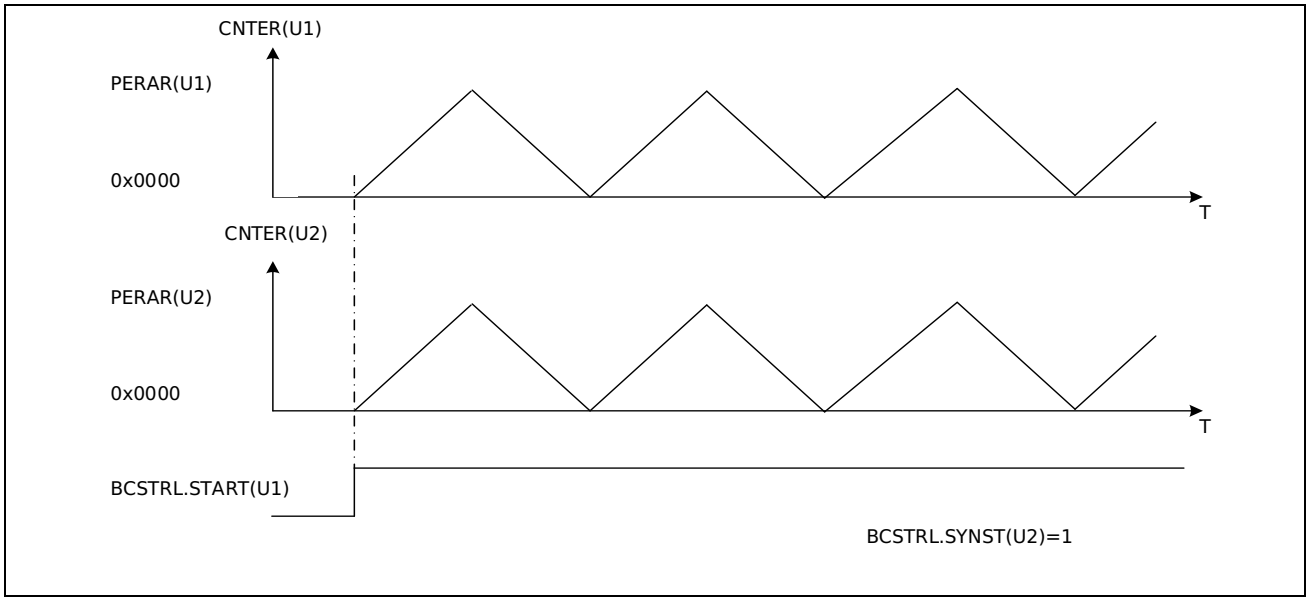


Figure 22-6 Software Synchronization Action

In addition, the product is also equipped with a count synchronization enable register (TMR_SYNENR) to synchronously start all the timers (including Timer0, Timer4, Timer6, TimerA) carried by the product. for application method and Register Description, please refer to the [Universal Timer (Timer0)] section.

22.3.8 Digital Filtering

The TMRA_<t>_CLKA, TMRA_<t>_CLKB, TMRA_<t>_TRIG, and TMRA_<t>_PWMm (capture input function) port inputs of each unit have digital filtering functions. The enabling of the filtering function of each port and the selection of the filtering clock can be achieved by setting the corresponding bits of the filter control register (FCONR) and the capture control register (CCONRm) (m=1~4).

When the filtered sampling reference clock is sampled to the same level three times on the port, the level is transferred to the module as an effective level. A level less than three times consistent will be filtered out as external interference and not transmitted to the module. Figure 22-7 shows an example of TMRA_1_CLKA port filtering operation.

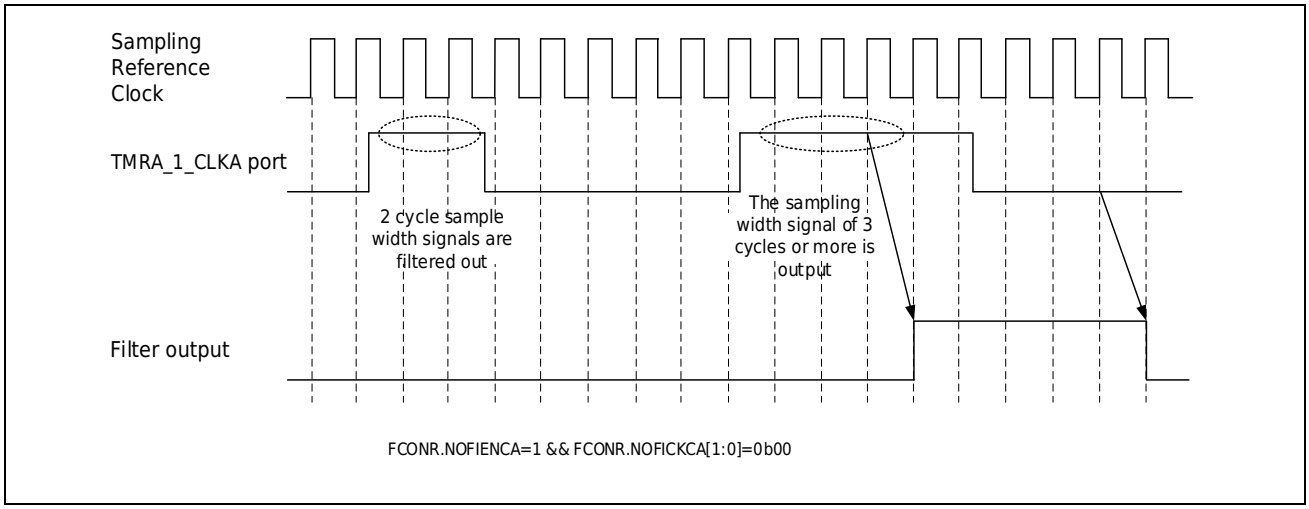


Figure 22-7 Filtering function of the clock input port

22.3.9 Cache Function

The four compare value registers (CMPARm) of TimerA can be used in pairs to implement the cache function (m=1~4). That is, CMPAR2 is used as the cache comparison value of CMPAR1, and CMPAR4 is used as the cache comparison value of CMPAR3. The cache control register (BCONRm) realizes the control of four sets of cache functions respectively (m=1~2).

When the BEN bit of the cache control register (BCONRm) is set, the cache function becomes valid (m=1~2). When the counter reaches a certain point in time, a cache transfer occurs (CMPAR2/4->CMPAR1/3). The "specific point in time" has the following circumstances:

a) In hardware counting mode, when the counter overflows (HCUPR[12:0]!=0) or underflows (HCDOR[12:0]!=0) or hardware clears the counter

b) In sawtooth counting mode (BCSTRL.MODE=0), the counter counts to the overflow point (BCSTRL.DIR=1) or underflow point (BCSTRL.DIR=0) or the hardware clears the counter.

c) In triangle wave counting mode (BCSTRL.MODE=1), count to the peak point (BCSTRL.DIR=1 && BCONRm.BSE0=1) (m=1~2)

d) In triangle wave counting mode (BCSTRL.MODE=1), count to valley point (BCSTRL.DIR=0 && BCONRm.BSE1=1) (m=1~2)

When the BSEN bit of the cache control register (BCONRm) is set, the cache special function becomes valid (m=1~2). When the CNTER of the symmetric unit is read, a cache transfer occurs in the CMPAR of this unit (CMPAR2/4->CMPAR1/3).

Figure 22-9 below is a schematic diagram of buffer transmission and special buffertransmission in sawtooth wave mode.

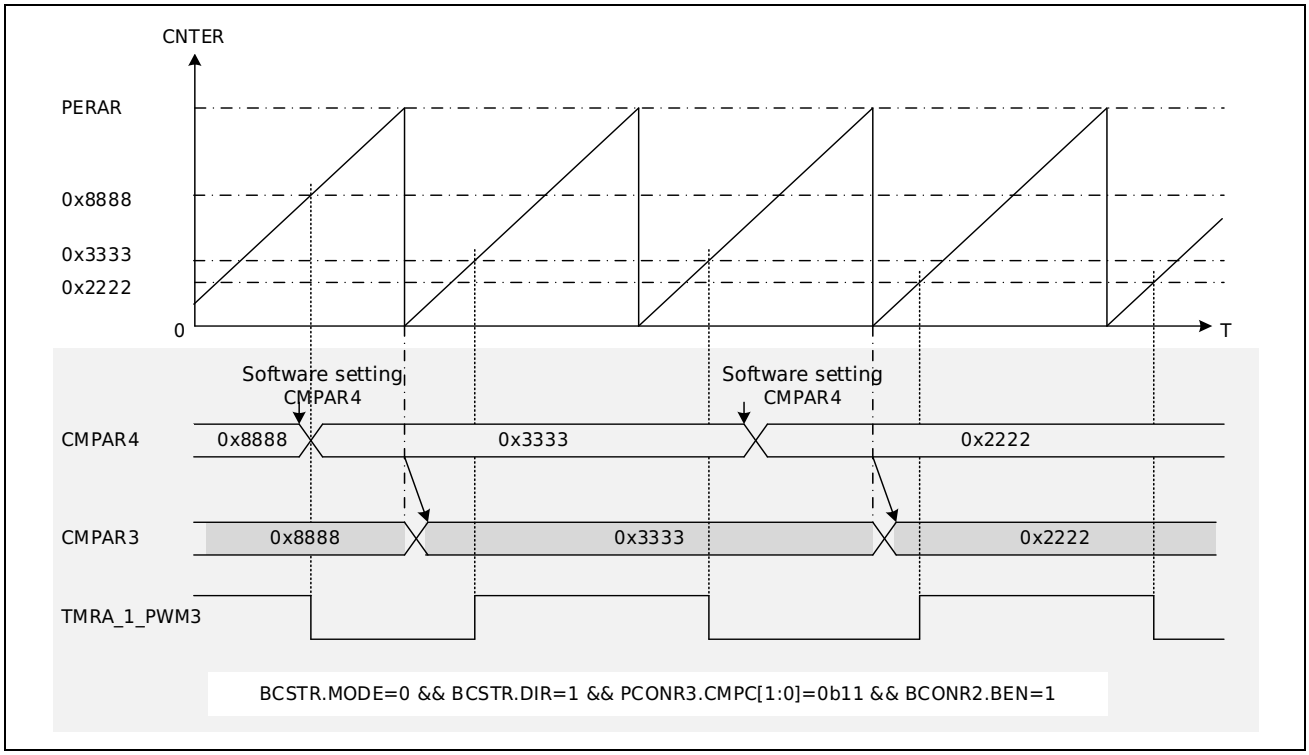


Figure 22-8 Cache Action in Sawtooth Wave Mode

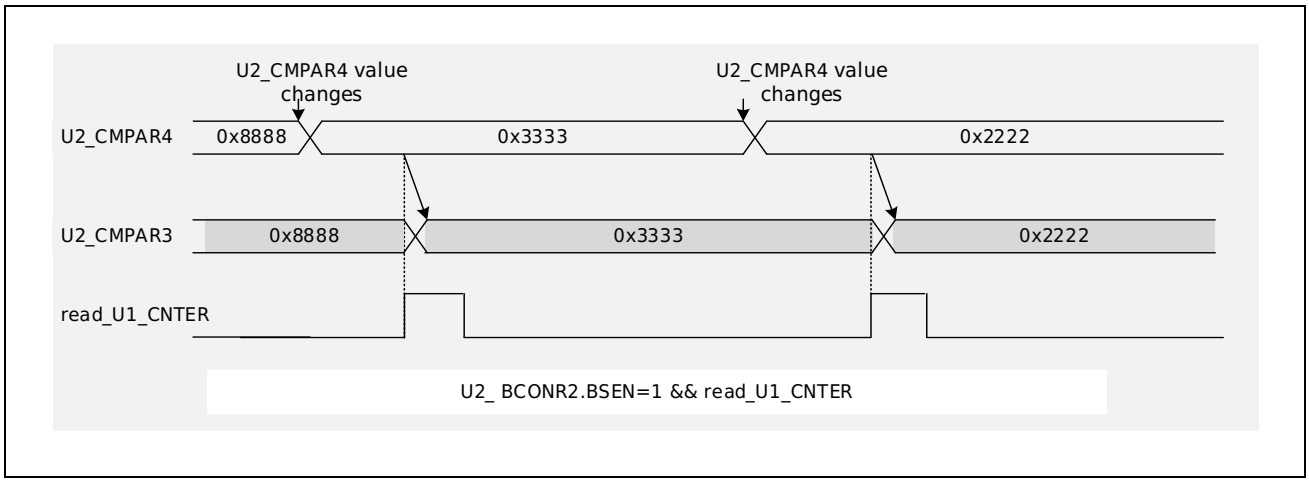


Figure 22-9 Cache and Special Cache Actions in Sawtooth Mode

22.3.10 Cascade Count

In the Count timer selection chapter, when timer is d, the count clock for this cell is selected as the count overflow (count overflow or count underflow) event for the symmetric cell, at which time the two cell levels combine and implement 32-bit counter. In the cascade counting, the CNTER of the symmetric unit is the low 16-bit counter and the CNTER of the unit is the high 16-bit counter.

For example, in the triangle wave upward counting mode (BCSTRL.MODE=0, BCSTRL.DIR=1), set the counting clock of unit 1 to PCLK, set the counting clock source of unit 2 to the counting overflow event of unit 1 (TMRA_HCUPR.HCUP11=1 of unit 2), and start counting of units 2 and 1 (start unit

2 first, then start unit 1) to realize cascade counting. Unit 1 has a low 16-bit counter at CNTER bit and Unit 2 has a high 16-bit counter at CNTER bit. As Figure 22-10 shown is a schematic diagram of unit 1, 2 cascade counting.

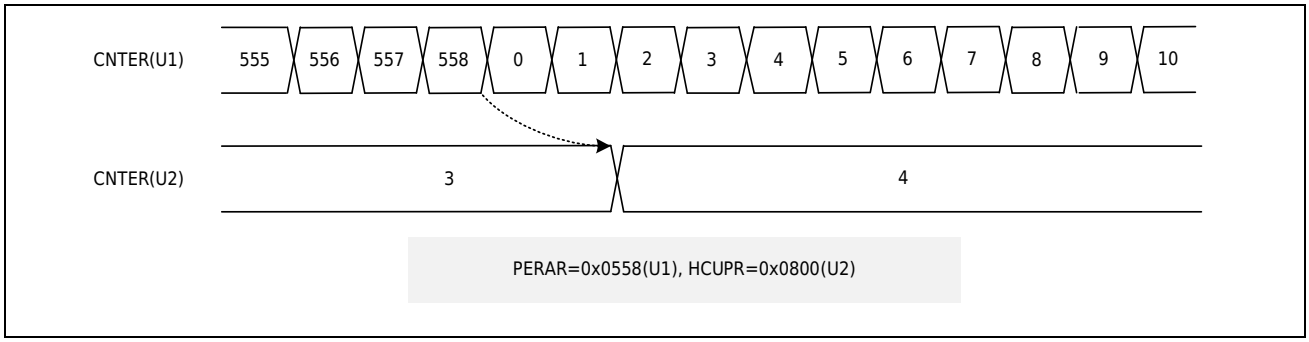


Figure 22-10 32-Bit Cascade Counting Action

22.3.11 PWM Output

22.3.11.1 Unidirectional Aligned PWM Output

In sawtooth wave counting mode, each channel in a cell can be set by various port control settings to realize one-sided aligned PWM output (aligned at counting cycle point). Setting, for example, to flip when a baseline comparison matches (PCONR.CMPC = 11) and to flip when a cycle baseline comparison matches (PCONR.PERC=0b11) enables a single-sided aligned PWM output to be generated in one cycle.

The Figure 22-11 shows an example of a single-side aligned PWM output waveform in sawtooth wave mode.

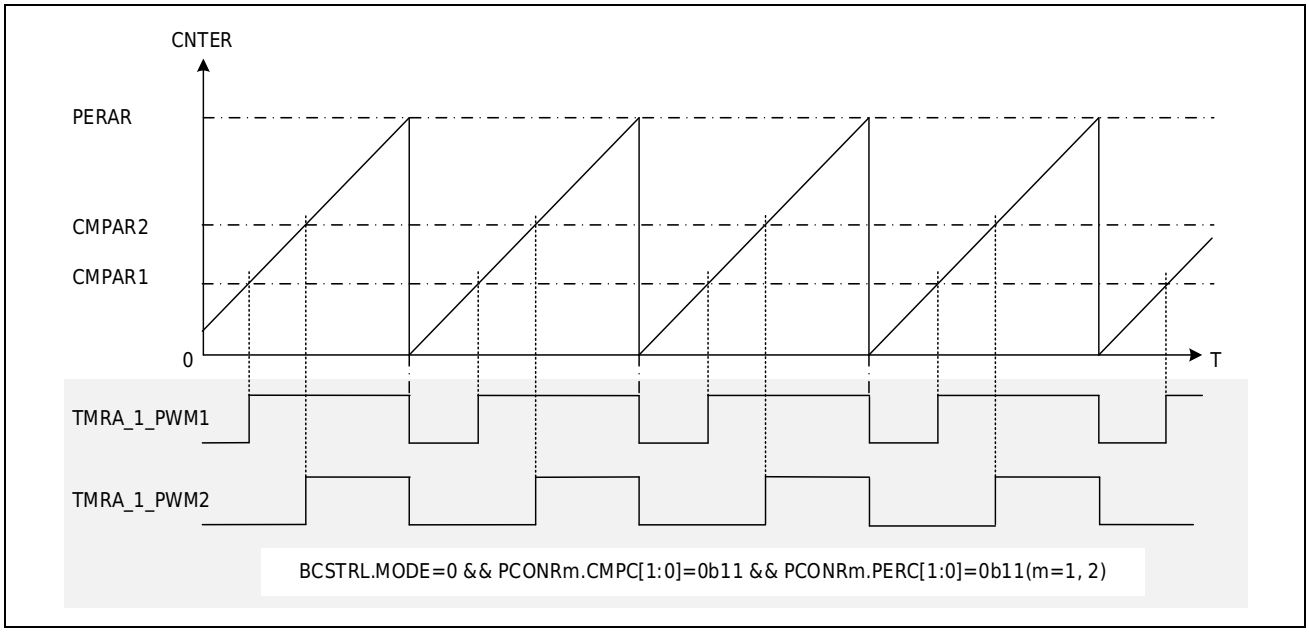


Figure 22-11 Single-edge Aligned PWM Output Example

22.3.11.2 Bilateral Symmetric PWM Output

In the triangulation mode, each channel in a cell can be set by various port controls to realize bilateral symmetric PWM output (denoted by counting peak points). Independent PWM output or complementary PWM output can be realized according to the output relationship between channels. Figure 22-12 The figure shows an example of the bilaterally symmetrical PWM output waveform of channels 1, 2, 3, and 4 in triangle wave mode. Among them, channels 1 and 2 can be used as a pair of complementary PWM outputs.

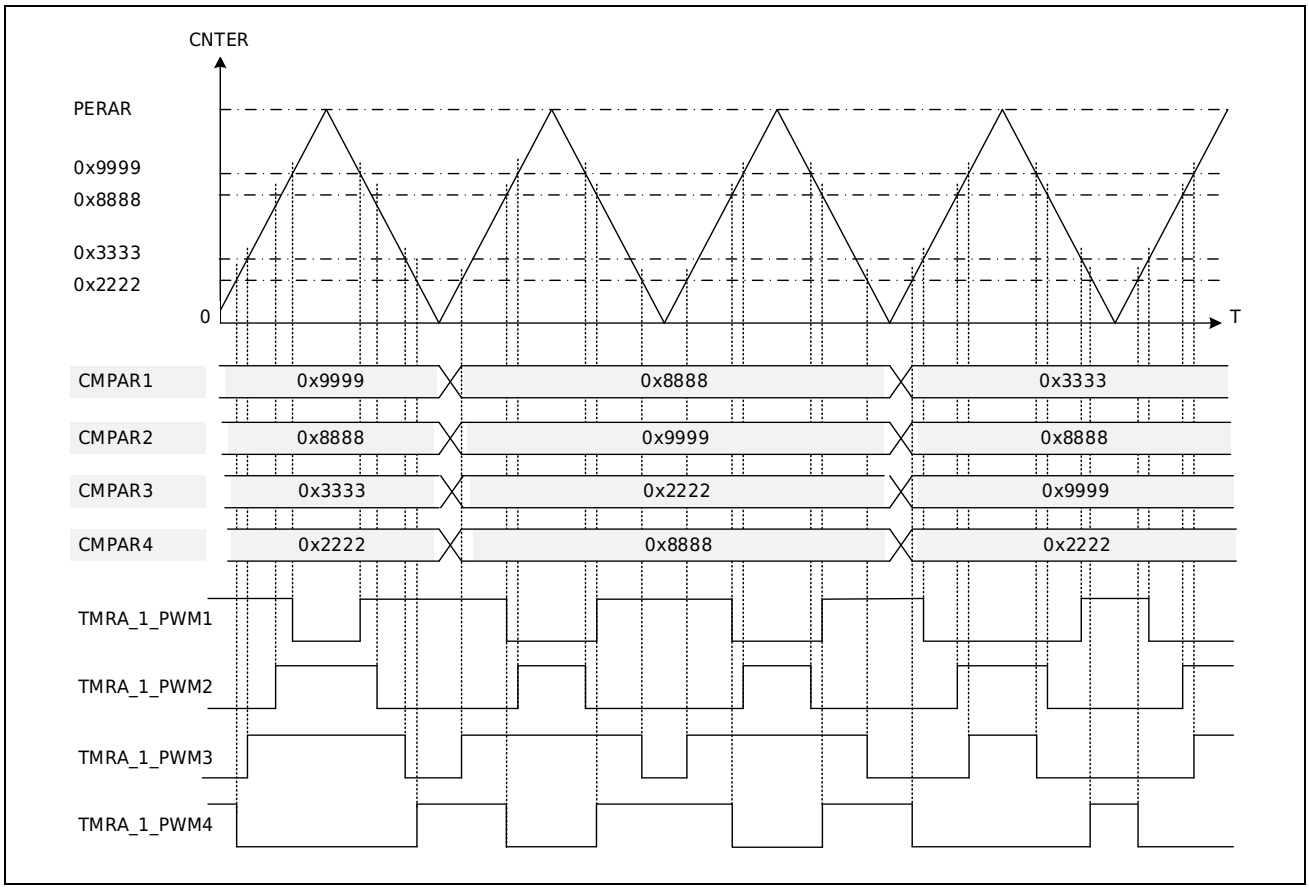


Figure 22-12 Example of Bilateral Symmetrical PWM Output

22.3.12 Orthogonal Coding Count

By considering the TMRA_<t>_CLKA input as AIN input, the TMRA_<t>_CLKB input as BIN input, and the TMRA_<t>_TRIG input as ZIN input, TimerA can implement quadrature encoding counting of three inputs.

The single action of AIN and BIN of each unit can realize the position counting mode; the combined action of AIN, BIN and ZIN of two units can realize the revolution counting mode, and the unit used for position counting is called position counting unit, which is used for revolution counting The unit is called revolution counting unit. In the revolution counting mode, every two units are combined with each other (unit 1, 2 combination; unit 3, 4 combination), and the position counting unit and revolution counting unit in the combination can be specified arbitrarily.

The counting condition enable of AIN and BIN is realized by setting the orthogonal relationship between TMRA_<t>_CLKA and TMRA_<t>_CLKB in the hardware increment event selection register (HCUPR) and the hardware decrement event selection register (HCDOR); the input action of ZIN is realized by setting the clear enable bit of the hardware trigger event selection register (HCONR) of the position counting unit to clear the position timer, and by setting the hardware increment event selection register (HCUPR) of the revolution unit to realize the revolution timer counting.

22.3.12.1 Position Counting Mode

The quadrature encoding position counting mode refers to realizing the basic counting function, phase difference counting function and direction counting function according to the input of AIN and BIN.

Basic Count

The basic counting action is to count according to the input clock of the AIN or BIN port, as shown in Figure 22-13 below.

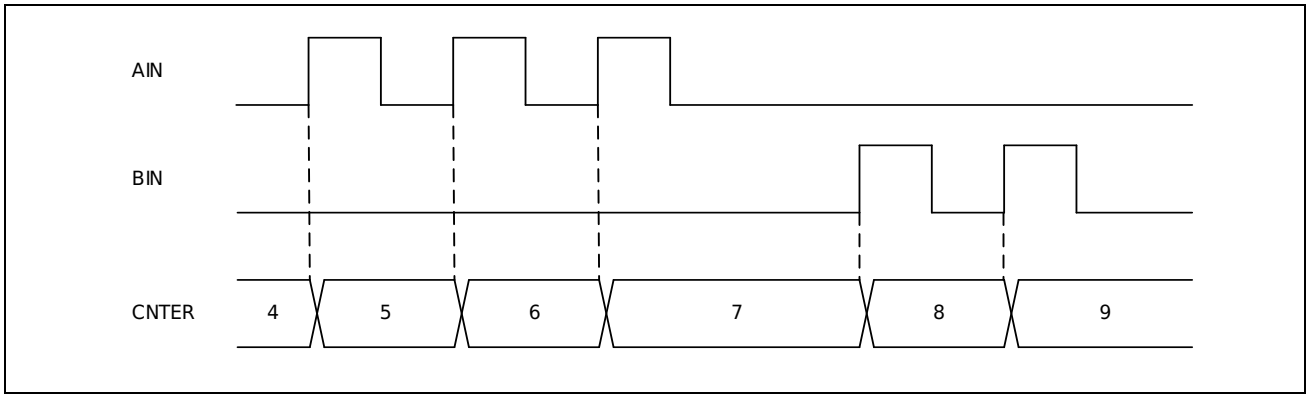


Figure 22-13 Position Mode - Basic Counting

Phase Difference Count

Phase difference counts are counted according to the phase relationship between AIN and BIN. According to different settings, 1-fold counting, 2-fold counting, 4-fold counting, etc. can be realized, as shown in the following Figure 22-14 ~ Figure 22-16.

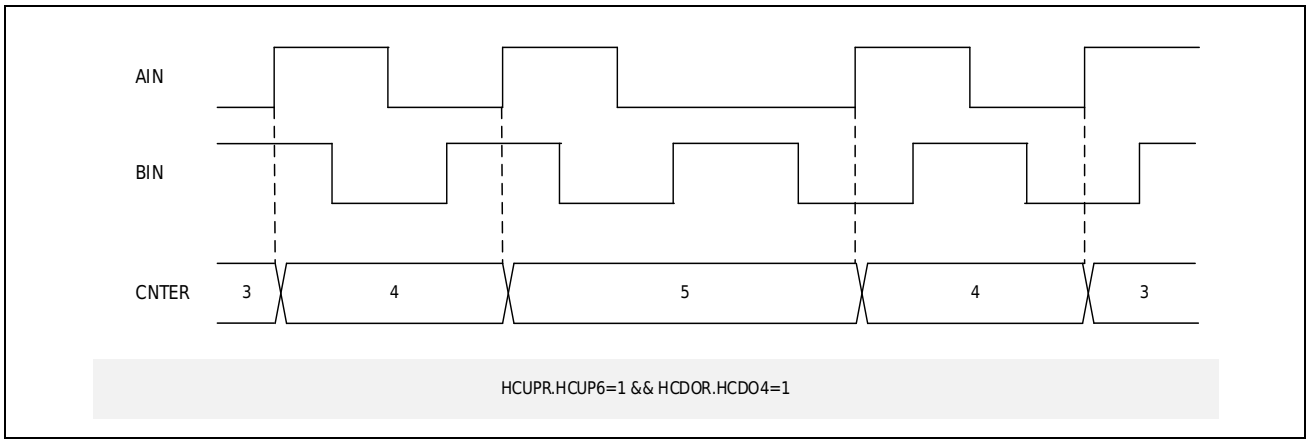


Figure 22-14 Position counting mode - phase difference counting (1 times counting)

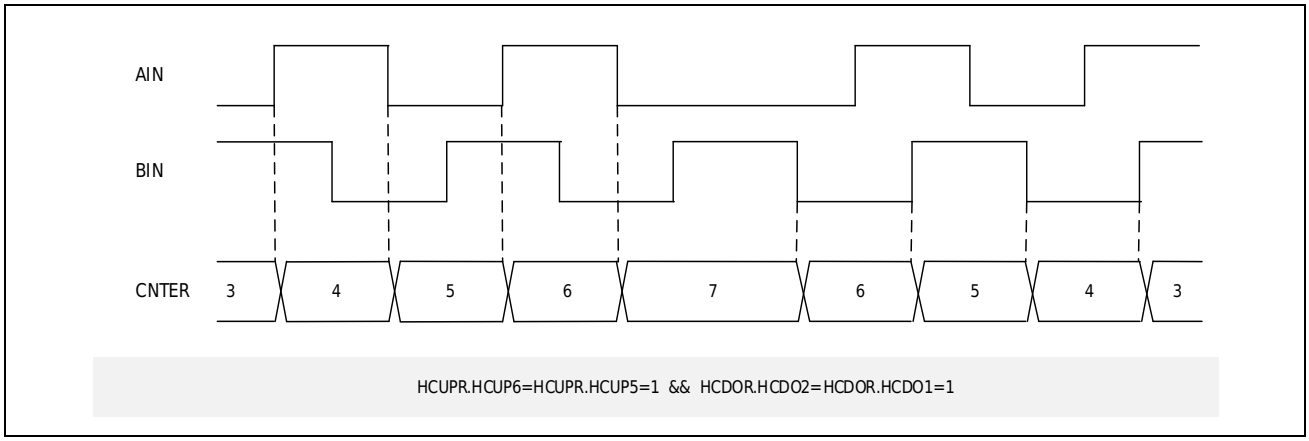


Figure 22-15 Position Counting Mode - Phase Difference Counting (2 Times Counting)

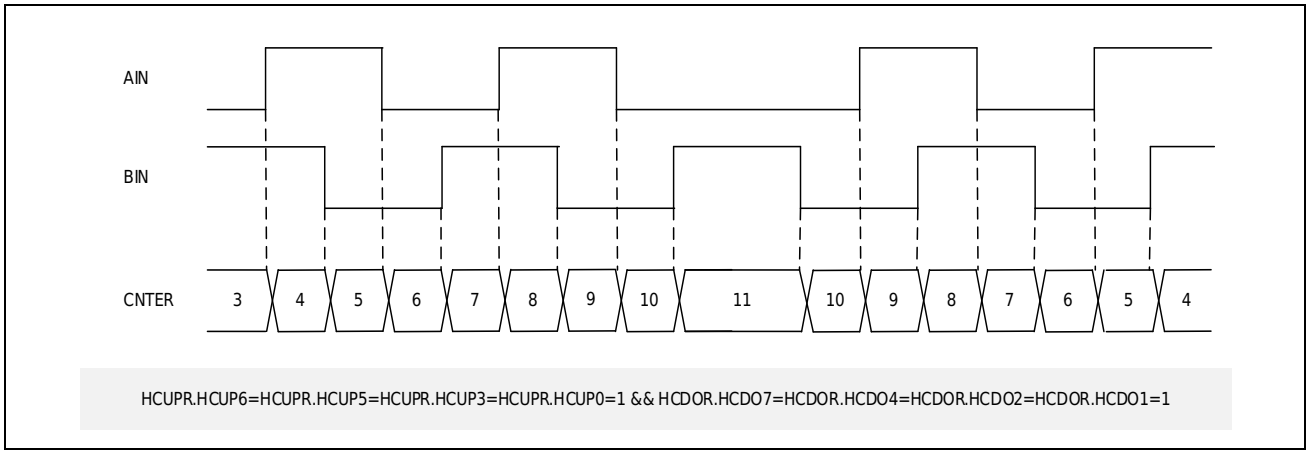


Figure 22-16 Position Counting Mode - Phase Difference Counting (4 Times Counting)

Direction count

Direction counting refers to setting the input state of AIN as direction control, and using the input of BIN as clock counting, as shown in the Figure 22-17 below.

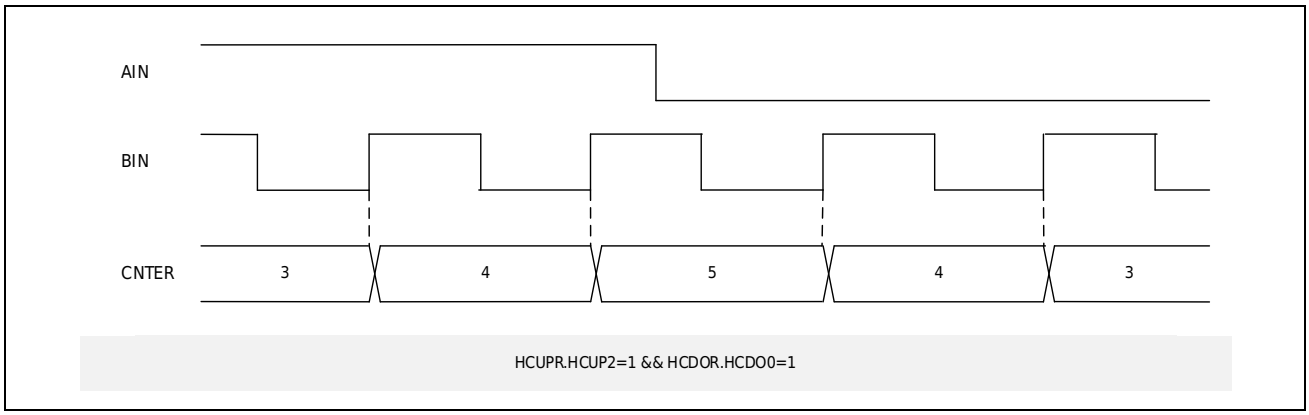


Figure 22-17 Position Counting Mode - Direction Counting

22.3.12.2 Revolution Counting Mode

Orthogonal encoding revolution counting mode refers to the addition of ZIN input events on the basis of AIN and BIN counting to realize the judgment of the number of revolutions, etc. In the revolution counting mode, according to the counting method of the revolution timer, the Z-phase counting function, the position overflow counting function and the mixed counting function can be realized.

Z Phase Count

Z-phase counting refers to the counting action that the revolution counting unit counts according to the input of ZIN, and the position counting unit is cleared at the same time. As shown in Figure 22-18 below.

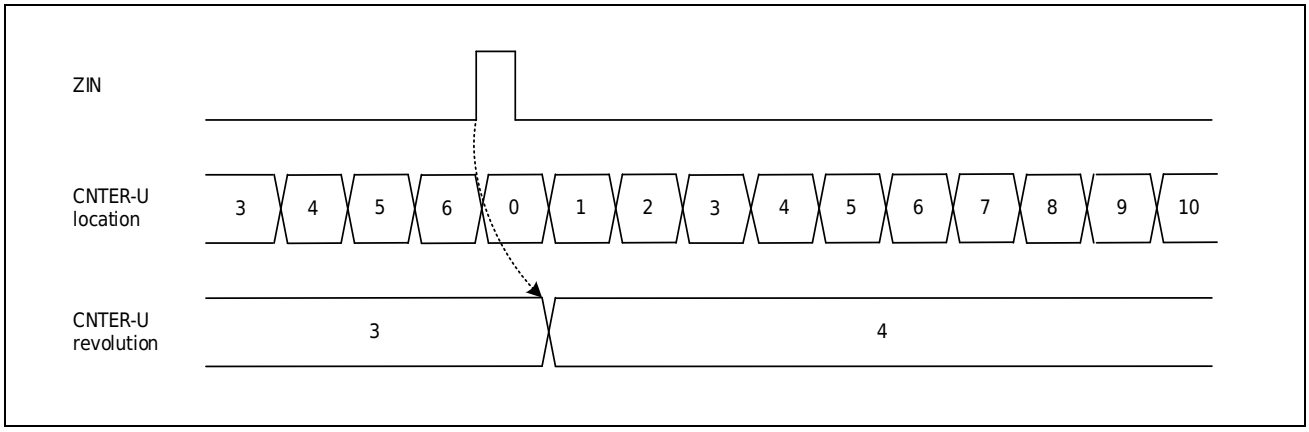


Figure 22-18 Revolution Counting Mode - Z Phase Counting

Position overflow count

Position overflow counting means that when the counting of the position counting unit overflows or underflows, an overflow event is generated, thereby triggering the timer of the revolution counting unit to count once (in this counting mode, the input of ZIN does not perform the counting action of the revolution counting unit and the clearing action of the position counting unit).

The increment (decrement) event bit12~11 of the hardware increment (decrement) event selection register (HCUPR or HCDOR) of the revolution counting unit is enabled, and the overflow event of the position counting unit can trigger the revolution counting unit to achieve a count. As shown in Figure 22-19 below.

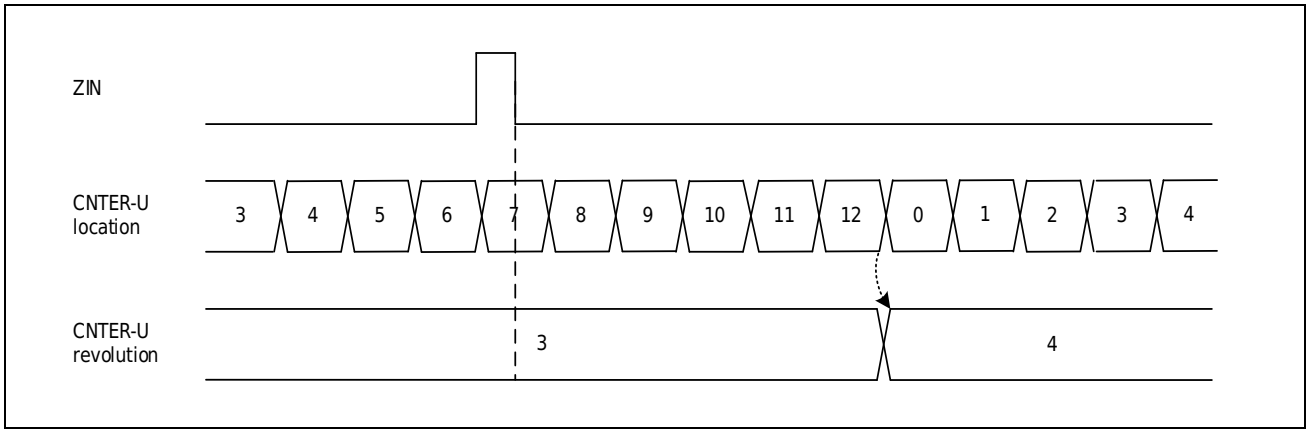


Figure 22-19 Revolution Counting Mode - Position Overflow Counting

Mixed Count

Mixed counting refers to the counting action that combines the above two counting methods of Z-phase counting and position overflow counting, and its realization method is also a combination of the above two counting methods. As shown in Figure 22-20 below.

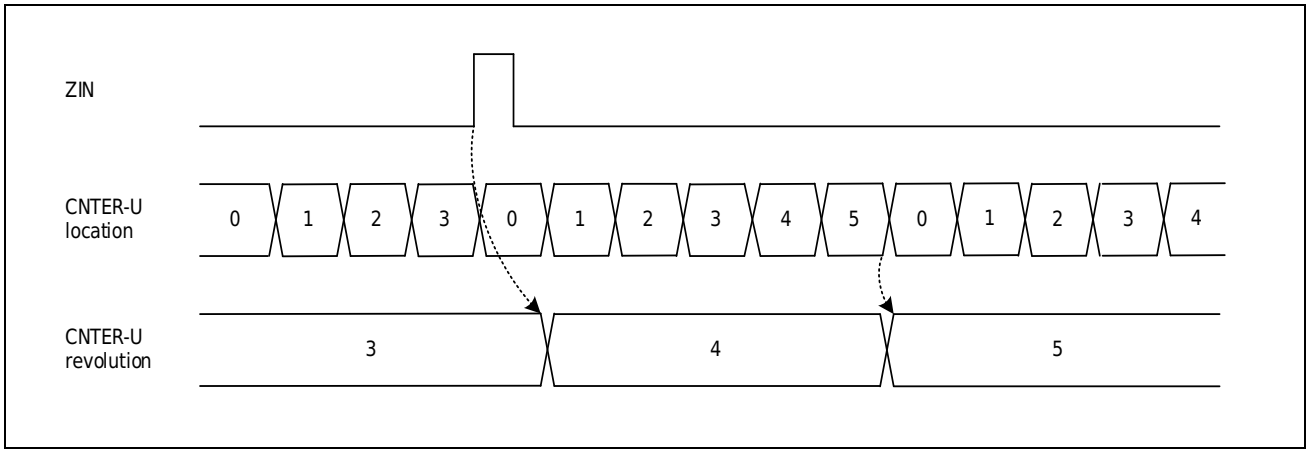


Figure 22-20 Revolution Counting Mode - Mixed Counting

22.4Interrupt and Events

TimerA contains three interrupt outputs and three event outputs: One comparison match interrupt and event, two period match interrupt and event.

22.4.1 Compare Matched Interrupt and Events

When a match occurs between the comparison reference value register (CMPARm) and the count value, the corresponding bit (STFLR.CMPFm) in the status flag register (STFLR) will be set to 1. At this time, if the corresponding bit (ICONR.ITENm) of the interrupt control register (ICONR) is set to 1, the corresponding interrupt request (TMRA_<t>_CMP) will be triggered; if the corresponding bit (ECONR.ETENm) of the event control register (ECONR) is set to 1, the corresponding event request (TMRA_<t>_CMP) will be triggered (m=1~4).

When the capture input valid condition selected by the capture control register (CCONRm) occurs, the capture input action occurs. At this time, if the corresponding bit (ICONR.ITENm) of the interrupt control register (ICONR) is set to 1, the corresponding interrupt request (TMRA_<t>_CMP) is triggered; if the corresponding bit (ECONR.ETENm) of the event control register (ECONR) is set to 1, the corresponding event request (TMRA_<t>_CMP) will be triggered (m=1~4).

The compare match interrupts and compare match events of the 8 reference values within each unit are not output independently. The compare match interrupts are aggregated into one interrupt output to the interrupt module through "OR logic" (see the [Interrupt controller (INTC)] section), and the compare match events are aggregated into one event output through "OR logic" to select and trigger other modules..

22.4.2 Periodic Matching Interrupt and Events

The OVFF or UDFF bits of the control state register (BCSTRH) are set to 1. At this time, if the BCSTRH.ITENOVF or BCSTRH.ITENUDF bit is set to 1 to enable interrupts, the period match interrupt (TMRA_<t>_OVF and TMRA_<t>_UDF) can be triggered at the corresponding cycle point to be output to the interrupt module (INTC); the period match event has no corresponding enable bit control, and the period match event (TMRA_<t>_OVF and TMRA_<t>_UDF) output is triggered at the corresponding counting cycle point to select triggering other modules.

22.5Register Description

Table 22-4 TimerA Base Address

Name	Base Address	Description
TimerA1	0x4003 A000	TimerA1 base address
TimerA2	0x4003 A400	TimerA2 base address
TimerA3	0x4003 A800	TimerA3 base address
TimerA4	0x4003 AC00	TimerA4 base address
TimerA5	0x4002 6000	TimerA5 base address

Table 22-5 TimerA Register List

Register	Description	Offset Address	Bit Width	Reset Value
TMRA_CNTER	Universal count register	0x0000	32/16	0x0000 0000/ 0x0000
TMRA_PERAR	Cycle reference register	0x0004	32/16	0xFFFF FFFF/ 0xFFFF
TMRA_CMPAR1	Compare Base Value Register 1	0x0040	32/16	0xFFFF FFFF/ 0xFFFF
TMRA_CMPAR2	Compare Base Value Register 2	0x0044	32/16	0xFFFF FFFF/ 0xFFFF
TMRA_CMPAR3	Compare Base Value Register 3	0x0048	32/16	0xFFFF FFFF/ 0xFFFF
TMRA_CMPAR4	Compare Base Value Register 4	0x004C	32/16	0xFFFF FFFF/ 0xFFFF
TMRA_BCSTRL	Control status register L	0x0080	8	0x02
TMRA_BCSTRH	Control Status Register H	0x0081	8	0x00
TMRA_ICONR	Interrupt control register	0x0090	16	0x0000
TMRA_ECONR	Event control register	0x0094	16	0x0000
TMRA_FCONR	Filter control register	0x0098	16	0x0000
TMRA_STFLR	Status flag register	0x009C	16	0x0000
TMRA_BCONR1	Cache Control Register 1	0x00C0	16	0x0000
TMRA_BCONR2	Cache Control Register 2	0x00C8	16	0x0000
TMRA_BCONR3	Cache Control Register 3	0x00D0	16	0x0000
TMRA_BCONR4	Cache Control Register 4	0x00D8	16	0x0000
TMRA_CCONR1	Capture Control Register 1	0x0100	16	0x0000
TMRA_CCONR2	Capture Control Register 2	0x0104	16	0x0000
TMRA_CCONR3	Capture Control Register 3	0x0108	16	0x0000
TMRA_CCONR4	Capture Control Register 4	0x010C	16	0x0000
TMRA_PCONR1	Port Control Register 1	0x0140	16	0x0000
TMRA_PCONR2	Port Control Register 2	0x0144	16	0x0000
TMRA_PCONR3	Port Control Register 3	0x0148	16	0x0000
TMRA_PCONR4	Port Control Register 4	0x014C	16	0x0000
TMRA_HCONR	Hardware trigger event selection register	0x0084	16	0x0000
TMRA_HCUPR	Hardware Adder Event	0x0088	16	0x0000

Register	Description	Offset Address	Bit Width	Reset Value
	Selection Register			
TMRA_HCDOR	Hardware Decrease Event Selection Register	0x008C	16	0x0000

Note:

– *The bit widths of the Universal count value register, Period Reference Value Register, and Comparison Reference Value Register of unit 1 are all 32 bits, and the reset values are different from those of units 2 to 5. The reset value of the Universal count value register is 0x0000 0000, and the reset value of the Period Reference Value Register and the Comparison Reference Value Register is 0xFFFF FFFF.*

22.5.1 Generic Count Value Register (TMRA _ CNTER)

Unit 1: Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
CNT[31:16]															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
CNT[15:0]															

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b0	CNT[31:0]	Count value	Current timer count **Note:** 1. Can only be written when the timer is stopped; 2. In sawtooth wave upward counting mode, the initial value of CNT is set to 0; in sawtooth wave downward counting mode, the initial value of CNT is set to pear; 3. The initial value of the triangle wave mode count CNT is set to 1;	RW

Unit 2~5: Reset value: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
CNT[15:0]															

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b15~b0	CNT[15:0]	Count value	Current timer count **Note:** 1. Can only be written when the timer is stopped; 2. In sawtooth wave upward counting mode, the initial value of CNT is set to 0; in sawtooth wave downward counting mode, the initial value of CNT is set to pear; 3. The initial value of the triangle wave mode count CNT is set to 1;	RW

22.5.2 Periodic Reference Value Register (TMRA _ PERAR)

Unit 1: Reset value: 0xFFFF FFFF

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
PER[31:16]															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
PER[15:0]															

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b0	PER[31:0]	Count cycle value	Set the count cycle value for each round	RW

Unit 2~5: Reset value: 0xFFFF

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
PER[15:0]															

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b15~b0	PER[15:0]	Count cycle value	Set the count cycle value for each round	RW

22.5.3 Comparison Reference Value Register (TMRA_CMPARm, m=1~4)

Unit 1: Reset value: 0xFFFF FFFF

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
CMP[31:16]															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
CMP[15:0]															

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b0	CMP[31:0]	Count baseline	Set baseline	RW

Unit 2~5: Reset value: 0xFFFF

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
CMP[15:0]															

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b15~b0	CMP[15:0]	Count baseline	Set baseline	RW

22.5.4 Control Status Register L (TMRA_BCSTRL)

Reset Value: 0x02

b7	b6	b5	b4	b3	b2	b1	b0
CKDIV[3:0]				SYNST	MODE	DIR	START

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b7~b4	CKDIV[3:0]	Counting clock selection	0000: PCLK 0001: PCLK/2 0010: PCLK/4 0011: PCLK/8 0100: PCLK/16 0101: PCLK/32 0110: PCLK/64 0111: PCLK/128 1000: PCLK/256 1001: PCLK/512 1010: PCLK/1024	RW
b3	SYNST	Synchronous enable	0: The synchronous start of even-numbered units and odd-numbered units in the symmetrical unit invalid 1: The synchronous start of even-numbered units and odd-numbered units in the symmetrical unit valid **Note:** The bit setting of the odd-numbered TimerA or the TimerA without a symmetrical unit in the symmetrical unit invalid and is read as 0.	RW
b2	MODE	Counting mode	0: Sawtooth mode 1: Triangular wave mode	RW
b1	DIR	Counting direction	0: Counter down 1: Counter up **Note:** In triangle wave mode, this bit indicates the counter direction status.	RW
b0	START	Timer start	0: Timer off 1: Timer Startup **Note:** 1. This bit will automatically change to 0 if the hardware stop condition is valid 2. When the synchronous start function of the even-numbered unit in the symmetrical unit valid, after the odd-numbered unit software is started, this bit of the even-numbered unit will also be set	RW

Note:

1. *For TimerA unit 1 to unit 4, PCLK is PCLK0; for TimerA unit 5, PCLK is PCLK1.*
2. *In triangle wave mode, do not write to this register while the timer is in operation.*

22.5.5 Control Status Register H (TMRA_BCSTRH)

Reset Value: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
UDFF	OVFF	ITENUDF	ITENOVF	Res			OVSTP

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b7	UDFF	Underflow mark	0: Count down, no count down 1: Count down, count down	RW
b6	OVFF	Overflow mark	0: When counting up, no overflow occurs 1: When counting up, the count overflows	RW
b5	ITENUDF	Underflow interrupt enable	0: Count underflow interrupt not enabled 1: Count underflow interrupt enable	RW
b4	ITENOVF	Overflow interrupt enable	0: Count overflow interrupt not enabled 1: Count overflow interrupt enable	RW
b3~b1	Res	Reserved Bit	Read at “0”, write at “0”	RW
b0	OVSTP	Count overflow stop control	0: Counter continues counting after overflow or underflow 1: Counter stops counting after overflowing or underflowing	RW

22.5.6 Interrupt Control Register (TMRA _ ICONR)

Reset value: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res												ITEN4	ITEN3	ITEN2	ITEN1

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b15~b8	Res	Reserved Bit	Read at “0”, write at “0”	RW
b3	ITEN4	Count Match interrupt enable 4	0:CMPAR4 register is invalid when it is equal to the count value or when a capture input event occurs 1: CMPAR4 This interrupt is enabled when CMPAR1register is equal to the count value, or when a capture input event occurs	RW
b2	ITEN3	Count Match interrupt enable 3	0:CMPAR3 register is invalid when it is equal to the count value or when a capture input event occurs 1: CMPAR3 This interrupt is enabled when CMPAR1register is equal to the count value, or when a capture input event occurs	RW
b1	ITEN2	Count Match interrupt enable 2	0:CMPAR2 register is invalid when it is equal to the count value or when a capture input event occurs 1: CMPAR2 This interrupt is enabled when CMPAR1register is equal to the count value, or when a capture input event occurs	RW
b0	ITEN1	Count Match interrupt enable 1	0:CMPAR1 register is invalid when it is equal to the count value or when a capture input event occurs 1: CMPAR1 This interrupt is enabled when CMPAR1register is equal to the count value, or when a capture input event occurs	RW

22.5.7 Event Control Register (TMRA _ ECONR)

Reset value: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res												ETEN4	ETEN3	ETEN2	ETEN1

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b15~b4	Res	Reserved Bit	Read at “0”, write at “0”	RW
b3	ETEN4	Count match event enable 4	0: CMPAR4 register output is invalid when it is equal to a count value or when a capture input event occurs 1: When CMPAR4 register is equal to the count value or a capture input event occurs, the event output is enabled	RW
b2	ETEN3	Count Match Event Enabled 3	0: CMPAR3 register output is invalid when it is equal to a count value or when a capture input event occurs 1: When CMPAR3 register is equal to the count value or a capture input event occurs, the event output is enabled	RW
b1	ETEN2	Count Match Event Enabled 2	0: CMPAR2 register output is invalid when it is equal to a count value or when a capture input event occurs 1: When CMPAR2 register is equal to the count value or a capture input event occurs, the event output is enabled	RW
b0	ETEN1	Count Match Event Enabled 1	0: CMPAR1 register output is invalid when it is equal to a count value or when a capture input event occurs 1: When CMPAR1 register is equal to the count value or a capture input event occurs, the event output is enabled	RW

22.5.8 Filter Control Register (TMRA _ FCONR)

Reset value: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res	NOFI CKCB[1:0]		NOFI ENCB	Res	NOFI CKCA[1:0]		NOFI ENCA	Res					NOFI CKTG[1:0]		NOFI ENTG

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b15	Res	Reserved Bit	Read at “0”, write at “0”	RW
b14~b13	NOFICKCB[1:0]	Filter Sampling Reference Clock Selection CB	00：PCLK 01：PCLK/4 10：PCLK/16 11：PCLK/64	RW
b12	NOFIENCB	Capture input port filter CB	0: TMRA_<t>_CLKB port input filter function invalid 1: TMRA_<t>_CLKB port input filter Function Enable	RW
b11	Res	Reserved Bit	Read at “0”, write at “0”	RW
b10~b9	NOFICKCA[1:0]	Filter sampling reference clock selection CA	00：PCLK 01：PCLK/4 10：PCLK/16 11：PCLK/64	RW
b8	NOFIENCA	Capture input port filter CA	0: TMRA_<t>_CLKA port input filter function invalid 1: TMRA_<t>_CLKA port input filter Function Enable	RW
b7~b3	Res	Reserved Bit	Read at “0”, write at “0”	RW
b2~b1	NOFICKTG[1:0]	Filter sampling reference clock selection TG	00：PCLK 01：PCLK/4 10：PCLK/16 11：PCLK/64	RW
b0	NOFIENTG	Capture input port filter TG	0: TMRA_<t>_TRIG input port filtering function invalid 1: TMRA_<t>_TRIG input port filter Function Enable	RW

Note:

For TimerA unit 1 to unit 4, PCLK is PCLK0; for TimerA unit 5, PCLK is PCLK1.

22.5.9 Status Flag Register (TMRA _ STFLR)

Reset value: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res				ICPF4	ICPF3	ICPF2	ICPF1	Res				CMPF 4	CMPF 3	CMPF 2	CMPF 1

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b15~b12	Res	Reserved Bit	Read at “0”, write at “0”	RW
b11	ICPF4	Repeat capture flag 4	0: No capture completion occurs after the count match flag CMPF4 is 1 1: After the count match flag CMPF4 is 1, the capture completion action occurs again	RW
b10	ICPF3	Repeat capture flag 3	0: No capture completion occurs after the count match flag CMPF3 is 1 1: After the count match flag CMPF3 is 1, the capture completion action occurs again	RW
b9	ICPF2	Repeat capture flag 2	0: No capture completion occurs after the count match flag CMPF2 is 1 1: After the count match flag CMPF2 is 1, the capture completion action occurs again	RW
b8	ICPF1	Repeat capture flag 1	0: No capture completion occurs after the count match flag CMPF1 is 1 1: After the count match flag CMPF1 is 1, the capture completion action occurs again	RW
b7~b4	Res	Reserved Bit	Read at “0”, write at “0”	RW
b3	CMPF4	Count Match Flags 4	0: The value of the CMPAR4 register is not equal to the count value, and the capture completion action has not occurred. 1: The value of the CMPAR4 register is equal to the count value, or a capture completion action occurs	RW
b2	CMPF3	Count Match Flags 3	0: The value of the CMPAR3 register is not equal to the count value, and the capture completion action has not occurred. 1: The value of the CMPAR3 register is equal to the count value, or a capture completion action occurs	RW
b1	CMPF2	Count Match Flags 2	0: The value of the CMPAR2 register is not equal to the count value, and the capture completion action has not occurred. 1: The value of the CMPAR2 register is equal to the count value, or a capture completion action occurs	RW
b0	CMPF1	Count Match Flags 1	0: The value of the CMPAR1 register is not equal to the count value, and the capture completion action has not occurred. 1: The value of the CMPAR1 register is equal to the count value, or a capture completion action occurs	RW

Note:

The flag bit in this register can only be written to 0, not 1.

22.5.10 Cache Control Register (TMRA_BCONRm, m=1~4)

Reset value: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res												BSEN	BSE1	BSE0	BEN

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b15~b4	Res	Reserved Bit	Read at “0”, write at “0”	RW
b3	BSEN	Cache special enable	0: When the CNTER register of the symmetric unit is read, this unit does not transfer the cache value 1: When the CNTER register of the symmetric unit is read, this unit transfers the cache value, that is: CMPARm -> CMPARn (m=2, 4，n=1, 3)	RW
b2	BSE1	Triangulation caching option 1	0: Cached value does not pass when triangulation mode counts to valley point 1: When the triangulation mode counts to the valley point, the cached value is transferred, i.e.: CMPARm -> CMPARn (m=2, 4，n=1, 3) **Note:** The configuration of this bit valid only when BEN is 1.	RW
b1	BSE0	Triangulation caching option 0	0: Cached value does not pass when triangulation mode counts to peak. 1: When the triangulation mode counts to the peak, the cached value is transferred, i.e.: CMPARm -> CMPARn (m=2, 4，n=1, 3) **Note:** The configuration of this bit valid only when BEN is 1.	RW
b0	BEN	Cache Enable	0: Invalid Cache Function for CMPARn Benchmark 1: The Cache Function of CMPARn Benchmark (n=1, 3)	RW

Note:

- *Please do not set BEN and BSEN to 1 at the same time.*
- *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 cache value is transferred.*

22.5.11 Capture Control Register (TMRA _ CCONRm, m = 1 ~ 4)

Reset value: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res	NOFI CKCP[1:0]		NOFI ENCP	HICP 6	HICP 5	HICP 4	HICP 3	Res	HICP 2	HICP 1	HICP 0	Res			CAPM D

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b15	Res	Reserved Bit	Read at “0”, write at “0”	RW
b14~b13	NOFICKCP[1:0]	Filter sampling reference clock selection CP	00：PCLK 01：PCLK/4 10：PCLK/16 11：PCLK/64	RW
b12	NOFIENCP	Capture input port filtering CP	0: TMRA_<t>_PWMm port input filter function invalid 1: TMRA_<t>_PWMm port input filter function enable (m=1~4)	RW
b11	HICP6	Capture input condition enable 6	0: When the TMRA_<t>_XOR signal is sampled to the falling edge, no capture input action occurs 1: When the TMRA_<t>_XOR signal is sampled to the falling edge, a capture input action is generated **Note:** This bit is valid only for CCONR4 register. That is, after this bit is valid 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
b10	HICP5	Capture input condition enable 5	0: When the TMRA_<t>_XOR signal is sampled to the rising edge, no capture input action occurs 1: When the TMRA_<t>_XOR signal is sampled to the rising edge, a capture input action is generated **Note:** This bit is valid only for CCONR4 register. That is, after this bit is valid 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
b9	HICP4	Capture input condition enable 4	0: When the TMRA_<t>_TRIG port input is sampled to the falling edge, no capture input action occurs 1: When the TMRA_<t>_TRIG port input is sampled to the falling edge, a capture input action is generated **Note:** This bit is valid only for CCONR3 register. That is, after this bit is valid 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
b8	HICP3	Capture input condition enable 3	0: When the TMRA_<t>_TRIG port input is sampled to the rising edge, no capture input action occurs 1: When the TMRA_<t>_TRIG port input is sampled to the rising edge, a capture input action is generated **Note:** This bit is valid only for CCONR3 register. That is, after this bit is valid 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
b7	Res	Reserved Bit	Read at “0”, write at “0”	RW
b6	HICP2	Capture input condition enable 2	0: When the event specified in the TMRA_TRGSEL register occurs, no capture input action occurs 1: When the event specified in the TMRA_TRGSEL register occurs, a capture input action is generated **Note:** Please refer to the notes in the Register Description section for details.	RW
b5	HICP1	Capture input condition enable 1	0: When the TMRA_<t>_PWMm port input is sampled to the falling edge, no capture input action occurs 1: When the TMRA_<t>_PWMm port input is sampled to the falling edge, a capture input action is generated (m=1~4)	RW
b4	HICP0	Capture input condition enable 0	0: When the TMRA_<t>_PWMm port input is sampled to the rising edge, no capture input action occurs 1: When the TMRA_<t>_PWMm port input is sampled to the rising	RW

			edge, a capture input action is generated (m=1~4)	
b3~b1	Res	Reserved Bit	Read at “0”, write at “0”	RW
b0	CAPMD	Functional mode selection	0: Comparison output function 1: Capture input function	RW

Note:

1. *For TimerA unit 1 to unit 4, PCLK is PCLK0; for TimerA unit 5, PCLK is PCLK1.*
2. *The TMRA_<t>_XOR signal is the signal obtained by XORing the quadrature inputs AIN and BIN of each unit.*

22.5.12 Port Control Register (TMRA _ PCONRm, m = 1 ~ 4)

Reset value: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res			OUTEN	Res		FORC[1:0]		PERC[1:0]		CMPC[1:0]		STPC[1:0]		STAC[1:0]	

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b15~b13	Res	Reserved Bit	Read at “0”, write at “0”	RW
b12	OUTEN	Output enable	0: TMRA_<t>_PWMm port output invalid when in PWM output function 1: TMRA_<t>_PWMm port output valid when PWM output function is in progress (m=1~4)	RW
b11~b10	Res	Reserved Bit	Read at “0”, write at “0”	RW
b9~b8	FORC[1:0]	Forced port status setting	0x: Invalid setting 10: The next cycle starts, and the TMRA_<t>_PWMm port output is set to a low level 11: The next cycle starts, and the TMRA_<t>_PWMm port output is set to a high level (m=1~4) **Note:** 1. The next cycle refers to the hardware counting mode or the zigzag wave counting to the overflow point or underflow point, the triangular wave counting to the valley point 2. This register bit can be used to achieve 0% or 100% control of PWM output duty cycle.	RW
b7~b6	PERC[1:0]	Port status setting when period values match	00: When the count value is equal to PERAR, the TMRA_<t>_PWMm port output is set to low level 01: When the count value is equal to PERAR, the TMRA_<t>_PWMm port output is set to high level 10: When the count value is equal to PERAR, the TMRA_<t>_PWMm port output maintains the previous state 11: When the count value is equal to PERAR, the TMRA_<t>_PWMm port output is set to the inverted level (n = 1 ~ 4) Note: In sawtooth wave mode, the period point for upward counting is PERAR, while the period point for downward counting in sawtooth wave mode and for triangular wave mode is zero.	RW
b5~b4	CMPC[1:0]	Port status setting when comparison values match	00: When the count value is equal to CMPARn, the TMRA_<t>_PWMm port output is set to low level 01: When the count value is equal to CMPARn, the TMRA_<t>_PWMm port output is set to high level 10: When the count value is equal to CMPARn, the TMRA_<t>_PWMm port output maintains the previous state 11: When the count value is equal to CMPARn, the TMRA_<t>_PWMm port output is set to the inverted level (m=1~4)	RW
b3~b2	STPC[1:0]	Port status setting when count stops	00: When counting stops, the TMRA_<t>_PWMm port output is set to low level 01: When counting stops, the TMRA_<t>_PWMm port output is set to high level 10: When counting stops, the TMRA_<t>_PWMm port output maintains the previous state 11: When counting stops, the TMRA_<t>_PWMm port output maintains the previous state (m=1~4)	RW
b1~b0	STAC[1:0]	Port status setting at start of count	00: When counting starts, the TMRA_<t>_PWMm port output is set to low level 01: When counting starts, the TMRA_<t>_PWMm port output is set to high level 10: When counting starts, the TMRA_<t>_PWMm port output maintains the previous state 11: When counting starts, the TMRA_<t>_PWMm port output maintains the previous state (m=1~4) **Note:** This bit setting is only valid when there is no frequency division (BCSTRL.CKDIV=0b0000). For other frequency divisions, please set it to 0b10 or 0b11	RW

22.5.13 Hardware Trigger Event Selection Register (TMRA _ HCONR)

Reset value: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
HCLE6	HCLE5	HCLE4	HCLE3	Res	HCLE2	HCLE1	HCLE0	Res	HSTP2	HSTP1	HSTP0	Res	HSTA2	HSTA1	HSTA0

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b15	HCLE6	Hardware zeroing condition 6	Condition: TMRA_<t>_PWM3 port input is sampled to the falling edge 0: Hardware zeroing is invalid when conditions match 1: Hardware zeroing is valid when conditions match	RW
b14	HCLE5	Hardware zeroing condition 5	Condition: TMRA_<t>_PWM3 port input sampling to the rising edge 0: Hardware zeroing is invalid when conditions match 1: Hardware zeroing is valid when conditions match	RW
b13	HCLE4	Hardware zeroing condition 4	Condition: The TRIG port input of the symmetric unit is sampled to the falling edge 0: Hardware zeroing is invalid when conditions match 1: Hardware zeroing is valid when conditions match	RW
b12	HCLE3	Hardware zeroing condition 3	Condition: The TRIG port input of the symmetric unit is sampled to the rising edge 0: Hardware zeroing is invalid when conditions match 1: Hardware zeroing is valid when conditions match	RW
b11	Res	Reserved Bit	Read at “0”, write at “0”	RW
b10	HCLE2	Hardware zeroing condition 2	Condition: The event specified in the TMRA_TRGSEL register occurs 0: Hardware zeroing is invalid when conditions match 1: Hardware zeroing is valid when conditions match Note: Please refer to the notes in the Register Description section for details.	RW
b9	HCLE1	Hardware zeroing condition 1	Condition: TMRA_TRIG port input is sampled to the falling edge 0: Hardware zeroing is invalid when conditions match 1: Hardware zeroing is valid when conditions match	RW
b8	HCLE0	Hardware zeroing condition 0	Condition: TMRA_TRIG port input is sampled to the rising edge 0: Hardware zeroing is invalid when conditions match 1: Hardware zeroing is valid when conditions match	RW
b7	Res	Reserved Bit	Read at “0”, write at “0”	RW
b6	HSTP2	Hardware stop condition 2	Condition: The event specified in the TMRA_TRGSEL register occurs 0: Hardware stop invalid when condition matches 1: Hardware stops working when conditions match Note: Please refer to the notes in the Register Description section for details.	RW
b5	HSTP1	Hardware stop condition 1	Condition: TMRA_<t>_TRIG port input is sampled to the falling edge 0: Hardware stop invalid when condition matches 1: Hardware stops working when conditions match	RW
b4	HSTP0	Hardware stop condition 0	Condition: TMRA_<t>_TRIG port input is sampled to the rising edge 0: Hardware stop invalid when condition matches 1: Hardware stops working when conditions match	RW
b3	Res	Reserved Bit	Read at “0”, write at “0”	RW
b2	HSTA2	Hardware startup condition 2	Condition: The event specified in the TMRA_TRGSEL register occurs 0: Hardware startup is invalid when conditions match 1: Hardware startup is valid when conditions match **Note:** Please refer to the notes in the Register Description section for details.	RW
b1	HSTA1	Hardware startup condition 1	Conditions: 1) When the synchronous start function invalid, the TMRA_<t>_TRIG port input is sampled to the falling edge 2) When the synchronous start function valid, the TMRA_<t>_TRIG port input of the symmetric unit is sampled to the falling edge (only channel 2 and channel 4 support this function) 0: Hardware startup is invalid when conditions match 1: Hardware startup is valid when conditions match	RW

b0	HSTA0	Hardware startup condition 0		Conditions: 1) When the synchronous start function invalid, the TMRA_<t>_TRIG port input is sampled to the rising edge 2) When the synchronous start function valid, the TMRA_<t>_TRIG port input of the symmetric unit is sampled to the rising edge (only channel 2 and channel 4 support this function) 0: Hardware startup is invalid when conditions match 1: Hardware startup is valid when conditions match	RW

22.5.14 Hardware Added Event Selection Register (TMRA _ HCUPR)

Reset value: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res			HCUP12	HCUP11	HCUP10	HCUP9	HCUP8	HCUP7	HCUP6	HCUP5	HCUP4	HCUP3	HCUP2	HCUP1	HCUP0

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b15~b13	Res	Reserved Bit	Read at “0”, write at “0”	RW
b12	HCUP12	Hardware Added Condition 12	Condition: Count underflow occurs in the symmetrical unit 0: When the condition is matched, the hardware increment of this unit invalid 1: When the conditions are matched, the hardware of this unit is incremented and effective	RW
b11	HCUP11	Hardware Added Conditions11	Condition: Count overflow occurs in the symmetrical unit 0: When the condition is matched, the hardware increment of this unit invalid 1: When the conditions are matched, the hardware of this unit is incremented and effective	RW
b10	HCUP10	Hardware Added Condition 10	Condition: The event specified in the TMRA_TRGSEL register occurs 0: Hardware add invalid when condition matches 1: Hardware addition is valid when a condition matches **Note:** Please refer to the notes in the Register Description section for details.	RW
b9	HCUP9	Hardware Added Condition 9	Condition: TMRA_<t>_TRIG port is sampled to the falling edge 0: Hardware add invalid when condition matches 1: Hardware addition is valid when a condition matches	RW
b8	HCUP8	Hardware Added Condition 8	Condition: TMRA_<t>_TRIG port is sampled to the rising edge 0: Hardware add invalid when condition matches 1: Hardware addition is valid when a condition matches	RW
b7	HCUP7	Hardware Added Condition 7	Condition: When the TMRA_<t>_CLKB port is high, the TMRA_<t>_CLKA port is sampled to the falling edge 0: Hardware add invalid when condition matches 1: Hardware addition is valid when a condition matches	RW
b6	HCUP6	Hardware Added Condition 6	Condition: When the TMRA_<t>_CLKB port is high, the TMRA_<t>_CLKA port is sampled to the rising edge 0: Hardware add invalid when condition matches 1: Hardware addition is valid when a condition matches	RW
b5	HCUP5	Hardware Added Condition 5	Condition: When the TMRA_<t>_CLKB port is low, the TMRA_<t>_CLKA port is sampled to the falling edge 0: Hardware add invalid when condition matches 1: Hardware addition is valid when a condition matches	RW
b4	HCUP4	Hardware Added Condition 4	Condition: When the TMRA_<t>_CLKB port is low, the TMRA_<t>_CLKA port is sampled to the rising edge 0: Hardware add invalid when condition matches 1: Hardware addition is valid when a condition matches	RW
b3	HCUP3	Hardware Added Condition 3	Condition: When the TMRA_<t>_CLKA port is high, the TMRA_<t>_CLKB port is sampled to the falling edge 0: Hardware add invalid when condition matches 1: Hardware addition is valid when a condition matches	RW
b2	HCUP2	Hardware Added Condition 2	Condition: When the TMRA_<t>_CLKA port is high, the TMRA_<t>_CLKB port is sampled to the rising edge 0: Hardware add invalid when condition matches 1: Hardware addition is valid when a condition matches	RW
b1	HCUP1	Hardware Added Condition 1	Condition: When the TMRA_<t>_CLKA port is low, the TMRA_<t>_CLKB port is sampled to the falling edge 0: Hardware add invalid when condition matches 1: Hardware addition is valid when a condition matches	RW
b0	HCUP0	Hardware Added Condition 0	Condition: When the TMRA_<t>_CLKA port is low, the TMRA_<t>_CLKB port is sampled to the rising edge 0: Hardware add invalid when condition matches 1: Hardware addition is valid when a condition matches	RW

22.5.15 Hardware Decrease Event Select Register (TMRA _ HCDOR)

Reset value: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res			HC DO12	HC DO11	HC DO10	HC DO9	HC DO8	HC DO7	HC DO6	HC DO5	HC DO4	HC DO3	HC DO2	HC DO1	HC DO0

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b15~b13	Res	Reserved Bit	Read at “0”, write at “0”	RW
b12	HCDO12	Hardware decrement condition 12	Condition: Count underflow occurs in the symmetrical unit 0: When the condition is matched, the hardware decrement of this unit invalid 1: When the conditions are matched, the hardware of this unit is decremented and effective	RW
b11	HCDO11	Hardware Decrement Condition \11	Condition: Count overflow occurs in the symmetric unit 0: When the condition is matched, the hardware decrement of this unit invalid 1: When the conditions are matched, the hardware of this unit is decremented and effective	RW
b10	HCDO10	Hardware decrement condition 10	Condition: The event specified in the TMRA_TRGSEL register occurs 0: Hardware decrement is invalid when conditions match 1: Hardware decrements are effective when conditions match Note: Please refer to the notes in the Register Description section for details.	RW
b9	HCDO9	Hardware decrement condition 9	Condition: TMRA_<t>_TRIG port is sampled to the falling edge 0: Hardware decrement is invalid when conditions match 1: Hardware decrements are effective when conditions match	RW
b8	HCDO8	Hardware decrement condition 8	Condition: TMRA_<t>_TRIG port is sampled to the rising edge 0: Hardware decrement is invalid when conditions match 1: Hardware decrements are effective when conditions match	RW
b7	HCDO7	Hardware decrement condition 7	Condition: When the TMRA_<t>_CLKB port is high, the TMRA_<t>_CLKA port is sampled to the falling edge 0: Hardware decrement is invalid when conditions match 1: Hardware decrements are effective when conditions match	RW
b6	HCDO6	Hardware decrement condition 6	Condition: When the TMRA_<t>_CLKB port is high, the TMRA_<t>_CLKA port is sampled to the rising edge 0: Hardware decrement is invalid when conditions match 1: Hardware decrements are effective when conditions match	RW
b5	HCDO5	Hardware decrement condition 5	Condition: When the TMRA_<t>_CLKB port is low, the TMRA_<t>_CLKA port is sampled to the falling edge 0: Hardware decrement is invalid when conditions match 1: Hardware decrements are effective when conditions match	RW
b4	HCDO4	Hardware decrement condition 4	Condition: When the TMRA_<t>_CLKB port is low, the TMRA_<t>_CLKA port is sampled to the rising edge 0: Hardware decrement is invalid when conditions match 1: Hardware decrements are effective when conditions match	RW
b3	HCDO3	Hardware decrement condition 3	Condition: When the TMRA_<t>_CLKA port is high, the TMRA_<t>_CLKB port is sampled to the falling edge 0: Hardware decrement is invalid when conditions match 1: Hardware decrements are effective when conditions match	RW
b2	HCDO2	Hardware decrement condition 2	Condition: When the TMRA_<t>_CLKA port is high, the TMRA_<t>_CLKB port is sampled to the rising edge 0: Hardware decrement is invalid when conditions match 1: Hardware decrements are effective when conditions match	RW
b1	HCDO1	Hardware decrement condition 1	Condition: When the TMRA_<t>_CLKA port is low, the TMRA_<t>_CLKB port is sampled to the falling edge 0: Hardware decrement is invalid when conditions match 1: Hardware decrements are effective when conditions match	RW
b0	HCDO0	Hardware decrement condition 0	Condition: When the TMRA_<t>_CLKA port is low, the TMRA_<t>_CLKB port is sampled to the rising edge 0: Hardware decrement is invalid when conditions match 1: Hardware decrements are effective when conditions match	RW

23 General Timer (Timer0)

23.1 Overview

General timer 0 (Timer0) is a basic timer that can implement synchronous counting and asynchronous counting. The timer contains two channels (CH-A and CH-B) that can generate compare match events and count overflow events during counting. This event can trigger an interrupt or be used to trigger other module actions. This series of products is equipped with 2 units of Timer0.

23.2 Functional Description

23.2.1 Timer0 Block Diagram

The basic block diagram of Timer0 is shown in Figure 23-1.

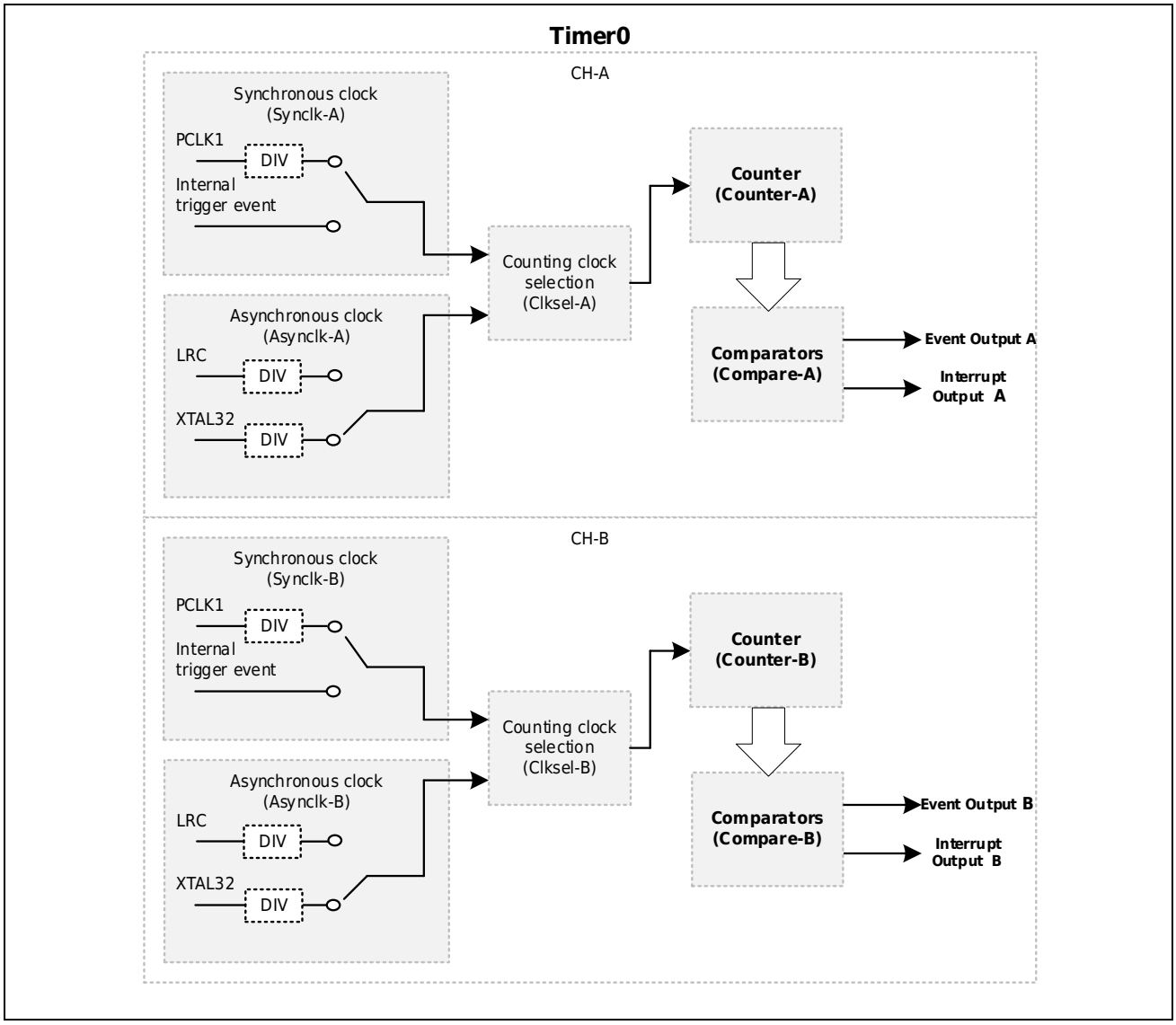


Figure 23-1 Timer0 Basic Block Diagram

23.2.2 Timer 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.

23.2.2.1 Synchronous Count Timer

In synchronous counting mode (BCONR.SYNSA<B>=0), the clock source can have the following options (BCONR.SYNCLKA<B> setting):

a) PCLK1 and PCLK1 divided by 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024 as synchronous count clock (BCONR.SYNCLK A<B>=0 & BCONR.CKDIV A<B> [3:0] setting)

b) Internal hardware trigger event input as synchronous count clock (BCONR.SYNCLK A<B>=1)

23.2.2.2 Asynchronous Count Timer

In asynchronous counting mode (BCONR.SYNS A<B>=1), the clock source can be selected from the following options (BCONR.ASYNCLK A<B> setting selection):

a) LRC clock source input and its 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024 divisions as asynchronous count clocks (BCONR.ASYNCLK A<B>=0 & BCONR.CKDIV A<B> [3:0] settings)

b) XTAL32 clock source input and its 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024 division as asynchronous counting clock (BCONR.ASYNCLK A<B>=1 & BCONR.CKDIV A<B> [3:0] setting)

23.2.3 Basic Count

Each channel of Timer0 can set a reference count value, and a count comparison match event will be generated when the count value is equal to the reference value. As shown in Figure 23-2.

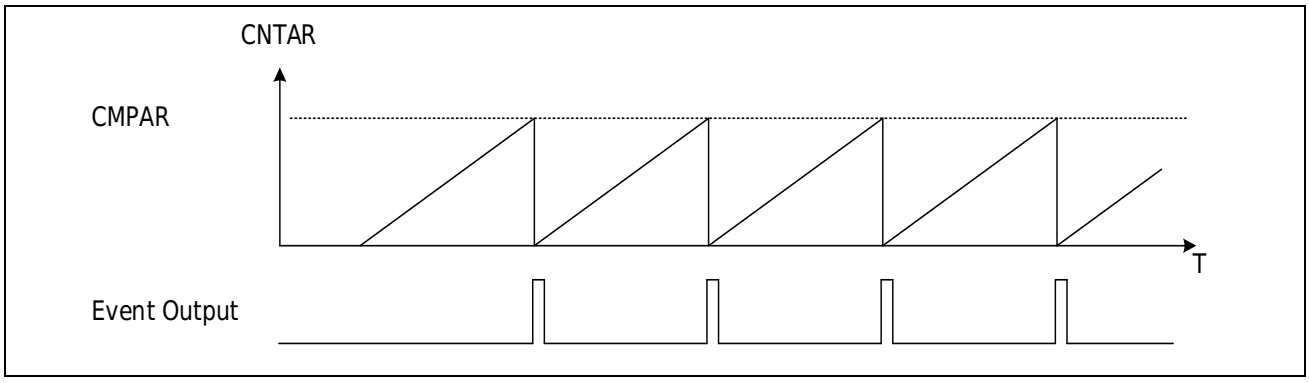


Figure 23-2 Timer0 Counting Timing Diagram

23.2.4 Hardware Triggering

The two channels of Timer0 have a common internal hardware trigger source, which can be used to control the state of the timer (start, stop, clear) and capture input actions through the relevant settings of the basic control register (BCONR).

The source selection of the hardware trigger source is achieved by entering the corresponding number into the hardware trigger selection register (TMR0_TRGSEL). For the register content, please refer to the [Automatic Operation System (AOS)] section. For the specific event correspondence, please refer to the [Interrupt controller (INTC)] section. When using the internal hardware trigger function, you need to enable the AOS function control bit of the function clock control register 0 (PWC_FCG0) first.

23.2.5 Software Synchronization Start

The product is equipped with a count synchronization enable register (TMR_SYNENR) to synchronously start all timers (including Timer0, Timer4, Timer6, TimerA) carried by the product. By setting multiple start bits in the TMR_SYNENR register to 1, multiple timers can start counting at the same time. For the specific allocation of the start bit, please refer to the Count Synchronization Enable Register (TMR_SYNENR).

Note:

1. *When using the synchronous start function, make sure that the PCLK clock frequency of all synchronous start timers is the same.*
2. *When using the synchronous start function, please ensure that all synchronous start timers are in synchronous counting mode.*
3. *When the counting clock selects the divided clock of PCLK, the start time of Timer6 is not synchronized with other timers. Please do not mix them.*

23.3Interrupt and Events

23.3.1 Interrupt

One Timer0 has 4 interrupt outputs, namely, the count compare match interrupt or capture input interrupt of channel A and channel B, and the count overflow interrupt of channel A and channel B.

There are 2 reference value registers (CMPAR, CMPBR), which can be compared with the count value registers (CNTAR, CNTBR) respectively to generate comparison matching valid signals. When the count compare matches, the STFLR.CMFA<B> bit in the status flag register (STFLR) will be set to 1. At this time, if the BCONR.CMENA<B> bit of the basic control register (BCONR) is set to enable interrupt, the corresponding interrupt request (TMR0_m_CMPn, m=1, 2; n=A, B) will also be triggered.

When the internal hardware triggers input as the capture input condition, the corresponding capture input action can be generated. At this time, if the BCONR.CMENA<B> bit of the basic control register (BCONR) is set to enable interrupt, the corresponding interrupt request (TMR0_m_CMPn, m=1, 2; n=A, B) is triggered.

When the count value register (CNTAR, CNTBR) counts to 0xFFFF, a count overflow event occurs. At this time, the STFLR.OVFA<B> bit in the status flag register (STFLR) will be set to 1. If the BCONR.OVENA<B> bit of the basic control register (BCONR) is set to enable interrupt, the corresponding interrupt request (TMR0_m_OVFn, m=1, 2; n=A, B) will also be triggered.

When the asynchronous counting mode is selected, the compare match interrupt generated by the reference value register (CMPAR) of unit 1 can be used to wake up the system in Stop Mode. For details, please refer to the [Interrupt controller (INTC)] section.

23.3.2 Events

One Timer0 has four event outputs, namely, the count compare match events or capture input events of channel A and channel B, and the count overflow events of channel A and channel B.

When a count compare match, capture input action or count overflow occurs during the counting process, the corresponding event request (TMR0_m_CMPn or TMR0_m_OVFn, m=1, 2; n=A, B) output signal will be generated, which can be used to select and trigger other modules.

23.4Register Description

Table 23-1 Timer0/TMR_SYNENR Base Address

Name	Base Address	Description
Timer0_1	0x4002 4000	Timer0_1 base address
Timer0_2	0x4002 4400	Timer0_2 base address
TMR_SYNENR	0x4005 5400	TMR_SYNENR Base Address

Table 23-2 Timer0 Register List

Register	Description	Offset Address	Bit Width	Reset Value
TMR0_CNTAR	Register count value	0x0000	32	0x0000 0000
TMR0_CNTBR	Register count value	0x0004	32	0x0000 0000
TMR0_CMPAR	Reference value register	0x0008	32	0x0000 FFFF
TMR0_CMPBR	Reference value register	0x000C	32	0x0000 FFFF
TMR0_BCONR	Basic control register	0x0010	32	0x0000 0000
TMR0_STFLR	Status flag register	0x0014	32	0x0100 0100

Table 23-3 TMR_SYNENR Register Llist

Register	Description	Offset Address	Bit Width	Reset Value
TMR_SYNENR	Count Synchronization Enable Register	0x14	32	0x0000 0000

23.4.1 Count Value Register (TMR0_CNTMR, m=A~B)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
CNTA<B>[15:0]															

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b16	Res	Reserved Bit	Read at “0”, write at “0”	R
b15~b0	CNTA<B>[15:0]	Count value	Current timer count **Note:** During counting, values cannot be written to this register.	RW

23.4.2 Reference Value Register (TMR0_CMPmR, m=A~B)

Reset value: 0x0000 FFFF

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
CMPA<B>[15:0]															

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b16	Res	Reserved Bit	When reading, it is "0", when writing, it is written "0"	R
b15~b0	CMPA<B>[15:0]	Reference value	Set the counting reference value to generate a compare match event	RW

23.4.3 Basic Control Register (TMR0 _ BCONR)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
HICPB	HCLEB	HSTPB	HSTAB	Res	ASYNCLKB	SYNCLKB	SYNSB	CKDIVB[3:0]				OVENB	CMENB	CAPMDB	CSTB
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
HICPA	HCLEA	HSTPA	HSTAA	Res	ASYNCLKA	SYNCLKA	SYNSA	CKDIVA[3:0]				OVENA	CMENA	CAPMDA	CSTA

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31	HICPB	Hardware trigger capture input B	Condition: Internal Hardware Trigger Event Valid 0: Capture input is invalid when a condition matches 1: Capture input is valid when conditions match	RW
b30	HCLEB	Hardware trigger clear B	Condition: Internal Hardware Trigger Event Valid 0: Timer zeroing is invalid when conditions match 1: Timer zeroing is valid when conditions match	RW
b29	HSTPB	Hardware trigger stop B	Condition: Internal Hardware Trigger Event Valid 0: Timer stops invalid when condition matches 1: When the condition matches, the timer stops working	RW
b28	HSTAB	Hardware trigger start B	Condition: Internal Hardware Trigger Event Valid 0: Timer start invalid when condition matches 1: Timer is active when condition matches	RW
b27	Res	Reserved Bit	Read at “0”, write at “0”	RW
b26	ASYNCLKB	Channel B asynchronous counting clock source selection	0：LRC 1：XTAL32 **Note:** When the timeout function of the corresponding channel of UART is enabled, the clock source is no longer XTAL32 (hardware The XTAL32 crystal oscillator may not exist in the system, but the clock tick generated by the UART baud rate generator.	RW
b25	SYNCLKB	Channel B synchronous counting clock source selection	0：PCLK1 1: Internal Hardware Triggering Event	RW
b24	SYNSB	Channel B counting mode selection	0: Synchronous counting mode 1: Asynchronous counting	RW
b23~b20	CKDIVB[3:0]	Channel B counting clock frequency division selection	Channel B counting clock frequency division selection: 0000: Timer 0001: Timer/2 0010: Timer/4 0011: Timer/8 0100: Timer/16 0101: Timer/32 0110: Timer/64 0111: Timer/128 1000: Timer/256 1001: Timer/512 1010: Timer/1024 Other: prohibitions **Note:** The timer to be divided can be a variety of timer at asynchronous counting, PCLK1 at synchronous counting.	RW
b19	OVENB	Count Overflow Interrupt Enable B	0: The interrupt is invalid when the count value (CNTBR) = 0xFFFF 1: When CNTBR = 0xFFFF, the interrupt is enabled	RW
b18	CMENB	Count match interrupt enable B	0: CMPBR register is equal to the count value (CNTBR), or the interrupt is invalid when a capture input event occurs 1: CMPBR register is equal to the count value (CNTBR), or the interrupt is enabled when a capture input event occurs	RW
b17	CAPMDB	Function mode selection B	0: Comparison output function 1: Capture input function	RW
b16	CSTB	Timer start	0: channel B timer off 1: Channel B timer starts **Note:** This bit automatically changes to 0 when the hardware trigger stop condition is valid	RW
b15	HICPA	Hardware Trigger Capture Input A	Condition: Internal Hardware Trigger Event Valid 0: Capture input is invalid when a condition matches 1: Capture input is valid when conditions match	RW

b14	HCLEA	Hardware triggered zeroing A	Condition: Internal Hardware Trigger Event Valid 0: Timer zeroing is invalid when conditions match 1: Timer zeroing is valid when conditions match	RW
b13	HSTPA	Hardware Trigger Stop A	Condition: Internal Hardware Trigger Event Valid 0: Timer stops invalid when condition matches 1: When the condition matches, the timer stops working	RW
b12	HSTAA	Hardware Trigger Startup A	Condition: Internal Hardware Trigger Event Valid 0: Timer start invalid when condition matches 1: Timer is active when condition matches	RW
b11	Res	Reserved Bit	Read at “0”, write at “0”	RW
b10	ASYNCLKA	Channel A asynchronous counting clock source selection	0：LRC 1：XTAL32 **Note:** When the timeout function of the corresponding channel of UART is enabled, the clock source is no longer XTAL32 (hardware The XTAL32 crystal oscillator may not exist in the system, but the clock tick generated by the UART baud rate generator.	RW
b9	SYNCLKA	Channel A Synchronous Count Timer Selection	0：PCLK1 1: Internal Hardware Triggering Event	RW
b8	SYNSA	Channel A counting mode selection	0: Synchronous counting mode 1: Asynchronous counting	RW
b7~b4	CKDIVA[3:0]	Channel A Clock Frequency Division Selection	Channel A count clock frequency division options: 0000: Timer 0001: Timer/2 0010: Timer/4 0011: Timer/8 0100: Timer/16 0101: Timer/32 0110: Timer/64 0111: Timer/128 1000: Timer/256 1001: Timer/512 1010: Timer/1024 Other: prohibitions **Note:** The timer to be divided can be a variety of timer at asynchronous counting, PCLK1 at synchronous counting.	RW
b3	OVENA	Count Overflow interrupt enable A	0: The interrupt is invalid when the count value (CNTAR) = 0xFFFF 1: When CNTAR = 0xFFFF, the interrupt is enabled	RW
b2	CMENA	Count Match interrupt enable A	0: CMPARregister is equal to the count value (CNTAR), or the interrupt is invalid when a capture input event occurs 1: CMPAR register is equal to the count value (CNTAR), or the interrupt is enabled when a capture input event occurs	RW
b1	CAPMDA	Functional mode selection A	0: Comparison output function 1: Capture input function	RW
b0	CSTA	Timer start	0: Channel A timer is off 1: Channel A timer starts **Note:** This bit automatically changes to 0 when the hardware trigger stop condition is valid	RW

Note:

– The internal hardware trigger events (bit31~bit28 and bit15~bit12) mentioned in this register and the XTAL32 clock source (bit26 and bit10) during asynchronous counting are all provided by the USART module when the TIMEOUT function of the USART module valid. For details, please refer to the introduction in the [Universal Synchronous Asynchronous Transceiver (USART)] section.

23.4.4 Status Flag Register (TMR0 _ STFLR)

Reset value: 0x0100 0100

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res							SYDB	Res					ICPB	OVFB	CMFB
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res							SYDA	Res					ICPA	OVFA	CMFA

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b25	Res	Reserved Bit	Read at “0”, write at “0”	RW
b24	SYDB	Synchronous completion flag B	0: Channel B register is synchronizing in asynchronous counting 1: B channel register synchronization completed under asynchronous counting	R
b23~b19	Res	Reserved Bit	Read at “0”, write at “0”	R
b18	ICPB	Repeat Capture Flag B	0: No capture input action occurs after the count match B flag CMFB is 1 1: After the count match B flag CMFB is 1, the capture input action occurs again	RW
b17	OVFB	Count overflow flag B	0: The count value (CNTBR) has not reached 0xFFFF 1: The count value (CNTBR) counts to 0xFFFF	RW
b16	CMFB	Count match flag B	0: The value of CMPBR register is not equal to the count value (CNTBR) and no capture input action occurs 1: The value of CMPAR register is equal to the count value (CNTBR) or captures the input action.	RW
b15~b9	Res	Reserved Bit	Read at “0”, write at “0”	RW
b8	SYDA	Synchronous completion flag A	0: A channel register is synchronizing in asynchronous counting 1: Asynchronous counting completes the synchronization of the A channel register	R
b7~b3	Res	Reserved Bit	Read as "0", write as "0"	R
b2	ICPA	Repeat capture flag A	0: No capture input action occurs after the count match A flag CMFA is 1 1: After the count match A flag CMFA is 1, the capture input action occurs again	RW
b1	OVFA	Count Overflow Flag A	0: The count value (CNTAR) has not reached 0xFFFF 1: The count value (CNTAR) counts to 0xFFFF	RW
b0	CMFA	Count Matching Flag A	0: The value of CMPARregister is not equal to the count value (CNTAR) and no capture input action occurs 1: The value of CMPARregister is equal to the count value (CNTAR) or captures the input action.	RW

Note:

– *The flag bit in this register can only be written to 0, not 1.*

23.4.5 Count Synchronization Enable Register (TMR_SYNENR)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res		HRPWMU6	HRPWMU5	HRPWMU4	HRPWMU3	HRPWMU2	HRPWMU1	Res			TMRAU5	TMRAU4	TMRAU3	TMRAU2	TMRAU1
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res				TMR6U4	TMR6U3	TMR6U2	TMR6U1	Res			TMR4U1	TMR0U2B	TMR0U2A	TMR0U1B	TMR0U1A

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b30	Res	Reserved Bit	Read at “0”, write at “0”	RW
b29	HRPWMU6	HRPWM unit 6 software start enable bit	0: HRPWM unit 6 software start invalid 1: HRPWM unit 6 software start valid and the timer starts counting	RW
b28	HRPWMU5	HRPWM unit 5 software start enable bit	0: HRPWM unit 5 software start invalid 1: HRWPM unit 5 software start valid and the timer starts counting	RW
b27	HRPWMU4	HRPWM unit 4 software start enable bit	0: HRPWM unit 4 software start invalid 1: HRWPM unit 4 software start valid and the timer starts counting	RW
b26	HRPWMU3	HRPWM unit 3 software start enable bit	0: HRPWM unit 3 software start invalid 1: HRWPM unit 3 software start valid and the timer starts counting	RW
b25	HRPWMU2	HRPWM unit 2 software start enable bit	0: HRPWM unit 2 software start invalid 1: HRWPM unit 2 software start valid and the timer starts counting	RW
b24	HRPWMU1	HRPWM unit 1 software start enable bit	0: HRPWM unit 1 software start invalid 1: HRWPM unit 1 software start valid and the timer starts counting	RW
b23~b21	Res	Reserved Bit	Read at “0”, write at “0”	RW
b20	TMRAU5	TMRA unit 5 software start enable bit	0: TMRA unit 5 software startup invalid 1: TMRA unit 5 software start valid and the timer starts counting	RW
b19	TMRAU4	TMRA unit 4 software start enable bit	0: TMRA unit 4 software startup invalid 1: TMRA unit 4 software start valid and the timer starts counting	RW
b18	TMRAU3	TMRA unit 3 software start enable bit	0: TMRA unit 3 software startup invalid 1: TMRA unit 3 software start valid and the timer starts counting	RW
b17	TMRAU2	TMRA unit 2 software start enable bit	0: TMRA unit 2 software startup invalid 1: TMRA unit 2 software start valid and the timer starts counting	RW
b16	TMRAU1	TMRA unit 1 software start enable bit	0: TMRA unit 1 software startup invalid 1: TMRA unit 1 software start valid and the timer starts counting	RW
b15~b12	Res	Reserved Bit	Read at “0”, write at “0”	RW
b11	TMR6U4	TMR6 unit 2 software start enable bit	0: TMR6 unit 4 software startup invalid 1: TMR6 unit 4 software start valid and the timer starts counting	RW
b10	TMR6U3	TMR6 unit 3 software start enable bit	0: TMRA unit 3 software startup invalid 1: TMRA unit 3 software start valid and the timer starts counting	RW
b9	TMR6U2	TMR6 unit 2 software start enable bit	0: TMR6 unit 2 software startup invalid 1: TMR6 unit 2 software start valid and the timer starts counting	RW
b8	TMR6U1	TMR6 unit 1 software start enable bit	0: TMR6 unit 1 software startup invalid 1: TMR6 unit 1 software start valid and the timer starts counting	RW
b7~b5	Res	Reserved Bit	Read at “0”, write at “0”	RW
b4	TMR4U1	TMR4 unit 1 software start enable bit	0: TMR4 unit 1 software start invalid 1: TMR4 unit 1 software start valid and the timer starts counting	RW
b3	TMR0U2B	TMR0 unit 2B channel software start enable bit	0: TMR0 unit 2 channel B software start invalid 1: TMR0 unit 2 channel B software start valid and the timer starts counting **Note:** Please use this function when synchronizing the count.	RW

b2	TMR0U2A	TMR0 unit 2A channel software start enable bit	0: TMR0 unit 2 channel A software start invalid 1: TMR0 unit 2 channel A software start valid and the timer starts counting **Note:** Please use this function when synchronizing the count.	RW
b1	TMR0U1B	TMR0 unit 1B channel software start enable bit	0: TMR0 unit 1 channel B software start invalid 1: TMR0 unit 1 channel B software start valid and the timer starts counting **Note:** Please use this function when synchronizing the count.	RW
b0	TMR0U1A	TMR0 unit 1A channel software start enable bit	0: TMR0 unit 1 channel A software start invalid 1: TMR0 unit 1 channel A software start valid and the timer starts counting **Note:** Please use this function when synchronizing the count.	RW

Note:

- *Do not use the count synchronization function when this timer is counting asynchronously.*
- *TMR_SYNENR is not controlled by the RMU_FRST register.*

23.5Precautions

1) In the asynchronous counting operation, first set the BCONR.ASYNCLKA<B> bit to select the asynchronous clock source, then set the BCONR.SYNSA<B> bit to select the asynchronous counting mode, and then start Timer0.

2) When asynchronous counting is selected, when the count value (CNTA<B>R), reference value (CMPA<B>R), start bit (BCONR.CSTA<B>), status bit (STFLR.CMFA<B>, STFLR.OVFA<B>, STFLR.ICPA<B>) are modified, Timer0 will write the modified value to the corresponding register after 6 asynchronous counting clocks after receiving the write action.

3) When asynchronous counting is selected, when continuously writing the count value (CNTA<B>R), reference value (CMPA<B>R), start bit (BCONR.CSTA<B>), status bit (STFLR.CMFA<B>, STFLR.OVFA<B>, STFLR.ICPA<B>), you need to wait for the status bit (STFLR.SYDA<B>) to be 1 before reading or writing any of the above registers.

4) To select asynchronous counting, set BCONR.SYNCLKA<B> to 0.

5) When writing BCONR.CSTA<B>, do not simultaneously modify BCONR.SYNSA<B> (BCONR.SYNSA<B> must retain its current value), otherwise BCONR.CSTA<B> cannot be written.

6) When triggering actions of other peripherals using a Timer0 event, the reference value of the corresponding Timer0 channel must be set greater than 5.

24 Real Time Clock (RTC)

24.1 Introduction

The 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.

24.2 Main Features

Table 24-1 shows its basic characteristics.

Table 24-1 Basic Specifications of RTC

Count clock source	External low-speed oscillator (32.768KHz) Internal low speed oscillator (32.768KHz) External clock pin XTAL Fractional Clock
Basic functions	BCD code indicates seconds, minutes, hours, days, weeks, months, years
	Software start or stop
	12/24 hour format optional, leap year automatic recognition
	Programmable Alarm Clock
	Distributed/uniform compensation 1Hz clock output
	Clock error compensation function
	Timestamp function
Interrupt	Cycle interruption
	Alarm interrupt

24.3Functional Description

24.3.1 RTC Block Diagram

The basic block diagram of the RTC is shown in the Figure 24-1.

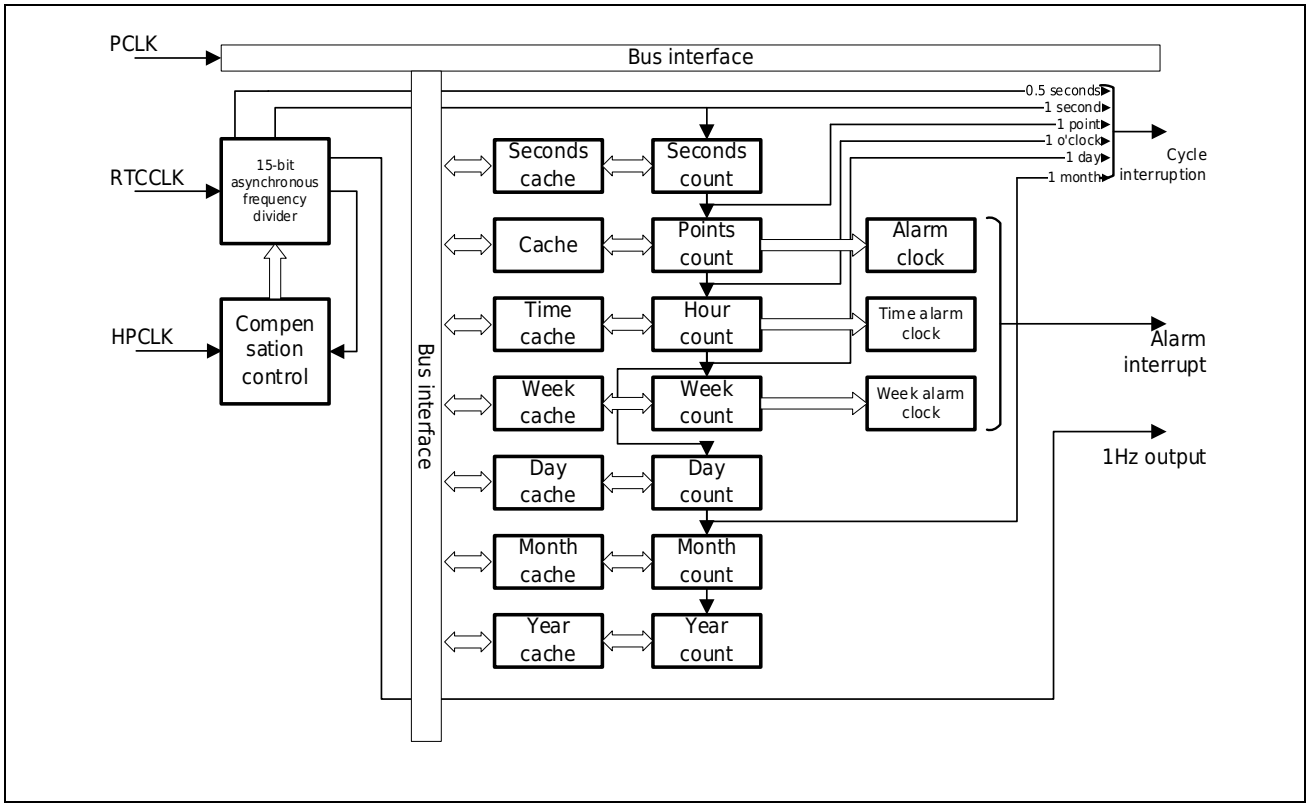


Figure 24-1 Basic block diagram of RTC

24.3.2 Power-on Setting

The RTC register value can be reset by power-on reset or setting RTC_CR0.RESET. After setting the control register, calendar initial value, alarm setting, etc., the RTC is started. After the RTC is started, other external reset requests cannot reset the RTC, and the RTC will always be in the working state. You can set the RTC_CR1.START bit of the control register to 0 to stop the RTC operation. When setting the RTC, the clock source must be stable.

24.3.3 RTC Count Start Setting

1. 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.
2. Set RTC_CR1.START=0, stop counting;
3. Set the system clock register, turn on the external low-speed oscillator, then set RTC_CR3 to select the RTC count clock source;
4. Set RTC_CR1, set time, period, 1Hz clock output;

5. Set the calendar counting registers for seconds, minutes, hours, weeks, days, months, and years;
6. When clock error compensation is required, set the counting clock error compensation registers RTC_ERRCRL, RTC_ERRCRH;
7. Clear the flag register bit in the register RTC_CR2 and enable the interrupt;
8. Set RTC_CR1.START=1, counting starts.

24.3.4 System Low Power Mode Switching

When the system switches to the low Power Consumption mode immediately after the RTC counting starts, please perform any of the following confirmations before switching the mode.

1) After setting RTC_CR1.START=1, switch modes after more than 2 RTC count clocks.
2) 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.

24.3.5 Read Count Register

1. After setting RTC_CR1.START=1, after more than 2 RTC count clocks, set RTC_CR2.RWREQ=1 to make a calendar register read request;
2. Query until RTC_CR2.RWEN=1;
3. Read all or part of the seconds, minutes, hours, weeks, days, months, and years counter register values;
4. Set RTC_CR2.RWREQ=0;
5. Query until RTC_CR2.RWEN=0.

24.3.6 Write Count Register

1. After setting RTC_CR1.START=1, after more than 2 RTC count clocks, set RTC_CR2.RWREQ=1 to make a calendar register write request;
2. Query until RTC_CR2.RWEN=1;
3. Write all or part of the seconds, minutes, hours, weeks, days, months, and years counter register values;
4. Set RTC_CR2.RWREQ=0. Note that all write operations must be completed within 1 second;
5. Query until RTC_CR2.RWEN=0.

24.3.7 Alarm Clock Setting

1. Set RTC_CR2.ALME=0, the alarm clock is disabled;
2. Set RTC_CR2.ALMIE=1, the alarm clock interruption permission;
3. Minute alarm clock RTC_ALMMIN, hour alarm clock RTC_ALMHOUR, week alarm clock RTC_ALMWEEK setting;
4. Set RTC_CR2.ALME=1, alarm clock permission;
5. Wait for the alarm to interrupt;
6. When an alarm occurs, RTC_CR2.ALMF=1, and the alarm interrupt processing begins.

24.3.8 Clock Error Compensation

Since the external low-speed crystal oscillator has deviations under various temperature conditions, the error needs to be compensated when high-precision counting results are required. Refer to Compensation Method-When Clock Error Compensation, Set The Counting Clock Error Compensation Registers RTC_ERRCRL, RTC_ERRCRH;.

24.3.9 1Hz Output

The RTC can output a 1Hz clock and provide three precision output modes: the first is a normal precision 1Hz output without clock compensation; the second is a distributed compensation 1Hz output with average compensation within every 32 seconds; and the third is a uniform compensation 1Hz output with compensation every second. When the clock error compensation function is valid RTC_ERRCRH.COMPEN=1, the distributed compensation 1Hz output and the uniform compensation 1Hz output can be selected. Of which,

The 1Hz output of ordinary precision is set as follows:

1. Set RTC_CR0.RESET=0, confirm that the RESET bit is 0, then set RTC_CR0.RESET=1 to reset the calendar count register;
2. Set RTC_CR1.START=0, counting stops;
3. 1Hz output pin setting;
4. RTC_CR1.ONEHZOE=1, clock output permission;
5. Set RTC_CR1.START=1, start counting;
6. Wait for more than 2 count cycles;
7. 1Hz output starts.

The distributed compensation 1Hz output setting is as follows:

1. Set RTC_CR0.RESET=0, confirm that the RESET bit is 0, then set RTC_CR0.RESET=1 to reset the calendar count register;
2. Set RTC_CR1.START=0, counting stops;
3. 1Hz output pin setting;
4. RTC_CR1.ONEHZOE=1, clock output permission;
5. Clock error compensation register RTC_ERRCRL.COMP[7:0] and RTC_ERRCRH.COMP[8] compensation number setting;
6. Clock error compensation register RTC_ERRCRH.COMPEN=1, error compensation is valid;
7. Set RTC_CR1.START=1, start counting;
8. Wait for more than 2 count cycles;
9. 1Hz output starts.

Uniform compensation 1Hz output setting is as follows:

1. Set RTC_CR0.RESET=0, confirm that the RESET bit is 0, then set RTC_CR0.RESET=1 to reset the calendar count register;
2. Set RTC_CR1.START=0, counting stops;
3. RTC output pin setting;
4. RTC_CR1.ONEHZOE=1, clock output permission;
5. RTC_CR1.ONEHZSEL=1, select output uniform compensation 1Hz clock;
6. Clock error compensation register RTC_ERRCRL.COMP[7:0] and RTC_ERRCRH.COMP[8] compensation number setting;
7. Clock error compensation register RTC_ERRCRH.COMPEN=1, accuracy compensation is valid;
8. Set RTC_CR1.START=1, start counting;
9. Wait for more than 2 count cycles;
10. 1Hz output starts.

24.4Interrupt

RTC supports two types of interrupts: timer alarm interrupt and fixed period interrupt.

24.4.1 Alarm Interrupt

Alarm clock interrupt RTC_ALM, when ALMIE=1 of the control register 2 (RTC_CR2) and ALME=1 of the control register 2 (RTC_CR2), if the current calendar time and the minute alarm clock register (RTC_ALMMIN), the hour alarm clock register (RTC_ALMHOUR), and the weekly alarm clock register (RTC_ALMWEEK) equal, trigger the alarm interrupt. The alarm is configured with an independent flag register bit, RTC_CR2.ALMF. Writing 0 to the RTC_CR2.ALMF bit clears the alarm flag.

24.4.2 Periodic Interrupt

Periodic interrupt RTC_PRD, when PRDIE=1 of the control register 2 (RTC_CR2), after the selected period occurs, a regular period wake-up interrupt is triggered. The fixed-cycle interrupt is configured with an independent flag register RTC_CR2.PRDF, and the fixed-cycle flag can be cleared by writing 0 to the RTC_CR2.PRDF bit.

24.5Register Description

Table 24-2 RTC Base Address

Name	Base Address	Description
RTC	0x400 4C000	RTC Base Address

Table 24-3 RTC Register List

Register	Description	Offset Address	Bit Width	Reset Value
RTC_CR0	Control register 0	0x0000	8	0x00
RTC_CR1	Control register 1	0x0004	8	0x00
RTC_CR2	Control register 2	0x0008	8	0x00
RTC_CR3	Control register 3	0x000C	8	0x00
RTC_SEC	Second Count Register	0x0010	8	0x00
RTC_MIN	Minute counter register	0x0014	8	0x00
RTC_HOUR	Time counter register	0x0018	8	0x12
RTC_WEEK	Week Count Register	0x001C	8	0x00
RTC_DAY	Day Count Register	0x0020	8	0x00
RTC_MON	Month Count Register	0x0024	8	0x00
RTC_YEAR	Year Count Register	0x0028	8	0x00
RTC_ALMMIN	Minute alarm register	0x002C	8	0x00
RTC_ALMHOUR	Time alarm register	0x0030	8	0x12
RTC_ALMWEEK	Weekly Alarm Register	0x0034	8	0x00
RTC_ERRCRH	Clock Error Compensation Register	0x0038	8	0x00
RTC_ERRCRL	Clock Error Compensation Register	0x003C	8	0x20

24.5.1 Control Register 0 (RTC_CR0)

Reset Value: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
Res							RESET

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b7~b1	Res	Reserved Bit	Read at “0”, write at “0”	RW
b0	RESET	RTC calendar counter reset	Write status 0: The initialization register is invalid 1: The initialization register is valid Initialize all RTC registers. Read status 0: Normal counting status or RTC software reset ends 1: RTC is in reset state	RW

24.5.2 Control Register 1 (RTC_CR1)

Reset Value: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
START	ONEHZSEL	ONEHZOE	Res	AMPM	PRDS[2:0]		

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b7	START	RTC counting starts	0: RTC count stop 1: RTC counting starts	RW
b6	ONEHZSEL	1Hz output selection	0: Distributed compensation 1Hz output 1: Uniform compensation 1Hz output **Note:** When RTC_ERRCRH.COMPEN=1, this bit setting is valid.	RW
b5	ONEHZOE	1Hz output enabled	0: 1Hz output disabled 1: 1Hz output permission	RW
b4	Res	Reserved Bit	Read at “0”, write at “0”	RW
b3	AMPM	Time selection	0:12 hour clock 1: 24-hour time format	RW
b2~0	PRDS[2:0]	Cycle Interrupt Selection	Cycle selection settings: PRDS[2] PRDS[1] PRDS[0] Cycle selection 0 0 0 Do not choose 0 0 1 Every 0.5 second cycle 0 1 0 Every 1 second cycle 0 1 1 Every 1 minute cycle 1 0 0 Every 1 hour cycle 1 0 1 Every 1 day cycle (every day at 00:00:00:00) 1 1 X Every January cycle (00:00:00 on the 1st of each month **Note:** When writing cycle selection during START=1 counting, please disable the cycle interrupt permission to prevent malfunction. And the relevant flag should be cleared after writing.	RW

24.5.3 Control Register 2 (RTC_CR2)

Reset Value: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
ALME	ALMIE	PRDIE	Res	ALMF	PRDF	RWEN	RWREQ

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b7	ALME	Alarm function enable	0: Alarm clock function disabled 1: Alarm clock function permission	RW
b6	ALMIE	Alarm clock interrupt enable	0: Alarm interrupt disabled 1: Alarm interrupt permission	RW
b5	PRDIE	Periodic interrupt enable	0: Periodic interrupt disabled 1: Periodic interrupt permission	RW
b4	Res	Reserved Bit	Read at “0”, write at “0”	RW
b3	ALMF	Alarm clock sign	0: Alarm does not match 1: Alarm match **Note:** It valid when ALME=1. When the alarm matches, it is set to "1" after one count clock. Writing "0" clears the flag, and writing "1" invaild.	RW
b2	PRDF	Cycle sign	0: The cycle does not occur 1: Periodic occurrence **Note:** After the set cycle occurs, this position is set to "1". Writing "0" clears the flag bit, writing "1" invaild	RW
b1	RWEN	Read/write enable	0: Read/write disabled 1: 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 seconds, minutes, hours, weeks, days, months, and year counting registers.	RW
b0	RWREQ	Read/write request	0: normal counting mode 1: 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" ". If RWREQ is set but not cleared within 1 second, the counter will stop counting and the real-time clock RTC will not be able to accurately time. If the RWREQ bit remains set, the real-time clock RTC will stop timing.	RW

24.5.4 Control Register 3 (RTC_CR3)

Reset Value: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
RCKSEL	Res		LRCEN	Res			

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b7	RCKSEL	RTC count clock selection	0: Select the external low-speed oscillator XTAL32 clock 1: Select the internal low-speed oscillator LRC clock or XTAL fractional frequency clock **Note:** When RCKSEL=1, the RTC count clock is determined by RTCCKSEL[1:0] in the PWC_PWRC6 register.	RW
b6-b5	Res	Reserved Bit	Read at “0”, write at “0”	RW
b4	LRCEN	Internal low speed oscillator enabled	0: Internal low speed oscillator LRC is stopped 1: Internal low speed oscillator LRC works **Note:** When the low-speed oscillator is used as the RTC clock source, please set the LRCEN bit to enable it.	RW
b3-b0	Res	Reserved Bit	Read at “0”, write at “0”	RW

24.5.5 Second Count Register (RTC_SEC)

Reset Value: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
Res	SECD[2:0]			SECU[3:0]			

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b7	Res	Reserved Bit	Read at “0”, write at “0”	RW
b6~b4	SECD[2:0]	Tens of seconds	Second tens digit count value	RW
b3~b0	SECU[3:0]	Seconds	Second count value	RW

Note:

Indicates 0-59 seconds, using decimal counting. Please write the BCD code of decimal 0-59. When writing an incorrect value, the written value will be ignored.

24.5.6 Minute Count Register (RTC_MIN)

Reset Value: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
Res	MIND[2:0]			MINU[3:0]			

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b7	Res	Reserved Bit	Read at “0”, write at “0”	RW
b6~b4	MIND[2:0]	Tens	Minute and tens count value	RW
b3~b0	MINU[3:0]	Minutes	Minute count value	RW

Note:

Indicates 0-59 points, using decimal counting. Please write the BCD code of decimal 0-59. When writing an incorrect value, the written value will be ignored.

24.5.7 Hour Count Register (RTC_HOUR)

Reset Value: 0x12

b7	b6	b5	b4	b3	b2	b1	b0
Res		HOURD[1:0]		HOURU[3:0]			

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b7~b6	Res	Reserved Bit	Read at “0”, write at “0”	RW
b5~b4	HOURD[1:0]	Time ten	Hour tens count value	RW
b3~b0	HOURU[3:0]	When ones	Hour count value	RW

Note:

1. In 24-hour format, it means 0-23 hours. In 12-hour format, b5=0 means AM, then 01~12 means morning; b5=1 means PM, then 21~32 means afternoon.
2. Please set the correct decimal 0~23 or 01~12, 21~32 BCD code according to the value of the control bit AMPM. Writing a value out of range will be ignored.

Please refer to the following table for the specific time:

24 hour clock	**AMPM=1**	**12 hour clock**	**AMPM=0**
time	**Register representation**	**time**	**Register representation**
SYNCMD[2]	00H	AM 12:00	12H
01 o'clock	01H	AM 1:00	01H
02 o'clock	02H	AM 2:00	02H
03 o'clock	03H	AM 3:00	03H
04 o'clock	04H	AM 4:00	04H
05 o'clock	05H	AM 5:00	05H
06 o'clock	06H	AM 6:00	06H
07 o'clock	07H	AM 7:00	07H
08 o'clock	08H	AM 8:00	08H
09 o'clock	09H	AM 9:00	09H
10 o'clock	10H	AM 10:00	10H
11 o'clock	11H	AM 11:00	11H
12 o'clock	12H	PM 12:00	32H
13 o'clock	13H	PM 1:00	21H
14 o'clock	14H	PM 2:00	22H
15 o'clock	15H	PM 3:00	23H
16 o'clock	16H	PM 4:00	24H
17 o'clock	17H	PM 5:00	25H
18 o'clock	18H	PM 6:00	26H
19 o'clock	19H	PM 7:00	27H

24 hour clock	AMPM=1	12 hour clock	AMPM=0
time	Register representation	time	Register representation
20 o'clock	20H	PM 8:00	28H
21 o'clock	21H	PM 9:00	29H
22 o'clock	22H	PM 10:00	30H
23 o'clock	23H	PM 11:00	31H

24.5.8 Day Count Register (RTC_DAY)

Reset Value: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
Res		DAYD[1:0]		DAYU[3:0]			

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b7~b6	Res	Reserved Bit	Read at “0”, write at “0”	RW
b5~b4	HDAYD[1:0]	Day tens digit	Day tens digit count value	RW
b3~b0	HDAYU[3:0]	Day units digit	Day units digit value	RW

Note:

Decimal means 1~31 days, automatically calculate leap year and month. The specific expression is as follows:

Months	**Day count display**
February (normal year)	01~28
February (leap year)	01~29
April, June, September, November	01~30
1, 3, 5, 7, 8, 10, 12 months	01~31

24.5.9 Week Count Register (RTC_WEEK)

Reset Value: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
Res					WEEK[2:0]		

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b7~b3	Res	Reserved Bit	Read at “0”, write at “0”	RW
b2~b0	WEEK[2:0]	Week	Week count	RW

Note:

Decimal 0~6 means Sunday~Saturday. Please write the BCD code of decimal 0~ 6. writing an incorrect value, the will be ignored. The corresponding relationship between the week count value is as follows:

week	**Week count**
Sunday	0x00
Monday	0x01
Tuesday	0x02
Wednesday	0x03
Thursday	0x04
friday	0x05
Saturday	0x06

24.5.10 Month Count Register (RTC_MON)

Reset Value: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
Res			MON[4:0]				

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b7~b5	Res	Reserved Bit	Read at “0”, write at “0”	RW
b4~b0	MON[4:0]	Month	Month count	RW

Note:

Decimal 1~12 means Sunday~ 12 January. Please write in the correct decimal 1~12 BCD code, other values will be ignored.

24.5.11 Year Count Register (RTC_YEAR)

Reset Value: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
YEARD[3:0]				YEARU[3:0]			

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b7~b4	YEARD[3:0]	Year tens digit	Year tens digit count value	RW
b3~b0	YEARU[3:0]	Year unit digit	Day unit digit value	RW

Note:

Decimal 0~99 means 0~99 years. Count according to the monthly round. Automatically calculate leap years such as: 00, 04, 08, ..., 92, 96, etc. Please write correct decimal year count value, wrong value will be ignored.

24.5.12 Minute Alarm Register (RTC_ALMMIN)

Reset Value: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
Res	ALMMIND[2:0]			ALMMINU[3:0]			

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b7	Res	Reserved Bit	Read at “0”, write at “0”	RW
b6~b4	ALMMIND[3:0]	Alarm clock alarm tens digit	Alarm clock Alarm tens digit match Value	RW
b3~b0	ALMMINU[3:0]	Minute alarm clock	Alarm clock alarm unit digit matching Value	RW

Note:

Please set the BCD code of decimal 0~59. Write other values, no alarm match will occur.

24.5.13 Hour Alarm register (RTC_ALMHOUR)

Reset Value: 0x12

b7	b6	b5	b4	b3	b2	b1	b0
Res		ALMHOURD[1:0]		ALMHOURU[3:0]			

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b7~b6	Res	Reserved Bit	Read at “0”, write at “0”	RW
b5~b4	ALMHOURD[1:0]	Alarm clock tens digit	Alarm clock tens digit match Value	RW
b3~b0	ALMHOURU[3:0]	Alarm clock unit digit	Alarm clock unit digit match Value	RW

Note:

Please set the correct alarm clock matching value according to the time system, otherwise the alarm clock matching will not happen.

24.5.14 Week Alarm Register (RTC_ALMWEEK)

Reset Value: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
Res	ALMWEEK[6:0]						

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b7	Res	Reserved Bit	Read at “0”, write at “0”	RW
b6~b0	ALMWEEK[6:0]	Week alarm clock	Week alarm match value. b0~b6 correspond to Sunday~Saturday respectively, and when they are set to "1", it means that the alarm clock is valid on that day of the week. For example, b0=1, b5=1 means that the Sunday and Friday alarm settings are valid.	RW

Note:

Please set the correct alarm clock matching value according to the time system, otherwise the alarm clock matching will not happen.

24.5.15 When Clock Error Compensation, Set The Counting Clock Error Compensation Registers RTC_ERRCRL, RTC_ERRCRH;

Reset Value: 0x00

RTC_ERRCRH

b7	b6	b5	b4	b3	b2	b1	b0
COMPEN	Res						COMP[8]

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b7	COMPEN	Compensation enable	0: Clock error compensation invalid 1: Clock error compensation valid	RW
b6~b1	Res	Reserved Bit	Read at “0”, write at “0”	RW
b0	COMP[8]	Compensation value	Set the compensation value together with COMP[7:0]	RW

Reset Value: 0x20

RTC_ERRCRL

b7	b6	b5	b4	b3	b2	b1	b0
COMP[7:0]							

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b7~b0	COMP[7:0]	Compensation value	By setting the compensation value, an accuracy compensation of +/-0.96ppm can be performed per second. The compensation value is 9-bit 2's complement code with a decimal point, and the last 5 bits are the decimal part. Compensable range -275.5ppm~+212.9ppm. The minimum resolution is 0.96ppm. Please refer to the following table for specific compensation accuracy:	RW

Compensation value setting										Compensation
COMPEN	COMP[8:0]									
1	1	0	0	0	0	0	0	0	0	-275.5ppm
	1	0	0	0	0	0	0	0	1	-274.6ppm
	~	~	~	~	~	~	~	~	~	~
	0	0	0	0	0	0	0	0	0	-30.5ppm
	~	~	~	~	~	~	~	~	~	~
	0	0	0	0	1	1	1	1	1	-0.96ppm
	0	0	0	1	0	0	0	0	0	0ppm
	0	0	0	1	0	0	0	0	1	+0.96ppm
	~	~	~	~	~	~	~	~	~	~
	0	0	1	0	0	0	0	0	0	+30.5ppm
	~	~	~	~	~	~	~	~	~	~
	0	1	1	1	1	1	1	1	0	+212.0ppm
	0	1	1	1	1	1	1	1	1	+212.9ppm
0	X	X	X	X	X	X	X	X	X	Invalid

Compensation calculation instructions:

When the 1Hz clock is directly output in the default state, the compensation target value is calculated by measuring the accuracy of the clock.

Assuming that the actual measured value is 0.9999888Hz, then:

Actual oscillation frequency = 32768 × 0.9999888 ≈ 32767.63

Compensation target value = (Actual oscillation frequency – Target frequency)/Target frequency×10^6

= (32767.63–32768)/ 32768×10^6

= - 11.29ppm

Setpoint calculation:

If the compensation target value is +20.3ppm, calculate the corresponding register value as follows:

COMP[8:0] = $(20.3\times2^{15}/10^6)_{\text{Take 2's complement}}$ + 0b0001.00000

= $(0.6651904)_{\text{Take 2's complement}}$ + 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] = $(-20.3\times2^{15}/10^6)_{\text{Take 2's complement}}$ + 0b0001.00000

= $(-0.6651904)_{\text{Take 2's complement}}$ + 0b0001.00000

= 0b1111.01011 + 0b0001.00000

= 0b0000.01011

25 Watchdog Counter (WDT/SWDT)

25.1 Introduction

This product has two watchdog counters, one is a special watchdog counter (SWDT) whose counting clock source is a dedicated internal RC (SWDTLRC: 10KHz), and the other is a universal watchdog counter (WDT) whose counting clock source is PCLK3. The special watchdog and universal watchdog are 16-bit down counters used to monitor software failures due to external disturbances or unforeseen logic conditions that deviate from the normal operation of the application program.

Both watchdogs support window functions. The window interval can be preset before the count starts, and when the count value is within the window interval, the counter can be refreshed and the count restarted. Basic characteristics such Table 25-1.

Table 25-1 Basic Characteristics Of Watchdog Counter

Counting clock	SWDT: 1/16/32/64/128/256/2048 frequency division of SWDTLRC WDT: 4/64/128/256/512/1024/2048/8192 division of PCLK3
Maximum overflow time	SWDT: 3.72 hours (SWDTLRC: 10KHz) WDT: 10.7 seconds (PCLK3=50MHz)
Counting mode	Decrement count
Window function	Can set window interval, define the allowable interval of refresh action
Start-up mode	1) Hardware startup 2) Software startup
Stop condition	1) Reset 2) Underflow, refresh error occurs Restart: Automatically starts after resetting or terminating the request output in hardware startup mode In software startup mode, set refresh register again
Interrupt/Reset Conditions	1) Count underflow 2) Refresh error

25.2Functional Description

25.2.1 Start the Watchdog

Watchdog counter can be started in two ways: Hardware and software.

The hardware startup mode refers to reading the watchdog counter setting information (ICG0register) from the main flash memory area during startup, and the counter starts counting automatically. Software startup means that after setting control register, write refresh register completes the refresh action, counter starts counting.

25.2.2 Hardware Startup Mode

When the bit 16 (WDTAUTS) and bit 0 (SWDTAUTS) of the ICG0 register are 0, it is the hardware startup mode. When the hardware startup mode is selected, the related setting information of the WDT_CR and SWDT_CR registers invalid.

In hardware startup mode, the WDT/ SWDT related settings (counting clock, window setting, counting period, etc.) in ICG0register are loaded into the WDT/ SWDT module during reset. After reset, the counter starts counting automatically according to the setting. Figure 25-1 is an example of the operation of the hardware startup method.

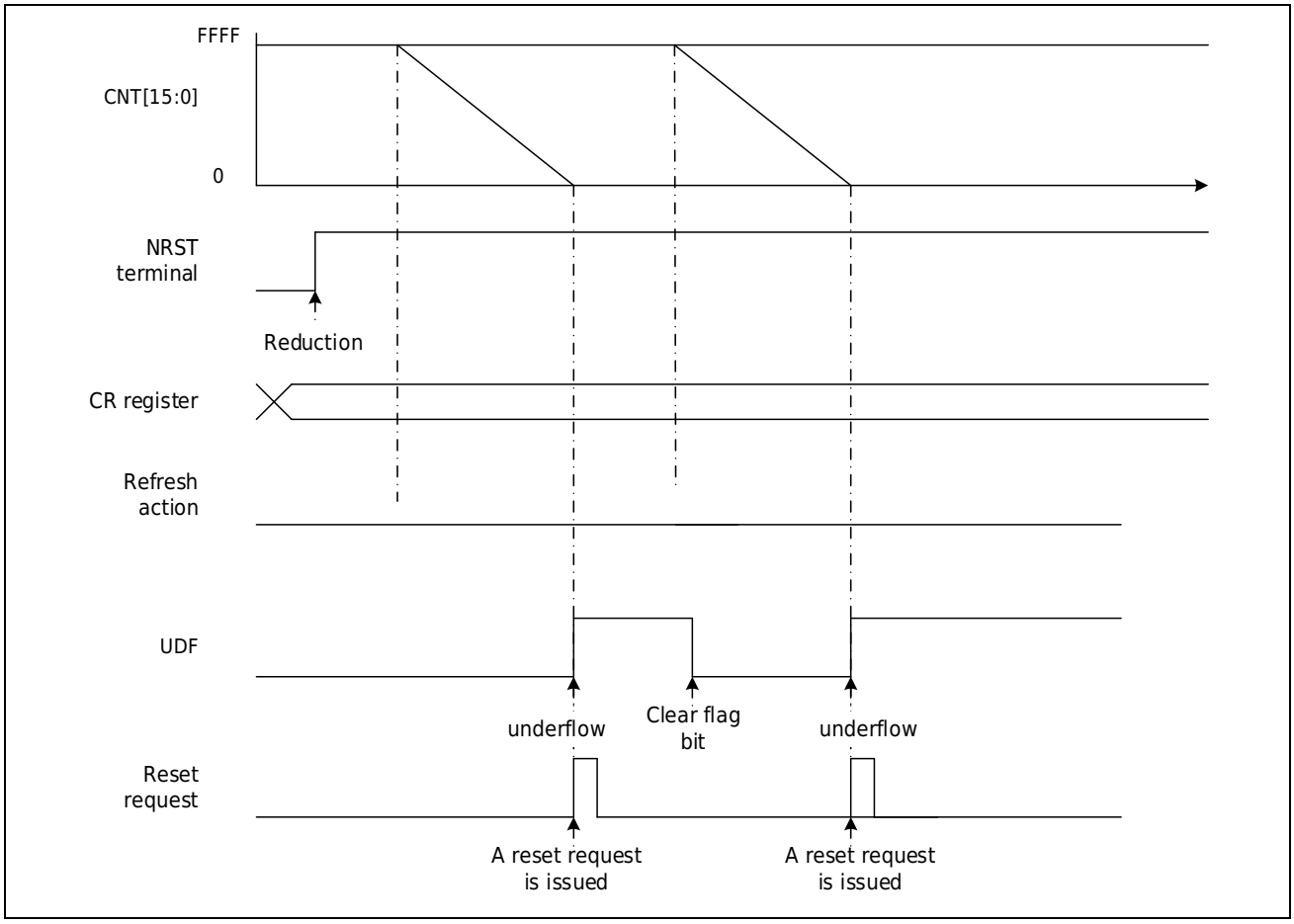


Figure 25-1 Hardware Startup Example

25.2.3 Software Startup Mode

When bit 16 (WDTAUTS) and bit 0 (SWDTAUTS) of the ICG0 register are 1, the WDT/SWDT is started by setting the refresh register as software start mode. After reset, set the count clock, window setting value, count period, etc. in the WDT_CR/SWDT_CR register, and then execute the refresh action, and the counter starts counting. WDT_CR/SWDT_CR can only be set once, and the written value will be invalid if it is set again. Figure 25-2 is an example of the operation of the software startup method.

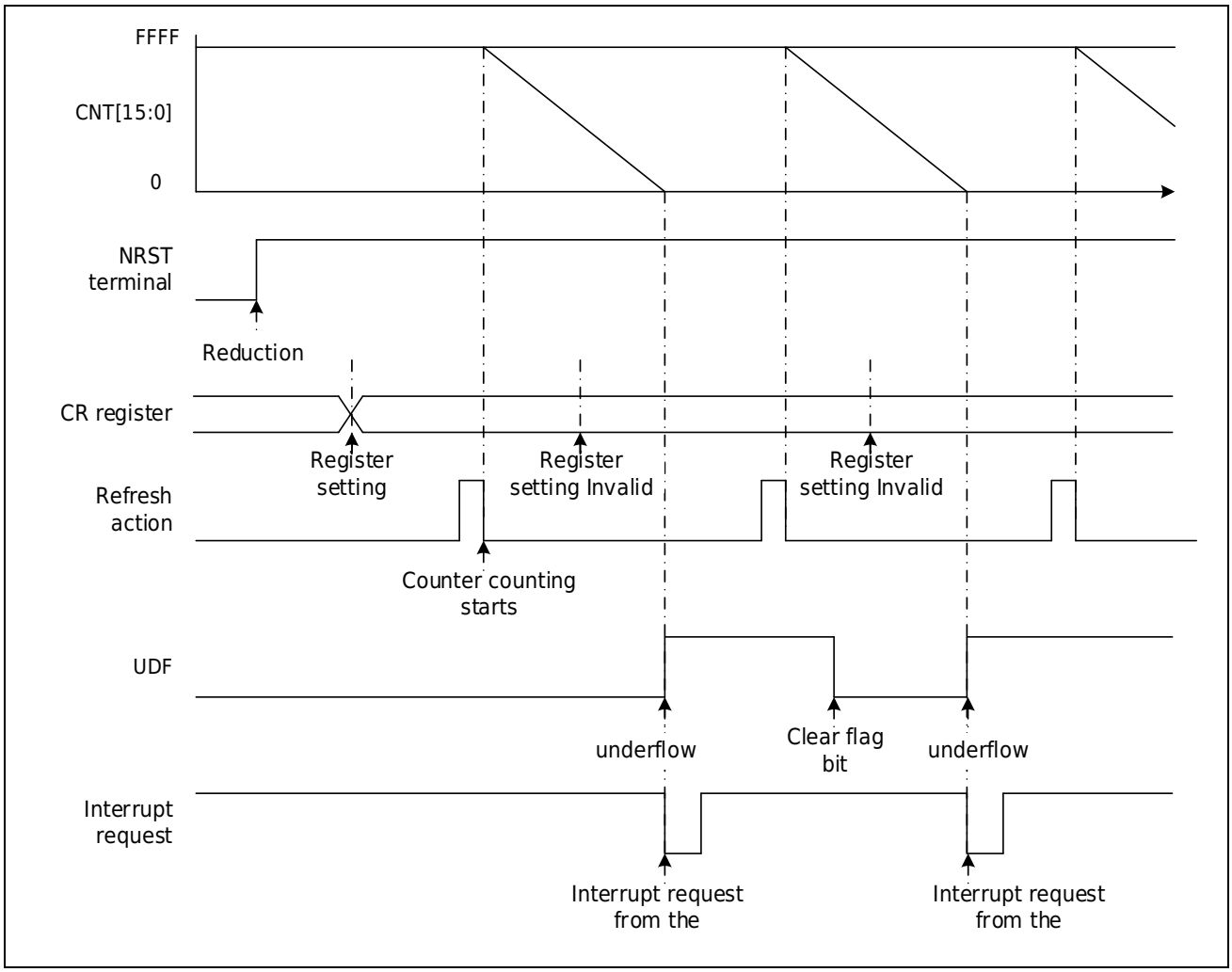


Figure 25-2 Software Startup Example

25.2.4 Refresh Action

First write 0x0123 and then write 0x3210 to the WDT/SWDT_RR register to complete a refresh action. The WDT/SWDT counter will start counting (software start) or restart counting.

When the WDT/SWDT_RR register is written to 0x0123 and 0x3210, if other registers are accessed or the WDT/SWDT_RR register is read, the normal refresh operation will not be affected asFigure 25-3 shown action.

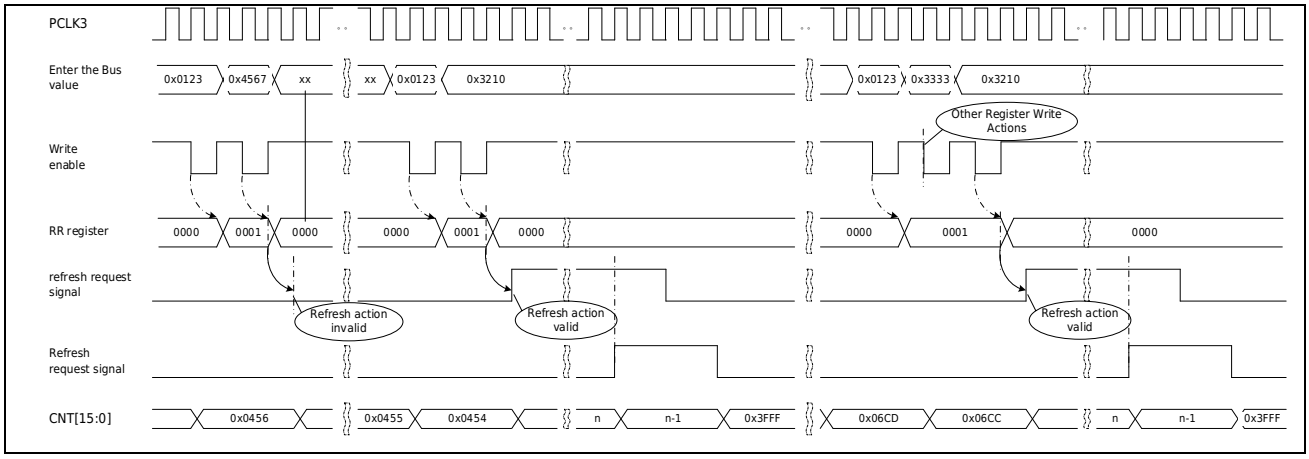


Figure 25-3 Examples of Refresh Action Timing (Action Confirmation, Refresh Request Signal Falling Edge, etc.)

The refresh action requires four count cycles to update the count values, so write the refresh register in advance of four count values in the refresh lower window and underflow position. Check the status register for the count value.

25.2.5 Flag Bit

The refresh error flag and count underflow flag are maintained in the case of an interrupt request. When an interrupt occurs, the cause of the interrupt can be confirmed by querying the flag bit. Clear the flag bit: read 1 first and then write 0.

When the refresh error or count underflow flag is set, the watchdog count does not stop in hardware startup mode; the watchdog count stops in software startup mode. When writing 0 to the flag bit, SWDT needs at most 3 SWDTLRC and 2 PCLK3 times before the register bit can be cleared; WDT needs at most 5 PCLK3 times before the register bit can be cleared. In addition, within a certain period of time when a refresh error or underflow error occurs, reading 1 to clear the flag bit is invalid. This period of time is: 1 count cycle + 2 SWDTLRC (SWDT module); 1 count cycle + 2 PCLK3 (WDT module).

25.2.6 Interrupt Reduction

WDT/SWDT can generate an interrupt request or a reset request when the counter underflows or refreshes an error. In hardware startup mode, the WDTITS/SWDTITS bit of ICG0 is used to determine whether an interrupt request or a reset request is generated. In software startup mode, the ITS bit in the WDT_CR/SWDT_CR register is set to determine whether an interrupt request or a reset request is generated.

There are two conditions for generating WDT/SWDT interrupt reset. One is that counter counts produce underflow; One is to perform a refresh action outside the refresh allowed interval, resulting in a refresh error.

25.2.7 Count Underflow

For Figure 25-4 example, underflow occurs when the decrement counts to zero as shown in Examples of counter underflow.

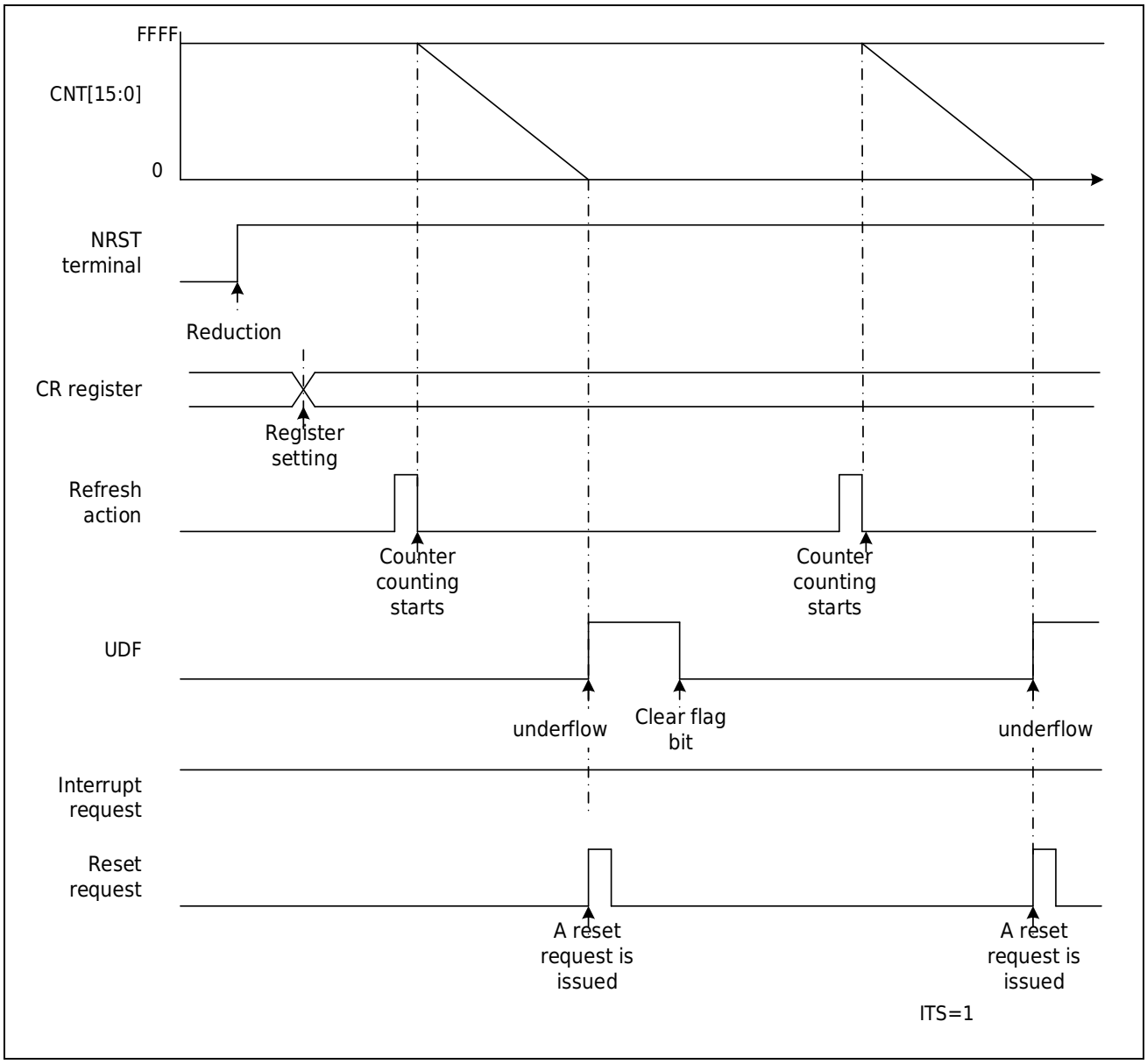


Figure 25-4 Counter Underflow Action Example

25.2.8 Refresh Error

After the window interval is set, the counter is refreshed and counted again only when the refresh action is executed in the window interval. A refresh error is generated when the refresh action is executed outside the window interval. Figure 25-5 Refresh action

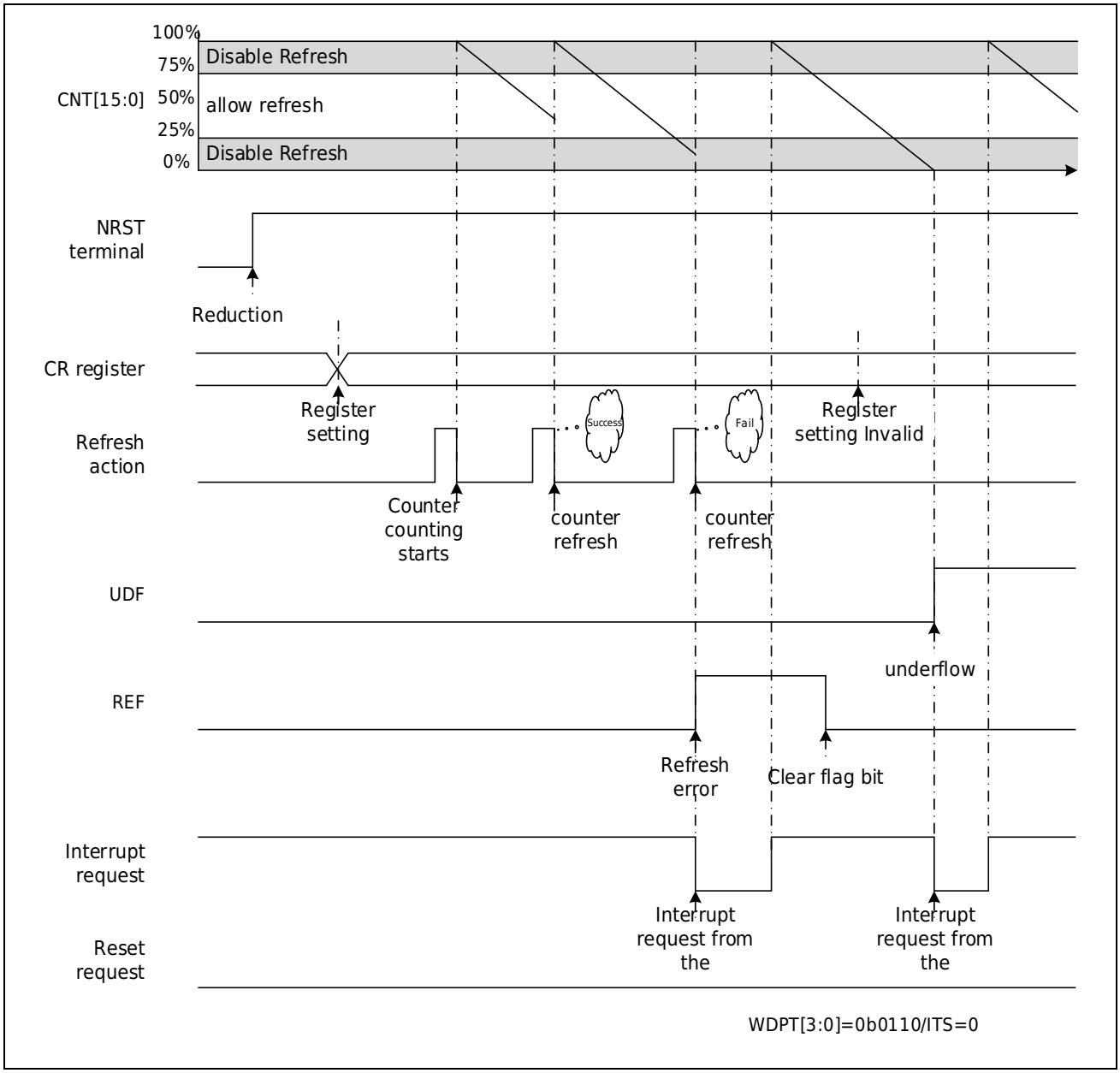


Figure 25-5 Counter Refresh Action Example

25.3Register Description

Table 25-2 WDT/SWDT Base Address

Name	Base Address	Description
WDT	0x4004 9000	WDT Base Address
SWDT	0x4004 9400	SWDT Base Address

Table 25-3 WDT Register List

Register	Description	Offset Address	Bit Width	Reset Value
WDT_CR	WDT control register	0x00	32	0x8001 0FF3
WDT_SR	WDT status register	0x04	32	0x0000 0000
WDT_RR	WDT Refresh register	0x08	32	0x0000 0000

Table 25-4 SWDT Register List

Register	Description	Offset Address	Bit Width	Reset Value
SWDT_CR	SWDT control register	0x00	32	0x8001 0FF3
SWDT_SR	SWDT status register	0x04	32	0x0000 0000
SWDT_RR	SWDT Refresh register	0x08	32	0x0000 0000

25.3.1 Control Registers (SWDT_CR, WDT_CR)

Reset value: 0x8001 0FF3

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
ITS	Res														SLPOFF
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res				WDPT				CKS[3:0]				Res		PERI[1:0]	

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31	ITS	Refresh Error/Overflow Interrupt/Reset Selection	0: Interrupt request 1: Reset request	RW
b30~b17	Res	Reserved Bit	Read at “0”, write at “0”	RW
b16	SLPOFF	Counting disabled in low power mode	WDT_CR: 0: WDT count permission in Sleep Mode 1: WDT Prohibition counting in Sleep Mode SWDT_CR: 0: SWDT count permission in Sleep/Stop Mode 1: SWDT count disabled in Sleep/Stop Mode	RW
b15~b12	Res	Reserved Bit	Read at “0”, write at “0”	RW
b11~b8	WDPT[3:0]	Refresh Allowed Area Count%	0000：0%~100% 0001：0%~25% 0010：25%~50% 0011：0%~50% 0100：50%~75% 0101：0%~25%，50%~75% 0110：25%~75% 0111：0%~75% 1000：75%~100% 1001：0%~25%，75%~100% 1010：25%~50%，75%~100% 1011：0%~50%，75%~100% 1100：50%~100% 1101：0%~25%，50%~100% 1110：25%~100% 1111：0%~100%	RW
b7~b4	CKS[3:0]	Counting clock	WDT_CR: 0010：PCLK3/4 0110：PCLK3/64 0111：PCLK3/128 1000：PCLK3/256 1001：PCLK3/512 1010：PCLK3/1024 1011：PCLK3/2048 1101：PCLK3/8192 Remaining values: Reserved function SWDT_CR: 0000：SWDTCLK 0100：SWDTCLK/16 0101：SWDTCLK/32	RW

			0110：SWDTCLK/64 0111：SWDTCLK/128 1000：SWDTCLK/256 1011：SWDTCLK/2048 Other values: Reserved	
b3~b2	Res	Reserved Bit	Read at “0”, write at “0”	RW
b1~b0	PERI[1:0]	Counting cycle	00：256 cycle 01：4096 cycle 10：16384 cycle 11：65536 cycle	RW

25.3.2 Status Register (SWDT_SR, WDT_SR)

Reset value: 0x0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res														REF	UDF
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
CNT[15:0]															

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b18	Res	Reserved Bit	Read at “0”, write at “0”	RW
b17	REF	Refresh error flag	0: No refresh error 1: Refresh error occurred to the alignment After reading 1 write 0 and reset.	RW
b16	UDF	Counting underflow mark	0: No count underflow 1: Occurrence count underflow to the alignment After reading 1 write 0 and reset.	RW
b15~b0	CNT[15:0]	Count value	Counter current count	R

25.3.3 Refresh Register (SWDT_RR, WDT_RR)

Reset value: 0x0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
RF[15:0]															

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b16	Res	Reserved Bit	Read at “0”, write at “0”	RW
b15~b0	RF[15:0]	Refresh value	Write 0x0123 and 0x3210 to complete the refresh. When 0x0123 is written to the register, the read value of the register is 0x0000 0001; in all other cases, the read value is 0x0000 0000.	RW

25.4Precautions

When SWDT is in operation, the operating frequency of the peripheral clock PCLK3 must be greater than or equal to 4 times the counting clock frequency, that is, PCLK3 frequency ≥ counting clock frequency x4.

26 Universal Synchronous/ Asynchronous Receiver Transmitter (USART)

26.1 Introduction

This product is equipped with 4 units of Universal Synchronous Asynchronous Receiver/sendter (USART) modules. It 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. Supports modem operation (CTS/RTS operation), multi-processor operation, and Driver Enable (DE) function. Cooperate with Timer0 module to support UART receive timeout function. USART_1 supports the Stop Mode wake-up function via the RX line.

26.2 Main Features

- UART: Full channel support
- Multiprocessor communications: full channel support
- Clock synchronous communication: full channel support
- RX line wake-up Stop Mode function: USART_1 support
- Fractional baud rate: all channels support
- LIN: Supported by USART_2, USART_3, USART_4
- Smart Card: USART_1 support
- UART receive timeout function: supported by USART_1, USART_2, USART_3, USART_4
- RS485 Driver Enable: supported by USART_1, USART_2, USART_3, USART_4

Key features of USART:

- Support full--duplex asynchronous communication, full-duplex clock synchronous communication
- Support LIN bus
- Support smart card interface (ISO/IEC7816-3)
- Transmitter and receiver have independent energy position
- Built-in double buffer
- LSB/MSB optional
- Supports modem operation (CTS/RTS)
- Transmission flag: send data register null, send data complete, receive data register full, receive error flag, UART receive timeout flag, LIN wake-up signal detection flag, LIN interval detection flag, LIN bus error flag
- Support RS485 Driver Enable function

Main Features of UART:

- Data length programmable: 8 bits/9 bits

- Configurable check function: Odd/even/no check
- Stop bit configurable: 1 bit/2 bit
- Timer optional: Internal timer (clock generated by internal baud rate generator)/External timer (clock entered by USARTn_CK pin)
- Reception error: Check error, frame error, overflow error
- Support for inter-processor communications
- Built-in digital filter
- Support receiving data TIMEOUT function
- Wakeup Unit 1 Stop Mode Function
- Support full-duplex/half-duplex communication

Main Features of Clock Synchronization Mode:

- Data length: 8 bits
- Reception error: Overflow error
- Clock source: internal clock source (clock generated by the internal baud rate generator)/external clock source (clock input by the USARTn_CK pin)
- Support full-duplex communication

Key Features of Smart Card Interface:

- Data length: 8 bits
- Support receiving data TIMEOUT function
- Support data retransmission

Main Features of LIN:

- Data length: 8 bits
- Detection of support wakeup signal
- Supports 10/11-bit synchronization interval (BF-Break Field) detection
- Supports synchronous segment measurement, register records measurement value
- Support 10/11/13/14-bit break segment (BF) transmission
- Support bus conflict detection
- Support loopback mode

26.3Functional Description

26.3.1 USART System Block Diagram

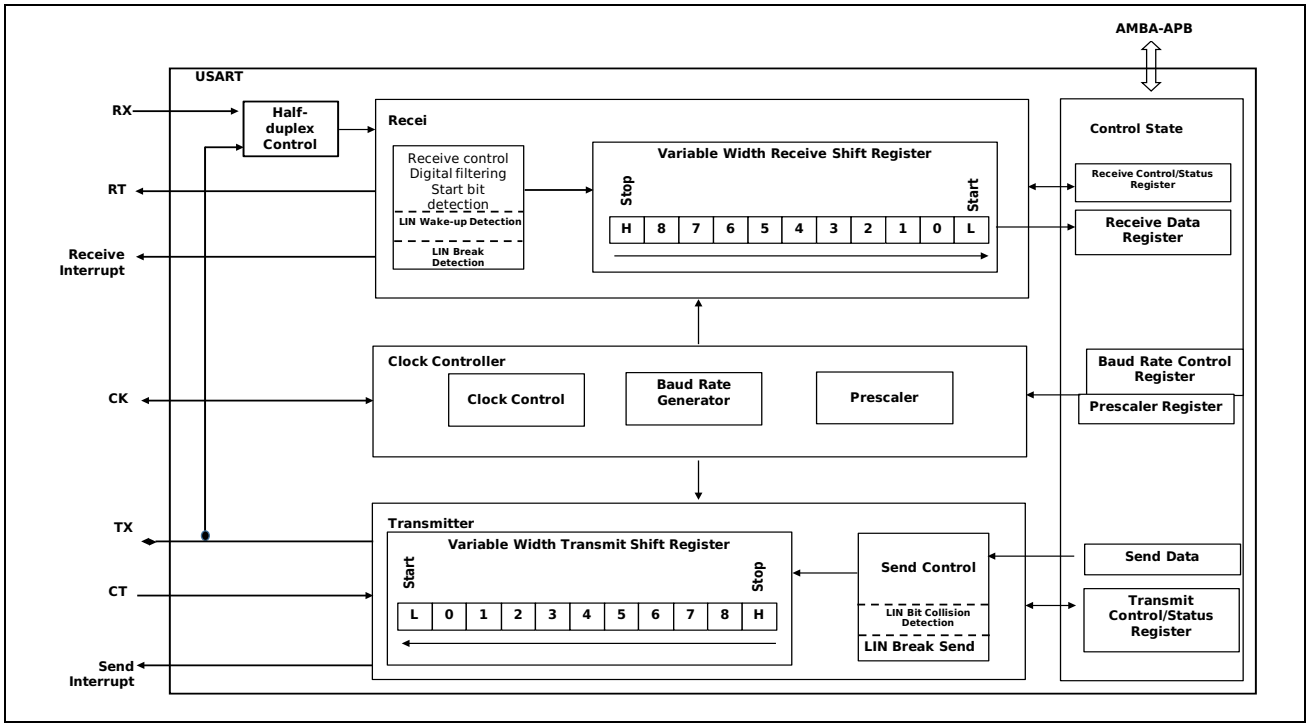


Figure 26-1 USART System Block Diagram

Table 26-1 USART Pin Descriptions

Pin Name	Direction	Functional Description
USARTn_CK	Input output	Clock
USARTn_TX	Input output	Send data pin It is also used as a receiving 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

n: 1~4

26.3.2 UARTFunction

26.3.2.1 Clock

UART can select the clock generated by the internal baud rate generator (internal clock source) or the clock input by the USARTn _ CK pin (external clock source) as the clock source for communication.

Internal Clock Source

USARTn_CR2.CLKC [1: 0] bit is set to 0b00 or 0b01, selects the internal clock source, the clock generated by the internal baud rate generator.

USARTn_CR2. CLKC [1: 0] = 0b00 , the USARTn _ CK pin is not used as a clock pin at and can be used as a common IO.

USARTn_CR2.CLKC [1: 0] = 0b01, the clock output from the USARTn _ CK pin is 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 setting of the USARTn_PR.PSC[1:0] bits as PCLK1, PCLK1/4, PCLK1/16, PCLK1/64. When USARTn_PR.ULBREN is 1, the ultra-low baud rate function is enabled, and the clock source of the internal baud rate generator is selected by the setting of the USARTn_PR.PSC[1:0] bits as PCLK1/128, PCLK1/256, PCLK1/512, PCLK1/1024.

External Clock Source

USARTn_CR2.CLKC [1: 0] bit is set to 0b10 or 0b11, the clock source is the external clock input from the USARTn_CK pin, and the input clock frequency is 16 times the baud rate (USARTn _ CR1.OVER8 = 0) or 8 times (USARTn _ CR1.OVER8 = 1).

Highest Baud Rate

For internal clock source, the Baud Rate formula generated by the internal Baud Rate Generator is:

B: Baud rate unit: MBps
C: When USARTn_PR.ULBREN is 0, the clock set by USARTn_PR.PSC[1:0] bits (PCLK1, PCLK1/4, PCLK1/16, PCLK1/64) Unit: MHz
When USARTn_PR.ULBREN is 1, the clock set by USARTn_PR.PSC[1:0] bits (PCLK1/128, PCLK1/256, PCLK1/512, PCLK1/1024) Unit: MHz
OVER8: USARTn_CR1.OVER8 setting
DIV _ Integer: USARTn _ BRR.DIV _ Integer setting

The highest baud rate is PCLK1/8 (Mbps).

When the clock source is external, the maximum frequency of the external input UART clock is required to be PCLK1/4, so when the clock source is an external input clock: when

USARTn_CR1.OVER8=0, the maximum baud rate is PCLK1/64 (Mbps); when USARTn_CR1.OVER8=1, the maximum baud rate is PCLK1/32 (Mbps).

It should be noted that in addition to the PCLK-based calculus described above, UART's highest communication baud rate needs to be referred to the highest communication baud rate specified in the **Data Sheet Electrical Characteristics** chapter.

26.3.2.2 Data Format

A frame of data in UART mode consists of start bits, data bits, check bits, and stop bits.

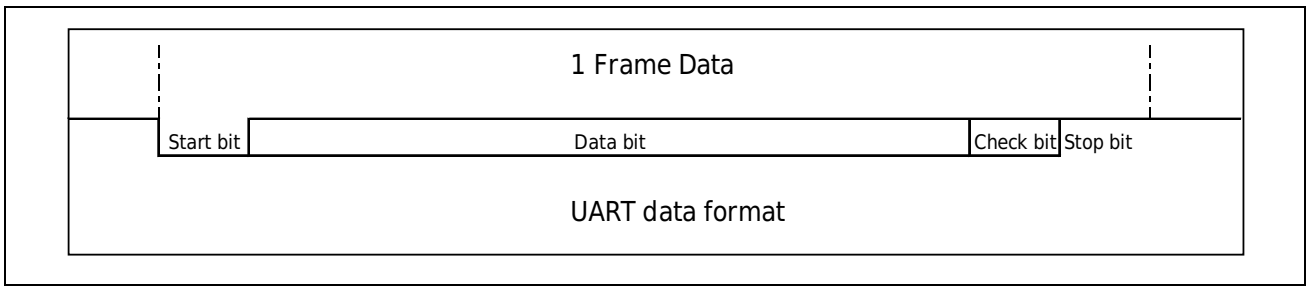


Figure 26-2 UART data format

Start Bit

The start bit is fixed with a low level of one bit.

Data Bit

Data bits can be configured as 8 bits or 9 bits.

Check Bit

The parity bits can be configured as 1-bit parity or 1-bit parity or no parity bits.

Stop Bit

Stops at a fixed high level and can be configured as 1 or 2 bits.

26.3.2.3 Modem Operation

Modem operations include CTS and RTS functions. When USARTn_CR3.RTSE=1, the RTS function is valid, and when USARTn_CR3.CTSE=1, the CTS function is valid.

CTS Function

The CTS function controls the sending of data through the input of the USARTn_CTS pin. The data can be sent only when the USARTn_CTS pin enters the low level. The data being sent is not affected if the USARTn_CTSpin enters the high level.

RTS Function

The RTS feature is a low level output from the USARTn_RTS pin to request the other party to send data.

The USARTn_RTS pin output low level needs to meet all of the following conditions:

- Receive enable (USARTn_CR1.RE=1) and not receiving data
- No unread received data in USARTn_RDR.RDR register
- No receive errors, including frame errors, check errors and overflow errors

26.3.2.4 Transmitter

The transmitter can send 8-bit or 9-bit data, depending on the setting value of the USARTn_CR1.M bit.

Transmitter enable bit (USARTn _ CR1.TE) 1. After writing the transmitted data, the transmitted data is serially output on the TX pin. The corresponding clock pulse can be output or not output at the USARTn _ CK pin.

The sequence of sending data is: Start bit - > Data bit (MSB/LSB) - > Check bit (with or without) - > Stop bit.

The transmission data registerTDR and the internal transmission shift register constitute a dual buffer structure, which can send data continuously.

To ensure the correctness of sending data, a request can only be written once by sending data register empty interrupt or DMA.

Sending Data Setting Procedure

1. Set the USARTn_CR1 register to the reset value;
2. Set the pins required for UART;
3. Select the clock source through USARTn_CR2.CLKC[1:0] bits;
4. Set USARTn_CR1, USARTn_CR2, USARTn_CR3 registers;
5. Set USARTn_PR to select the pre-scaling value, and USARTn_BRR register to set the communication baud rate (no need to set when the clock source is an external clock source);
6. Enable the sendter (USARTn_CR1.TE=1). If the send data register null interrupt is required, set USARTn_CR1.TXEIE=1.
7. Wait for the transmit data register to be empty, write the communication data to USARTn_TDR.TDR, transfer the data to the transmit shift register, and start sending;

   (When the CTS function is valid, the data is transferred to the transmit shift register when the USARTn_CTS input is low, and the transmission starts)
8. If you need to send data continuously, repeat step 7;
9. Confirm that the send is complete by confirming the USARTn _ SR.TC bit. When sending data continuously and using the send interrupt, the last send data can be written through the TI interrupt, and USARTn_CR1.TXEIE is written to 0 and USARTn_CR1.TCIE is written to 1. After the last frame of data is sent, a send completion interrupt is generated.

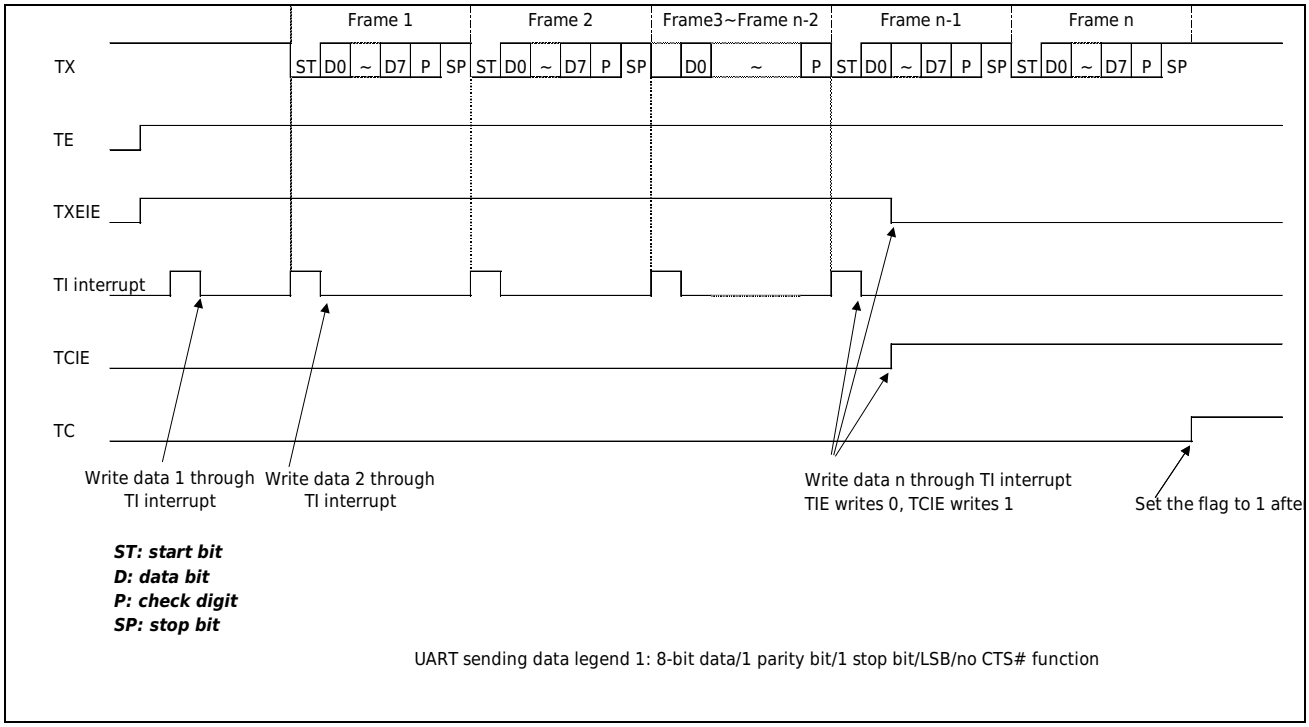


Figure 26-3 UART Send Data Legend 1

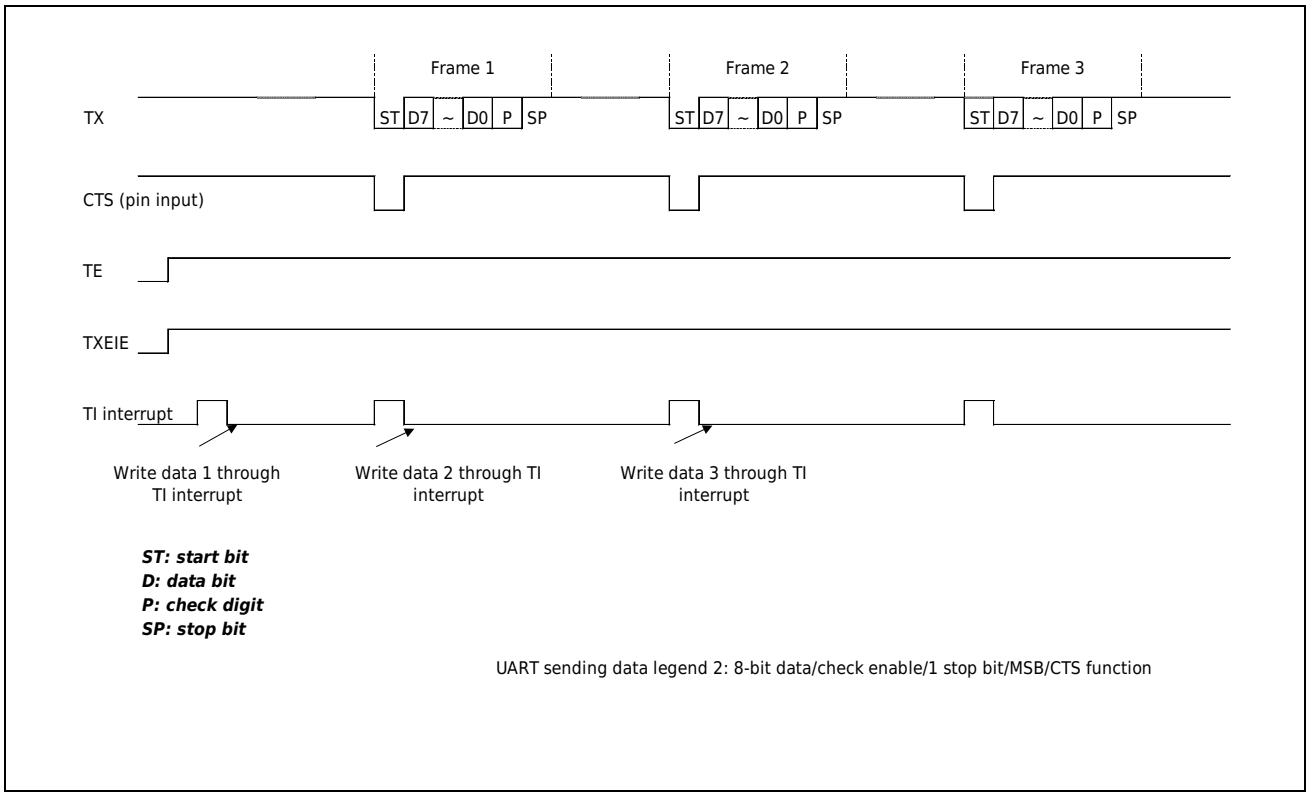


Figure 26-4 UART Send Data Legend 2

Transmitter Interrupt

The UART mode transmitter supports two types of interrupts, sending data register null interruptTI and sending complete interruptTCI.

TXEIE=1, TI interrupt occurs when the value of USARTn_TDR.TDR register is transferred to the transmit shift register.

TCIE=1, when the last bit of data is sent, if the USARTn_TDR.TDR register is not updated, a TCI interrupt occurs.

26.3.2.5 Receiver

Receivers can receive 8 or 9 bits of data depending on USARTn _ CR1.The set value of the 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. When a full frame of data is received, the data is transferred from the receive shift register to the receive data register USARTn_RDR.RDR.

The order of receiving data is: start bit -> data bit (MSB/LSB) -> check bit (yes or no) -> stop bit.

The receive data register USARTn_RDR. The RDR register and the internal receive shift register form a double buffer structure, which can continuously receive data.

A request can only read data once by receiving data register full interrupt or DMA read.

Start Bit Detection

Start bit detection can select low level mode or falling edge mode, depending on USARTn _ CR1.SBS bit, USARTn _ CR1.Low level detection at SBS = 0, USARTn _ CR1.When SBS = 1, it was the detection of falling edge.

Sampling and Reception Tolerance

After detecting the start condition (low level or falling edge), USART synchronizes the received data based on the internal basic clock to start the data reception.

Data sampling in the center of the data, USARTn _ CR1.OVER8 = 0 is sampled at the 8th internal base clock and USARTn _ MR. OVER8 = 1 is sampled at the 4th internal base clock.

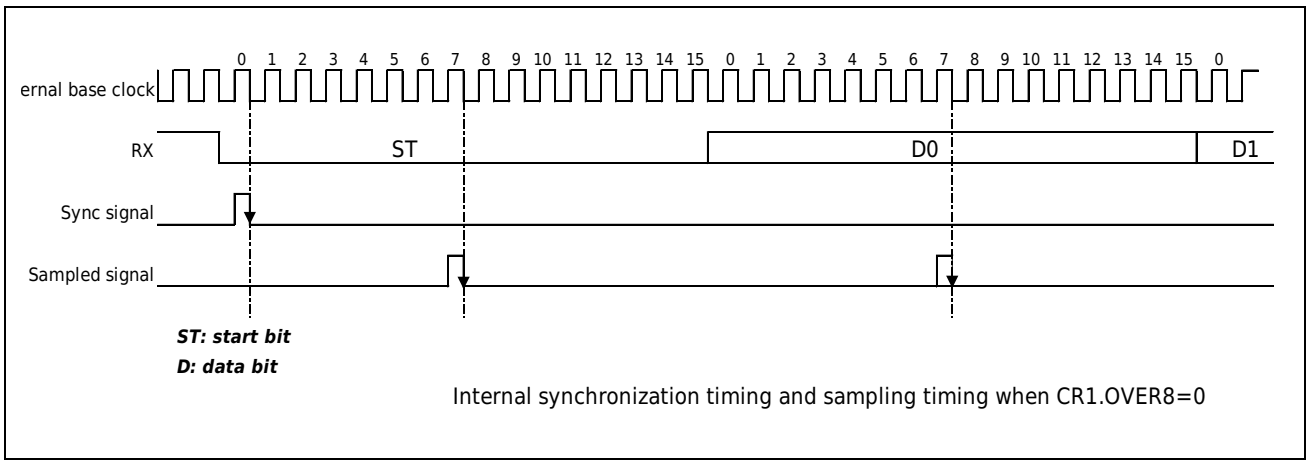


Figure 26-5 UART Internal Synchronization And Sampling Timing

The UART asynchronous receiver works only if the total clock system deviation is less than the tolerance of the UART receiver. Factors affecting the total deviation include:

- Deviation due to transmitter error (which also includes the local oscillator deviation of the transmitter)
- Error caused by quantization of baud rate of receiver
- Receiver local oscillator deviation
- Deviation caused by transmission lines

The maximum deviation allowed by the UART asynchronous receiver for correct reception of data depends on the following options:

- Data length FL (Frame Length). FL is determined by the 8 or 9 data bits defined by the M bit in the USART_CR1 register and the parity enable bit defined by the PCE bit
- 8x or 16x oversampling defined by the OVER8 bit in the USART_CR1 register
- Whether to use the fractional baud rate defined by the FBME bit in the USART_CR1 register

Table 26-2 Tolerance of the UART Receiver when DIV_Fraction is 0

FL	OVER8 bit	OVER8 bit=1
10	4.375%	3.75%
11	3.97%	3.41%
12	3.646%	3.125%

Table 26-3 Tolerance of UART Receiver when DIV_Fraction is not 0

FL	OVER8 bit	OVER8 bit=1
10	3.88%	3%
11	3.53%	2.73%
12	3.23%	2.5%

In special cases, when the RX pin receives a continuous high level for the data length FL time, the data specified in Table 26-2 and Table 26-3 may be slightly different.

Receiving Data Setting Procedure

1. Set the USARTn_CR1 register to the reset value;
2. Set the pins required for UART;
3. Select the clock source through USARTn_CR2.CLKC[1:0] bits;
4. Set USARTn_CR1, USARTn_CR2, USARTn_CR3 registers;
5. Set USARTn_PR to select the pre-scaling value, and USARTn_BRR register to set the communication baud rate (no need to set when the clock source is an external clock source);
6. Enable the receiver (USARTn_CR1. RE=1), if you need to use receive interrupt, set USARTn_CR1. RIE=1

7. When the start bit is detected, the receiver receives the received shift register and checks the check bit and stop bit
   - When an error is checked, the received data is transferred to the USARTn _ DR.RDRregister and the USARTn _ SR.PE flag is set.
   - When the stop bit is not high, a frame error occurs and the received data is sent to the USARTn _ DR.RDRregister and the USARTn _ SR.FE flag is set.
   - When an overflow error occurs, the data is lost and set to the USARTn _ SR.ORE flag
   - When no error occurs, the received data is transferred to the USARTn_RDR.RDR register and the USARTn_SR.RXNE flag is set. After reading the received data, repeat step 7 to continuously receive data.

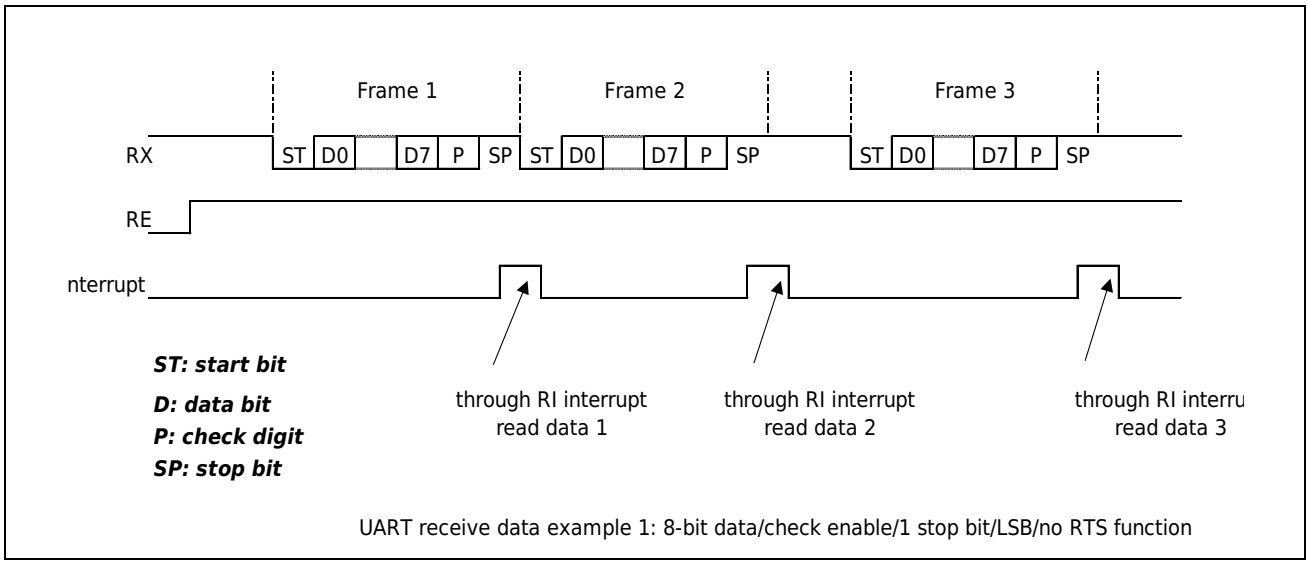


Figure 26-6 UART Received Data Legend 1

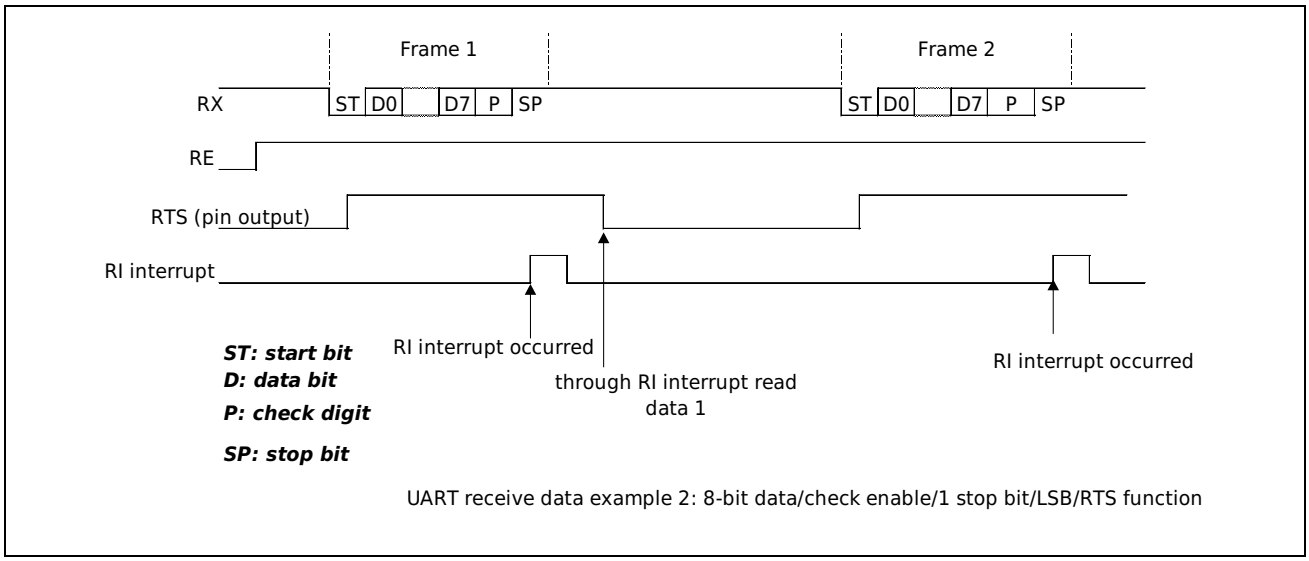


Figure 26-7 UART Received Data Legend 2

Error Handling

There are three types of receive errors when receiving data, namely overflow error (USARTn_SR.ORE), check error (USARTn_SR.PE) and frame error (USARTn_SR.FE). Any reception error can no longer be received. You can restart data reception by clearing all error flags by writing the corresponding zeroing register.

The condition for an overflow error is that a new frame of data is received without reading the USARTn_RDR.RDR register value, so the previous frame of data should be read before the last bit of the current frame is received.

Parity errors occur on condition that parity errors occur.

Frame errors occur on the condition that the stop bit is low and only the first stop bit is checked in the case of two stop bits.

Data loss received when an overflow error occurs, RIinterrupt does not occur.

When a check error occurs, the received data is transferred to USARTn_RDR.RDR, and the RI interrupt does not occur.

When a frame error occurs, the received data is transferred to USARTn_RDR.RDR, and the RI interrupt does not occur.

Receiver Interrupt

The UART mode receiver supports two interrupts, the receive data register full interrupt RI and the receive error interrupt EI.

USARTn _ CR.RIE = 1, no receiving error occurred, and RIinterrupt occurred when data was transferred from the receiving shift register to the receiving data register.

USARTn_CR.RIE=1, an EI interrupt occurs when an overflow error, a check error, or a frame error occurs during reception.

26.3.2.6 UART Receives TIMEOUT Function

The TIMEOUT counter starts when the stop bit of the UART receive data is detected. When the next frame of receive data is not detected after the set TIMEOUT time (the setting unit is the number of receive bits), TIMEOUT occurs. If CR1.RE=1 at this time, the TIMEOUT status bit USARTn_SR.RTOF is set. If USARTn_CR1.RE=0 at this time, the TIMEOUT status bit USARTn_SR.RTOF is set after waiting for USARTn_CR1.RE=1.

The TIMEOUT counter uses the counter of the Timer0 module, and the specific correspondence is as follows:

USART1: Timer0_1 A channel

USART2: Timer0_1 B channel

USART3: Timer0_2 A channel

USART4: Timer0_2 B channel

TIMEOUT Function Timer0 Comparison Counter Value Setting

Timer0 is a 16-bit counter, and the counting clock can be divided by up to 1024. The calculation formula for TMR0_CMPA<B>R value setting is as follows:

CMPA<B>R: TMR0_CMPA<B>R register value. Please round up the calculated result.

α: Delay caused by Timer0 asynchronous counting synchronization circuit

When the counting clock is not divided, α = 7

When the counting clock is divided by 2, α = 5

When the counting clock is divided by 4, 8, or 16, α = 3

Count clock divided by 32 or more, α = 2

RTB: Receive Timeout Bits, minimum value =

There is a certain error between the actual time when TIMEOUT occurs and the RTB value, the error is ≤ .

CKDIRA<B>: TMR0.BCONR.CKDIVA<B> bit register value

TIMEOUT function setting procedure

1. Set the USARTn_CR1 register to the reset value;
2. Set the pins required for UART;
3. Select the clock source through USARTn_CR2.CLKC[1:0] bits (if the internal clock source is selected, set CR2.CLKC[1:0]=0b01);
4. Set USARTn_CR1, USARTn_CR2, USARTn_CR3 registers;
5. Set USARTn_PR to select the pre-scaling value, and USARTn_BRR register to set the communication baud rate (no need to set when the clock source is an external clock source);
6. USARTn_CR1.RTOE=1, if you need to use interrupts, set USARTn_CR1.RTOIE=1;
7. Set TMR0_BCONR.CSTA<B>=0;
8. Set TMR0_CNTA<B>R to 0, set TMR0_CMPA<B>R register and TMR0_BCONR.CKDIVA<B> register to determine TIMEOUT time;
9. Set TMR0_BCONR.HCLEA<B>=1, TMR0_BCONR.HSTAA<B>=1, TMR0_BCONR.ASYNCLKA<B>=1, TMR0_BCONR.SYNCLKA<B>=0, TMR0_BCONR.SYNSA<B>=1;
10. Enable the receiver (USARTn_CR1. RE=1), if you need to use receive interrupt, set USARTn_CR1. RIE=1
11. After TIMEOUT is detected, follow the steps below to set and shut down the TMR0 timer and

clear the USARTn_SR.RTOF status bit.

Set TMR0_BCONR.SYNCLKA<B>=1 and TMR0_BCONR.SYNSA<B>=0 at the same time, and then set TMR0_BCONR.CSTA<B>=0, TMR0_BCONR.SYNCLKA<B>=0 and TMR0_BCONR.SYNSA<B>=1 at the same time;

Clear the USARTn_SR.RTOF status bit by writing USARTn_CR1.CRTOF.

26.3.2.7 RX Line Stop Mode Wake-up Enable

When the UART communication is idle, the system can enter the Stop Mode to save current consumption. Without changing the UART PORT setting, UART unit 1 can wake up the system from the Stop Mode through the RX line. Specific steps are as follows:

1. When UART communication is idle, set USART_1_WUPI interrupt vector and INTC_WKEN.RXWKEN bit to enable the UART receive signal line wake-up Stop Mode function;
2. System Enter Stop Mode.
3. When the system detects the falling edge of the RX line, it returns from the Stop Mode and disables this function in the USART_1_WUPI interrupt handler.

It should be noted that when the communication party needs to wake up the system, it needs to send a frame of wake-up data (recommended to be 0x00), this data will not be received by UART and the relevant flag will not be set . And the communication party needs to pass the time required for the system Stop Mode to wake up before transmitting the UART communication data.

The UART RX line wake-up function can filter the noise on the RX line. For details, please refer to the USART_NFC register.

26.3.2.8 UART Half-Duplex Communication Mode

UART mode supports one-wire half-duplex mode, UART mode by setting USARTn _ CR3.HDSEL = 1 Enables one-wire half-duplex mode.

Single-wire half-duplex mode:

- TX and RX lines are connected from inside and no RX pins are used.
- When there is no data transmission and the transmission function disable, the TX pin is in the released state. Therefore, when single-line half-duplex mode is used, floating input without data transmission is avoided by pulling up the TX line.
- When data is sent and the sending function is enabled, the TX pin outputs a high level.

In addition, half-duplex mode is similar to normal UART mode communication. Note that the sending process is not blocked by hardware, as long as the data is in USARTn _ CR1.When TE = 1, it is written and sent. Therefore, conflicts on the line must be managed by software.

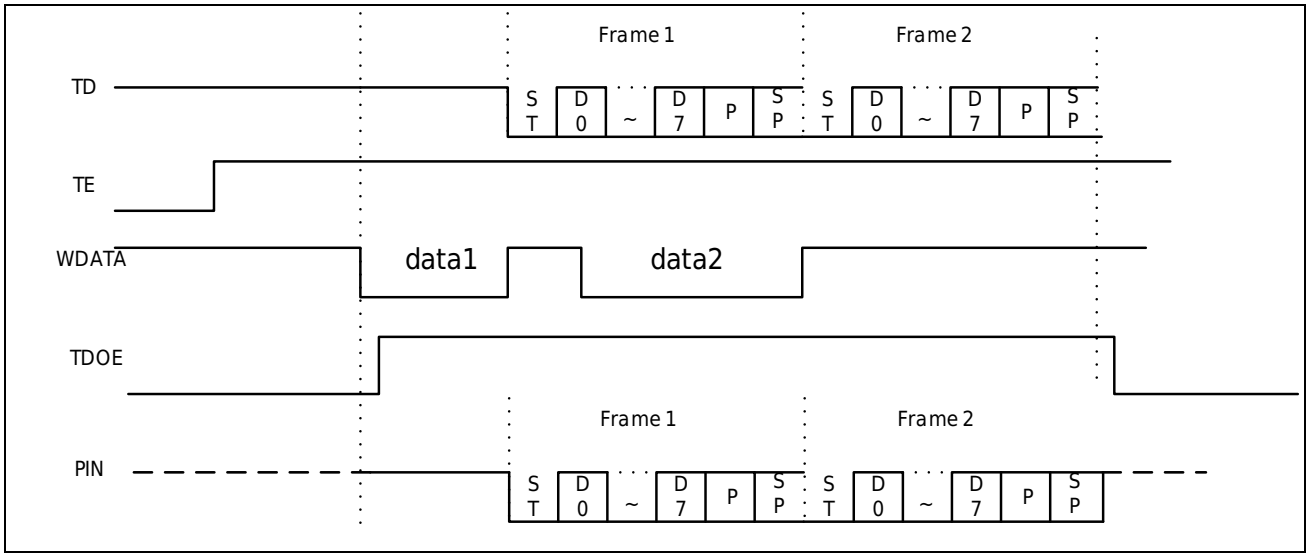


Figure 26-8 Half-duplex Working Sequence

26.3.2.9 UART Driver Enable Function

UART mode supports driver enable mode, and shares the output signal with RTS. You need to configure USART_CR3.DEM to high first, and then configure USART_CR3.RTSE to high.

The driver enable function is turned on by setting bit DEM in the USART_CR3 control register, allowing the user to activate external transceiver control through the DE (Driver Enable) signal. The assertion time programmed using the DEA[3:0] bit field in the USART_CR3 control register is the time between the activation of the DE signal and the start of the START bit.

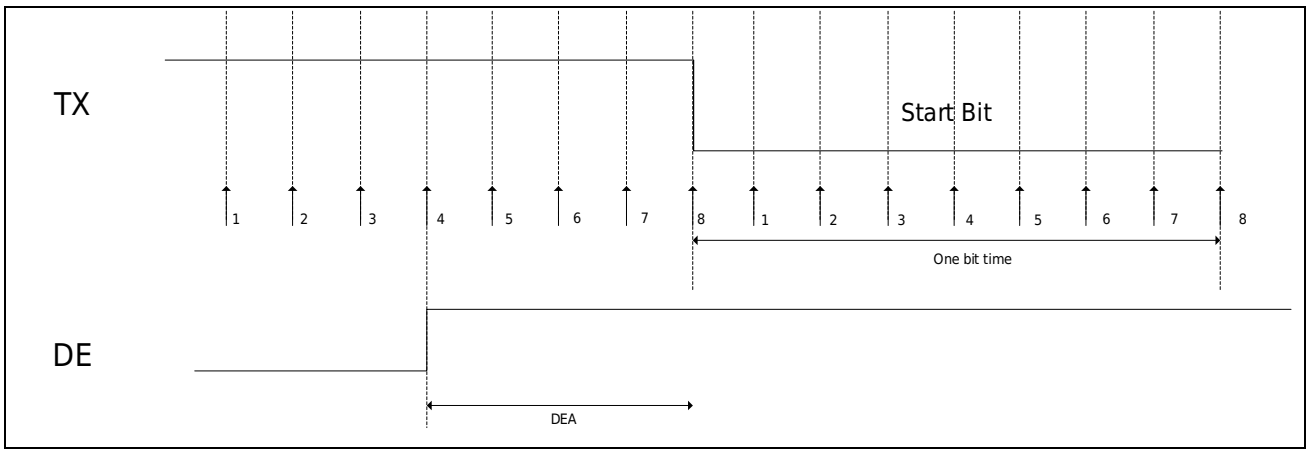


Figure 26-9 over8=1 DE assertion

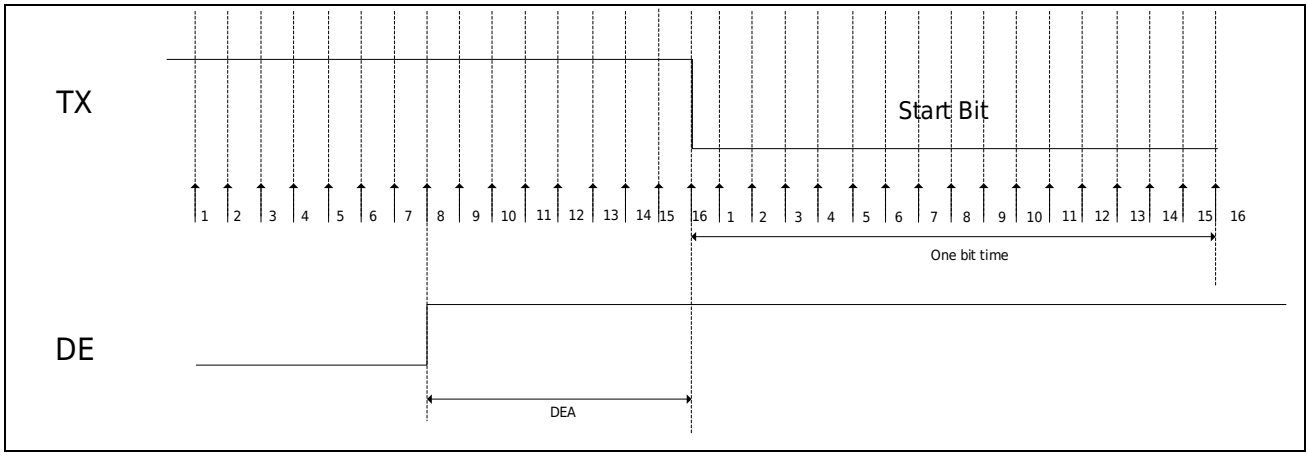


Figure 26-10 over8=0 DE assertion

The de-assertion time programmed using the DED[3:0] bit field in the USART_CR3 control register is the time between the end of the last stop bit and the de-assertion of the DE signal. The polarity of the DE signal can be configured using the DEP bit in the USART_CR3 control register.

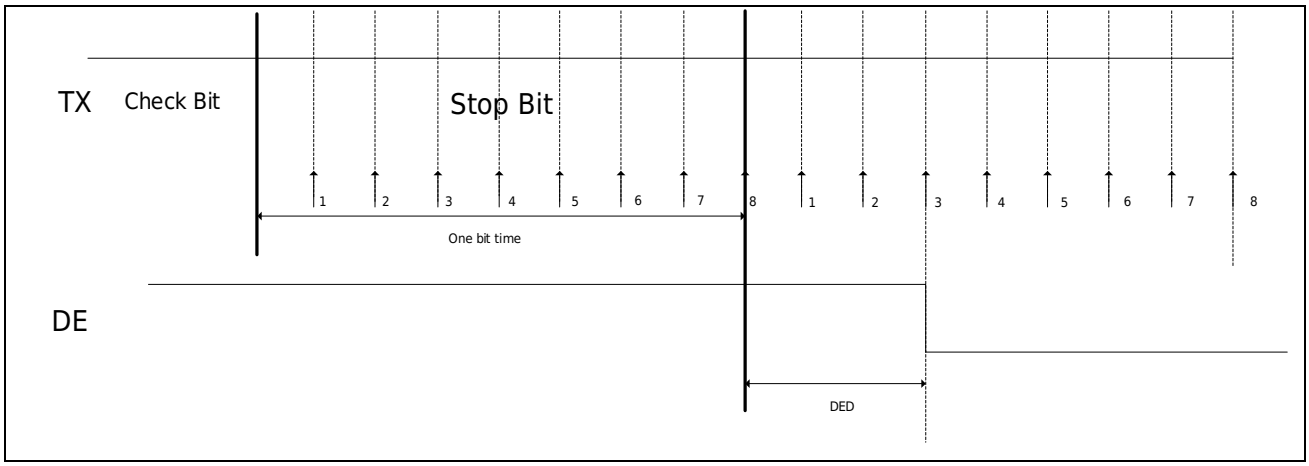


Figure 26-11 over8=1 DE De-asserts

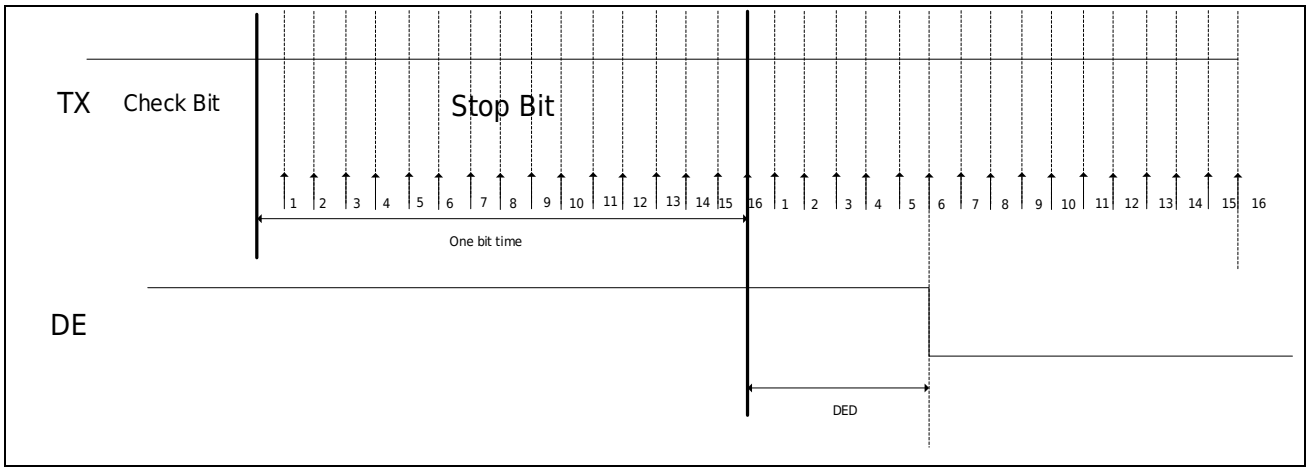


Figure 26-12 over8=0 DE de-asserts

26.3.2.10 UART interrupt and Events

Table 26-4 UART Interrupt/Event Table

Functional name	Enable bit (interrupt only)	Flag bit	Can I be the source of an event
Reception error interrupt	RIE	ORE，FE，PE	Yes
Received data register full interrupt	RIE	RXNE	Yes
Send data register empty interrupt	TXEIE	TXE	Yes
Send complete interrupt	TCIE	TC	Yes
Send complete interrupt	TENDIE	TEND	Yes
TIMEOUT interrupt	RTOIE	RTOF	Yes
RX Stop Mode wake-up interrupt	INT_WUPEN.RXWUEN	-	No

26.3.3 Multiprocessor Communication

26.3.3.1 Feature Brief

Multi-processor communication mode refers to a communication mode in which multiple processors share communication lines. The processor is divided into transmitting and receiving stations, and each receiving station has its own ID. There are two types of transmitting data: Receiving station ID and communication data. By adding MPB bits to the data format to distinguish whether the receiving station ID or communication data is currently being sent. When the MPB bit is 0, the current frame is the communication data, and when the MPB bit is 1, the current frame is the ID of the receiving station. All receiving stations can receive the ID sent by the transmitting station and compare it with their ID. If it is consistent, the receiving data will be received, and if it is inconsistent, the receiving station will enter silent mode (neither receiving data nor setting the relevant flag) until the ID is received again.

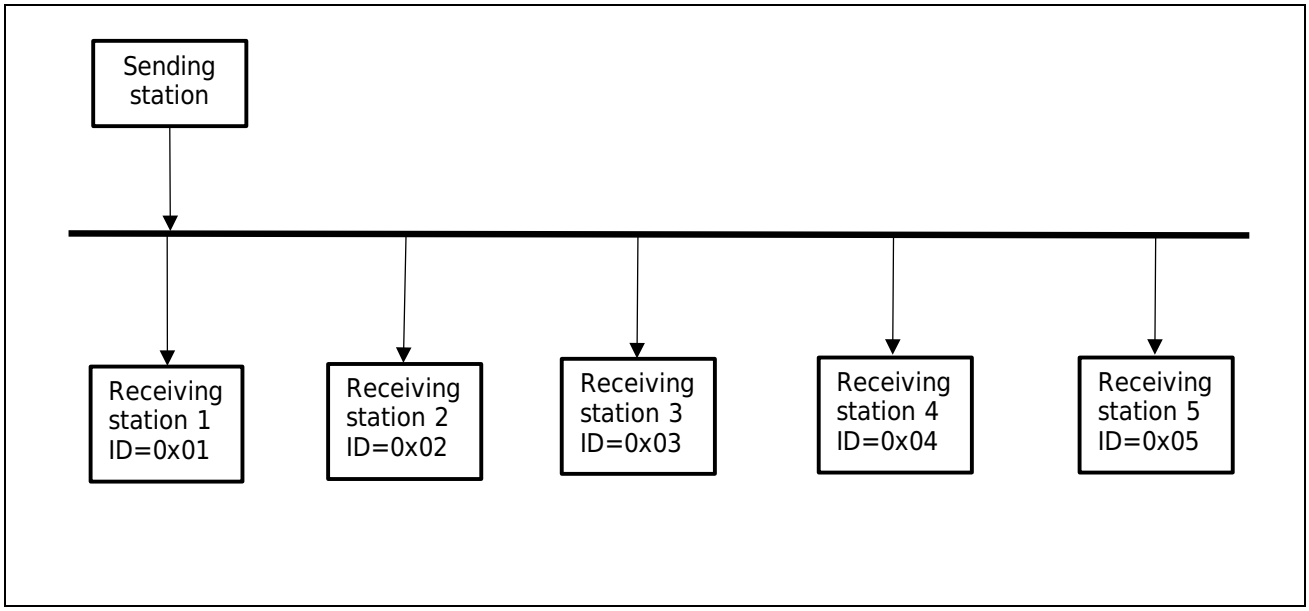
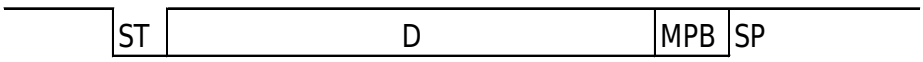


Figure 26-13 Multiprocessor Communication Diagrams

26.3.3.2 Data Format



*ST: Start bit (1 bit **low level**)*
D: Data bits (can be configured to 8 or 9 bits)
*MPB: **multiprocessor** bit (1: ID, 0: data)*
*SP: stop bit (1~2 **high level** can be configured)*

Figure 26-14 Multiprocessor Mode Data Format

26.3.3.3 Action Description

In the multi-processor mode, the check bit function is invalid, and the multi-processor bit function is added. The other functions such as clock and interrupt are the same as UART mode.

Transmitting Station Action

1. Set the USARTn_CR1 register to the reset value;
2. Set the pins to be used;
3. Select the clock source through USARTn_CR2.CLKC[1:0] bits;
4. Set USARTn_CR1, USARTn_CR2, USARTn_CR3 registers;
5. Set USARTn_PR to select the pre-scaling value, and USARTn_BRR register to set the communication baud rate (no need to set when the clock source is an external clock source);
6. Enable the sendter (USARTn_CR1.TE=1). If you need to use the send data register null interrupt, set USARTn_CR1.TXEIE=1.
7. Wait for the transmit data register to be empty, set USARTn_TDR.MPID bit to 1 (send ID), write ID value to USARTn_TDR, and send ID;

   (When the CTS function is valid, the data is transferred to the transmit shift register when the USARTn_CTS input is low, and the transmission starts)
8. Set USARTn_TDR.MPID bit to 0 (send data), write data to USARTn_TDR, and send data;

   (When the CTS function is valid, the data is transferred to the transmit shift register when the USARTn_CTS input is low, and the transmission starts)
9. If you need to send data continuously, repeat step 8. If you need to change the ID before sending data, repeat 7 and 8.
10. Confirm that the send is complete by confirming the USARTn _ SR.TC bit. To continuously send data and use the sending interrupt, the last sent data can be written via TIinterrupt and USARTn

_ CR1.TXEIE write 0, USARTn _ CR1.TCIE writes 1. After the last frame of data is sent, the transmission is interruptd.

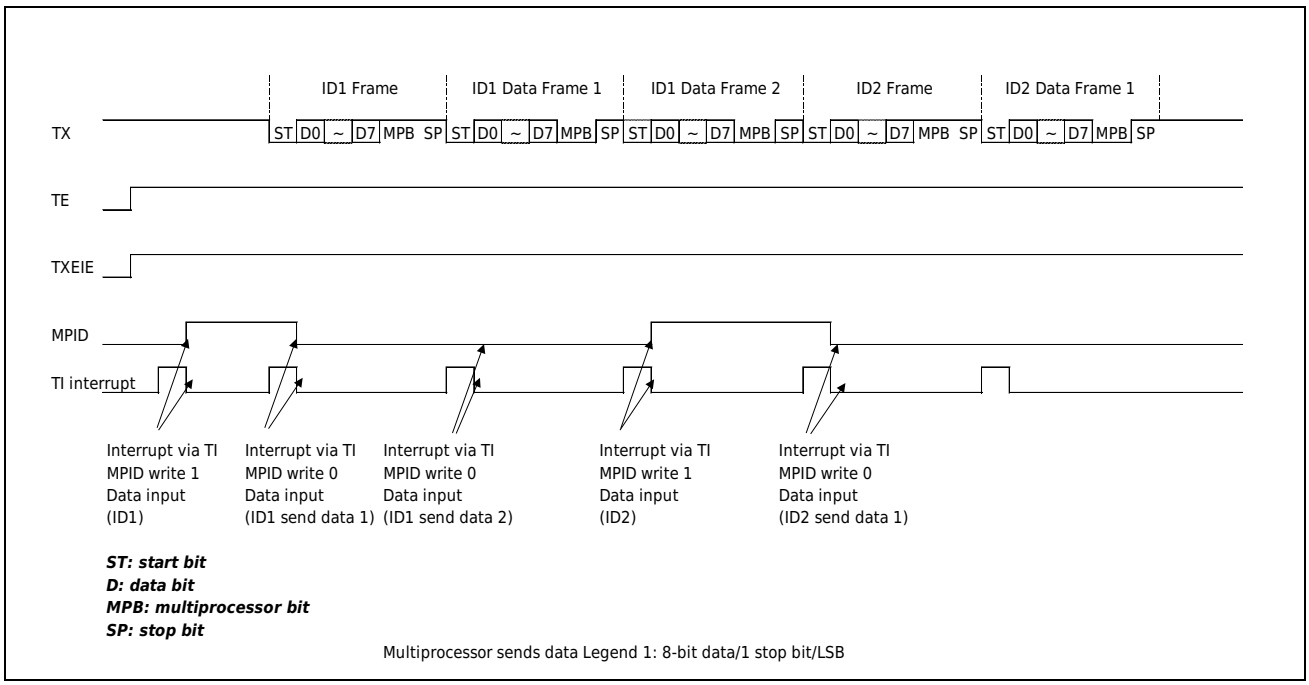


Figure 26-15 Diagram of Multi-Processor Mode Send Data

Receiving Station Action

In multiprocessor mode, the receiving station must ensure that each ID data is received and compared with its own ID. If it is consistent, the receiving station receives the data. If it is inconsistent, the receiving station enters silent mode (no data is received, and no related flag is set) until the next ID data is received. This function is implemented by the USARTn_CR1.SLME bit.

USARTn_CR1. Receive data normally at SLME = 0.

When USARTn_CR1. SLME = 1, unless the data with the MPB bit as 1 (ID) is received, no data is received, no RI interrupt occurs, and the error flags FE and ORE are not set. When receiving data with the MPB bit of 1 (ID), the USARTn_CR1.SLME bit is automatically cleared, and the data is received normally and an interrupt occurs.

Action steps:

1. Set the USARTn_CR1 register to the reset value;
2. Set the pins to be used;
3. Select the clock source through USARTn_CR1.CLKC[1:0] bits;
4. Set USARTn_CR1, USARTn_CR2, USARTn_CR3 registers;
5. Set USARTn_PR to select the pre-scaling value, and USARTn_BRR register to set the communication baud rate (no need to set when the clock source is an external clock source);

6. USARTn_CR1.RE=1, USARTn_CR1.SLME=1 (waiting for receiving ID). If using receive interrupt, set USARTn_CR1.RIE=1;

7. When the start bit is detected, the receiver receives the data into the receive shift register and checks the USARTn_SR.MPB bit;

8. If USARTn_SR.MPB=1, USARTn_CR1.SLME bit is automatically cleared, data is received normally, and the software compares the received ID with its own ID.

1) If the ID is consistent, the data is normally received, interruptd, and the error is detected, the same as the data received by UART

2) If the ID does not match, the software will USARTn _ CR1 again. SLME bit write 1, repeat 8

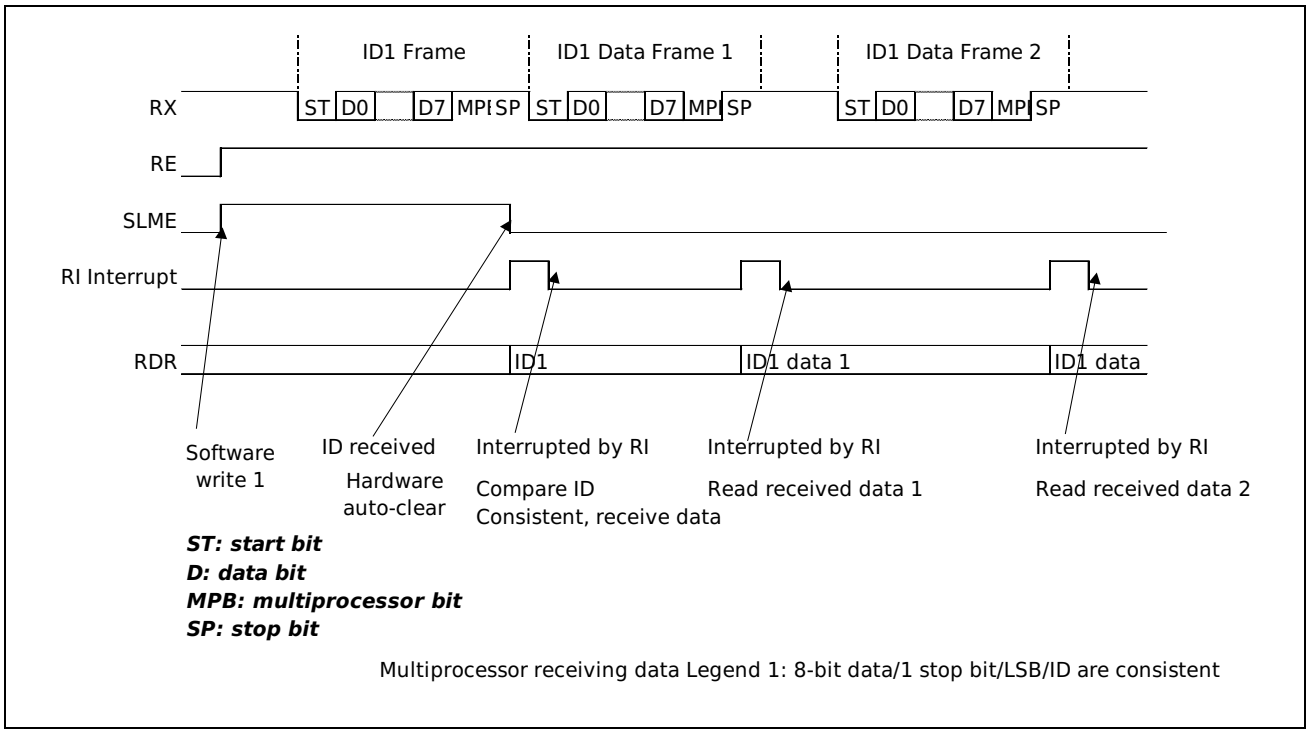


Figure 26-16 Diagram of Multiprocessor Mode Receive Data 1

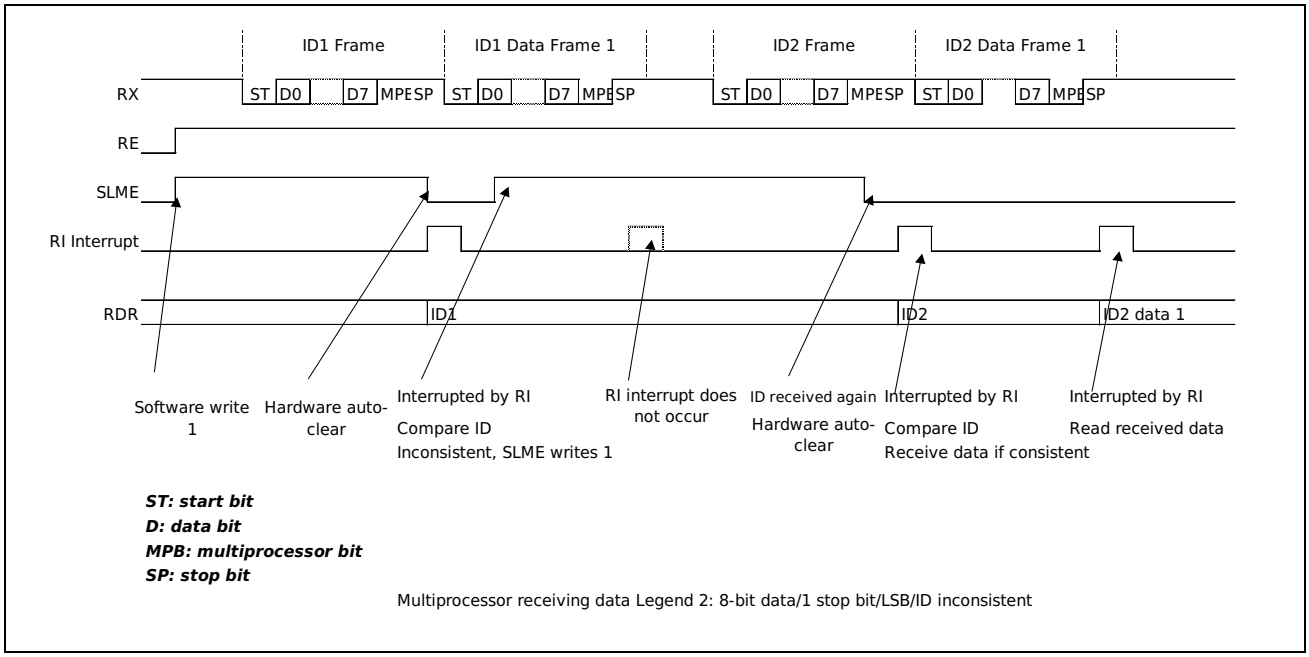


Figure 26-17 Diagram of Multiprocessor Mode Receive Data 2

26.3.3.4 Interrupt and Events

The interrupt mode is the same as the UART mode except for no check error.

Table 26-5 Multi-Processor Mode Interrupt/Event

Functional name	Enable bit (interrupt only)	Mark	Can I be the source of an event
Reception error interrupt	RIE	ORE, FE	Yes
Received data register full interrupt	RIE	RXNE	Yes
Send data register empty interrupt	TXEIE	TXE	Yes
Send complete interrupt	TCIE	TC	Yes
Send complete interrupt	TENDIE	TEND	No

26.3.4 UART-LIN

26.3.4.1 Feature Brief

LIN is short for Local Interconnect Network, which is a low-speed (1~20Kbps) serial communication protocol to reduce the cost of automotive network.

26.3.4.2 LIN Data Format

A frame of LIN data is composed of start bit + 8 bit data + 1 stop bit, which is used to send and receive data in LSB mode. The data behavior on the LIN bus is shown below.

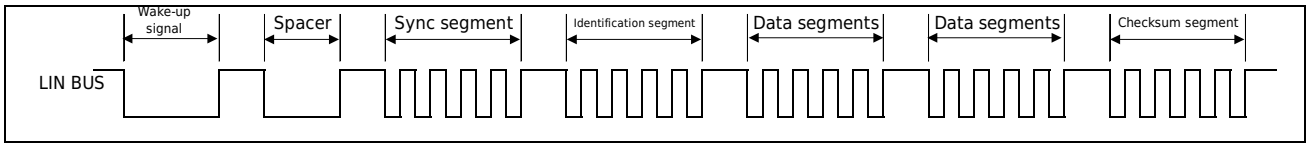


Figure 26-18 LIN Bus Data Behavior

26.3.4.3 LIN Sending

Sending of Wake Signal Frames

The signal frame is sent by sending data 0x80.

Interval Sending

The synchronization interval segment indicates the beginning of a frame of data. The hardware of this product supports the sending of LIN interval segments.

The width of the low level of the LIN interval is set by the USARTn_CR2.SBK register.

When USARTn_CR2.SBKM=0, write 1 to USARTn_CR2.SBK and the hardware will automatically send out the interval segment.

When USARTn_CR2.SBKM=1, after USARTn_CR2.SBK is written 1, write data 0x00 to the send data register USARTn_TDR.TDR to start sending the interval segment.

After the interval segment is sent, USARTn_CR2.SBK is automatically cleared.

Synchronization Segment Sending

Sending Synchronous Segment Data by Sending Data 0x55.

Bus Error Detection

When USARTn_CR2.BEE=1, the hardware detects that the data on the bus is inconsistent with the sent data, and sets the USARTn_SR.BE flag. When USARTn_CR2.BEIE=1, a corresponding interrupt is generated.

26.3.4.4 LIN Receiving

Detection of Wake Signal Frames

When USARTn_CR2.WKUPE=1, the hardware automatically detects the wake-up signal. When it detects that the low-level width of the RX line is greater than or equal to 2.5 bits of data width (130μs at 19.2Kbps), the USARTn_SR.WKUP flag is set. When USARTn_CR2.WKUPIE=1, a corresponding interrupt is generated.

The USARTn_CR2.WKUPE register is set to 1 only when the system needs to wait for the wake-up signal, and needs to be set to 0 in other cases.

Detection of Synchronous Interval Segments

The hardware of this product supports synchronous interval detection. When it is detected that the low-level width of the communication line is greater than or equal to the value set by the USARTn_CR2.LBDL register and the break delimiter is detected, the USARTn_SR.LBD flag is set. When USARTn_CR2.LBDIE=1, a corresponding interrupt is generated.

It should be noted that when the synchronization interval is detected, the USARTn_SR.FE flag will be set. When clearing the USARTn_SR.LBD flag, it is necessary to clear the USARTn_SR.FE flag and read the USARTn_RDR.RDR register (the read value is 0x00).

Detection of Synchronous Segments and Baud Rate Measurement

When the interval segment is detected, the hardware automatically measures the frequency of the synchronization segment. The clock of the measurement counter is set by the USARTn_PR.LBMPSC bit, and the value of the counter is saved in the USARTn_LBMC register. The baud rate of the master node can be obtained by dividing the selected counter clock frequency by the USARTn_LBMC register value.

It should be noted that the reading of the USARTn_LBMC register needs to be done after the synchronization segment is received.

Receiving Data Setting Procedure

1. Set the USARTn_CR1 register to the reset value;
2. Set the pins required for UART;
3. Select the clock source through USARTn_CR2.CLKC[1:0] bits;
4. Set USARTn_CR1 (RE=1, RIE=0), USARTn_CR2, USARTn_CR3 registers;

   (Steps 5-6 below are for the detection of wake-up signal frames, skipped if not necessary)
5. Set USARTn_CR2.WKUPE=1, wait for the wake-up signal;
6. Detect the wake-up signal, USARTn_CR2.WKUPE=0, wait for the synchronization segment interval;

   (Step 7 below is for synchronization segment interval field detection, skip if not necessary)
7. After detecting the synchronization interval, clear the USARTn_SR.PE/FE/LBD flag and confirm that the received data is 0x00;

   (Steps 8 to 9 below are for synchronization segment frequency measurement and communication baud rate calculation, skip if not necessary)
8. If you need to use the receive interrupt, set USARTn_CR1.RIE=1;
9. After the synchronization interval field is detected and the synchronization segment reception is completed, the USARTn_LBMC register is read to calculate the baud rate and confirm that the received data is 0x55;

   (Steps 10-12 below are Receive Identification Segments, Data and Checksum, the process is the same as the UART Receive Data process)
10. Receiving identification segment;

11. Receiving data;

12. Receiving checksum.

26.3.4.5 LIN interrupt and Events

Table 26-6 LIN interrupt/Event Table

Functional name	Enable bit (interrupt only)	Flag bit	Can I be the source of an event
error interrupt	RIE	ORE，FE	Yes
	BEIE	BE	Yes
Wake-up signal/break detection interrupt	WKUPIE	WKUP	Yes
	LBD	LDBIE	Yes
Received data register full interrupt	RIE	RXNE	Yes
Send data register empty interrupt	TXEIE	TXE	Yes
Send complete interrupt	TCIE	TC	Yes
Send complete interrupt	TENDIE	TEND	Yes

26.3.5 Smart Card

26.3.5.1 Connection Diagram

Support the smart card communication protocol specified by ISO/IEC 7816-3. underflow - Smart card Pattern mode with connected legend.

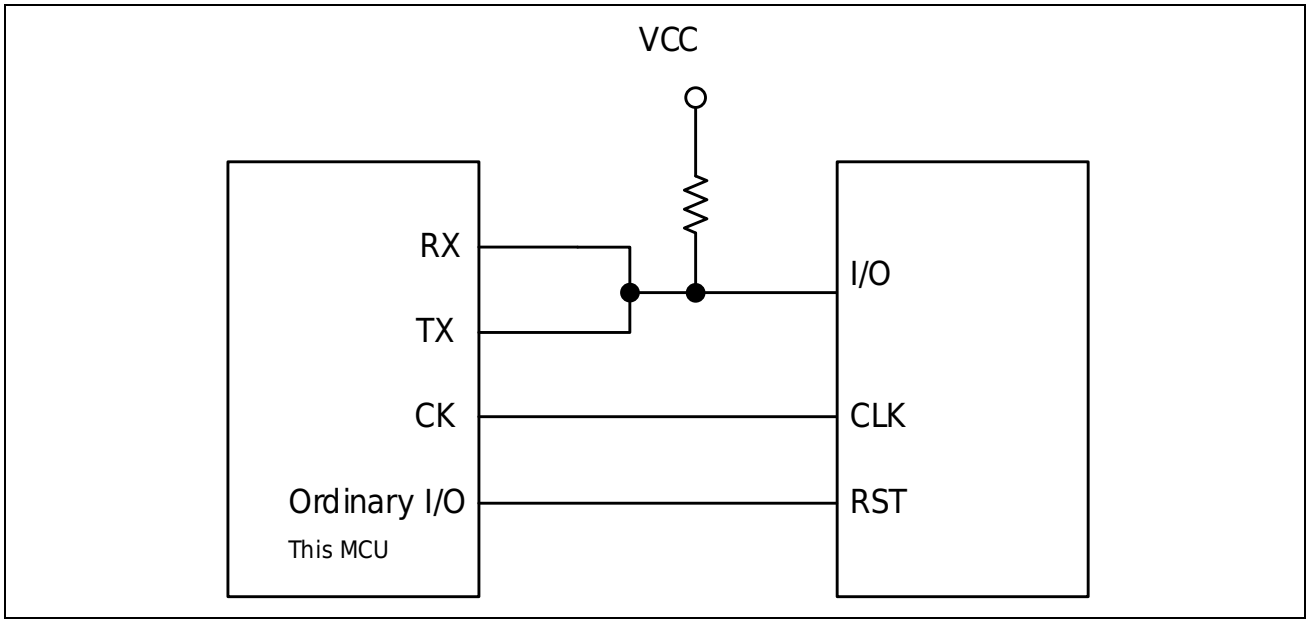


Figure 26-19 Smart card Pattern connected legend

26.3.5.2 Clock

Internal timer

Only the clock generated by the internal baud rate generator can be used as the clock source in smart card mode.

The basic clock number for one-bit data transmission is the setting value of USARTn_CR3.BCN[2:0].

The clock output of smart card mode is controlled by setting register USARTn_CR2.CLKC[1:0].

Highest baud rate

For internal timer, the Baud Rate formula generated by the internal Baud Rate Generator is:

B: Baud rate unit: Mbps
C: When USARTn_PR.ULBREN is 0, the clock set by USARTn_PR.PSC[1:0] bits (PCLK1, PCLK1/4, PCLK1/16, PCLK1/64) Unit: MHz
When USARTn_PR.ULBREN is 1, the clock set by USARTn_PR.PSC[1:0] bits (PCLK/128, PCLK/256, PCLK/512, PCLK/1024) is in MHz.
DIV _ Integer: USARTn _ BRR.DIV _ Integer setting
BCN: USARTn_CR3.BCN register setting value
When C is PCLK1, DIV_Integer=0, and BCN=0, the baud rate is the maximum baud rate PCLK1/64 (Mbps).

Sampling and reception tolerance

After detecting the fall of RX, USART will synchronize the clock of the received data based on the internal basic clock to start data reception. Received data will be sampled at the data center.

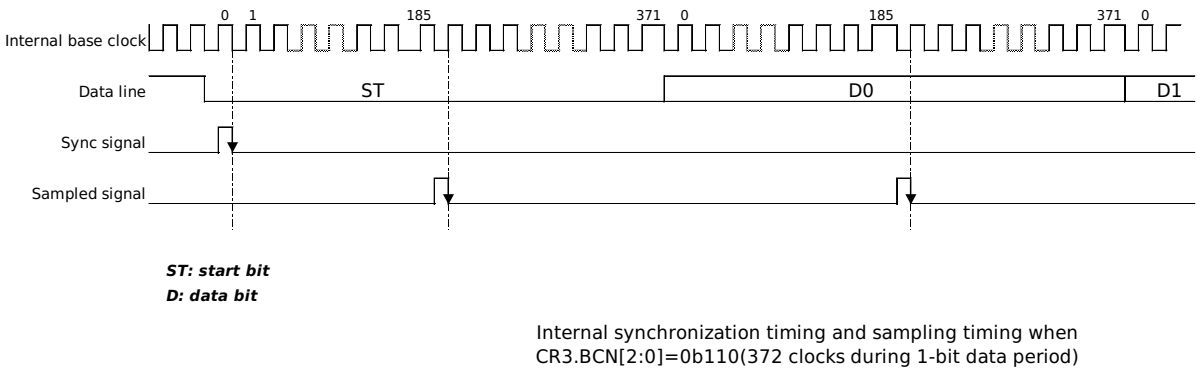


Figure 26-20 Smart Card Pattern Synchronization and Sampling Timing Figure

The following formula is used to calculate the receiving tolerance:

RM：*reception tolerance*
BCN: The number of clocks required for one bit data transmission (USARTn.CR3.BCN[2:0] setting value)
CFD: Clock frequency deviation

26.3.5.3 Data Format

In smart card mode, a frame of data consists of start bit, data bit and check bit.

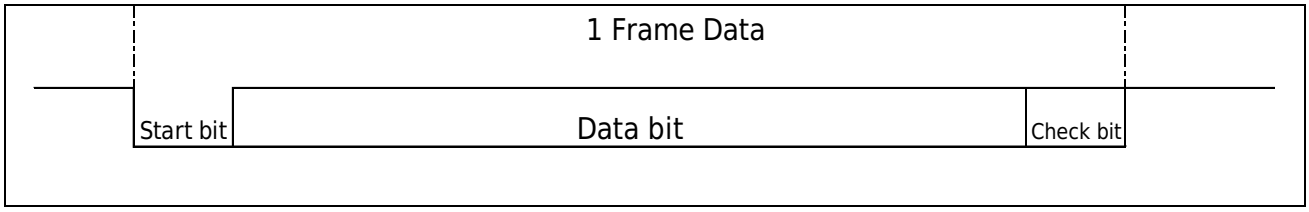


Figure 26-21 Smart Card ModeData Format

Start Bit

The start bit is fixed with a low level.

Data Bit

The data bits are fixed to 8-bit data.

Check Bit

The parity bit needs to be configured as 1-bit even parity.

26.3.5.4 Initial Setting Procedure of Smart Card

1. Set the USARTn_CR1 register to the reset value;
2. Set the pins to be used;
3. The status register confirms that the USARTn_SR register is set to the reset value
4. Set USARTn_CR1, USARTn_CR2, USARTn_CR3 registers;
5. Set USARTn_PR to select the pre-frequency division value and USARTn_BRR register to set the communication baud rate
6. USARTn_CR2.CLKC[1:0] bits set clock control
7. USARTn_CR1 register (TE, RE, RIE, TXEIE bit) setting, except for self-test, TE and RE should not be set to 1 at the same time

   When switching from sending mode to receiving mode, or switching from receiving mode to sending mode, you need to reset the above steps 1 to 7.

26.3.5.5 Smart Card Pattern Action Description

In smart card mode, the flag bit of TI interrupt (send data empty interrupt) is USARTn_SR.TC bit. TI interrupt is generated when USARTn_SR.TC=1 and USARTn_CR1.TXEIE=1.

Functional Overview

When sending data, there is a protection time of more than 2 etu (Elementary Time Unit) between two frames of data (from the end of the parity bit to the start bit of the next frame).

When sending data, if an error signal sent by the receiver is detected, the data will be resent automatically after 2 etu.

When a verification error occurs in the received data, a low level of 1etu is sent, which is an error signal. The timing of sending the error signal is 10.5etu after the start of reception.

Send Instructions

After a frame of data is sent, if an error signal sent by the receiver is detected, USARTn_SR.FE is set to 1 (if USARTn_CR.RIE=1, an error interrupt occurs), the USARTn_SR.TC flag is not set to 1 , and the data is automatically resent. The USARTn_SR.FE bit must be cleared before receiving the next frame parity bit.

After a frame of data is sent without error, the USARTn_SR.TC flag is set, and a TI interrupt occurs when USARTn_CR1.TXEIE=1. Write data again, then the data can be sent continuously.

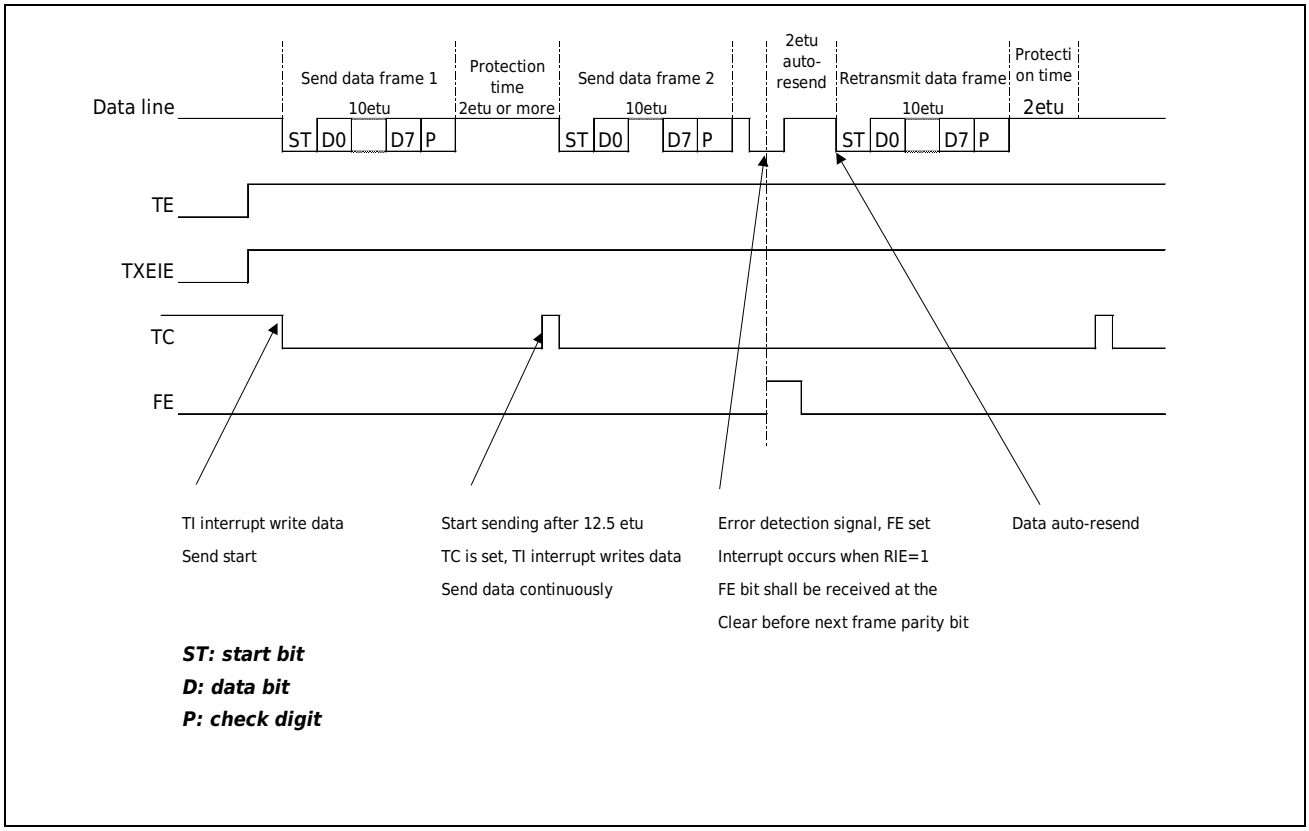


Figure 26-22 Smart Card Pattern Sending Data Legend

Receiving Instructions

1) When receiving data, if a parity error is detected, USARTn_SR.PE will be set, and when the interrupt is enabled, a EI interrupt will occur. DR.RDRregister and the USARTn _ SR.PE bit must be cleared before receiving the next frame parity bit.

2) When a verification error occurs, a low level of 1etu will be sent, that is, an error signal, requiring the sender to resend the data.

3) To receive data normally, you can read the received data through RI interrupt and receive continuously.

4) When receiving data, an overflow error is detected.

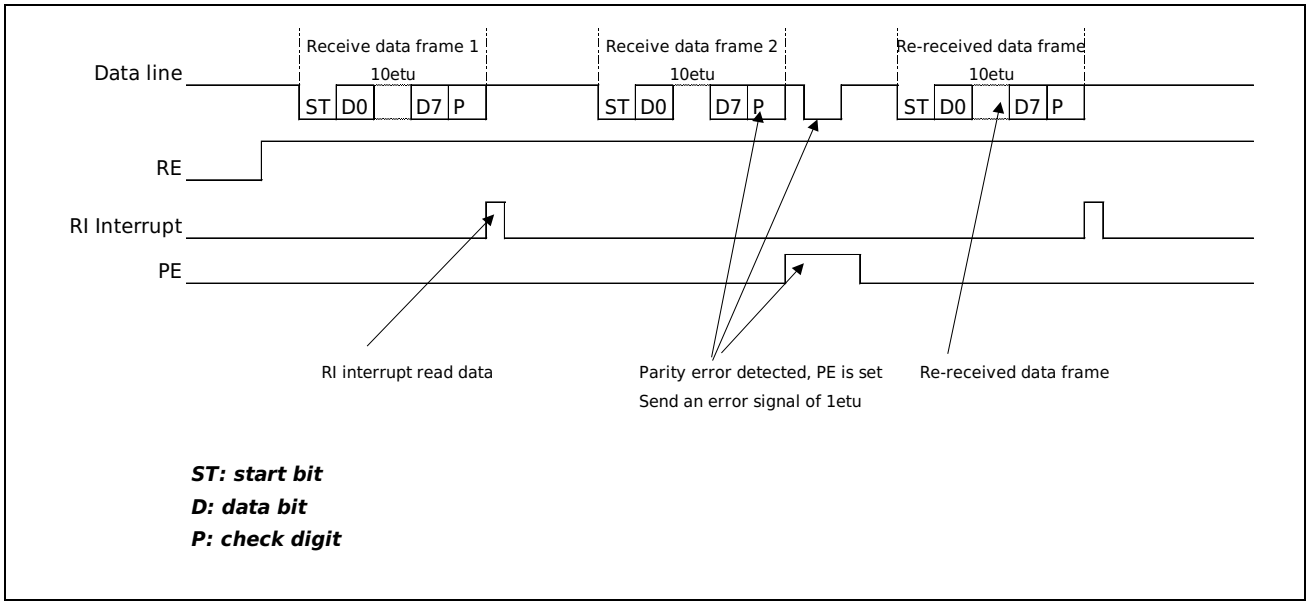


Figure 26-23 Smart Card Pattern Receive Data Legend

26.3.5.6 Interrupt and Events

Table 26-7 Smart Card Mode Interrupt/Event

Functional name	Enable bit (interrupt only)	Flag bit	Can I be the source of an event
error interrupt	RIE	ORE，PE，FE	Yes
Received data full Interrupt	RIE	RXNE	Yes
Send data null Interrupt	TXEIE	TC	Yes

26.3.6 Clock Synchronization Mode

26.3.6.1 Clock

The clock synchronization mode can select the clock generated by the internal baud rate generator (internal timer) or the clock input from the USARTn _ CK pin (external timer) as the timer for communication.

Internal timer

The synchronization clock is output from the USARTn _ CK pin, and a frame of data outputs eight clock pulses. When neither the data nor the data is sent, the clock output is fixed at a high level.

External timer

The external timer is the input clock from the USARTn _ CK pin as the communication clock.

Highest baud rate

For internal timer, the Baud Rate formula generated by the internal Baud Rate Generator is:

B: Baud rate unit: Mbps

C: When USARTn_PR.ULBREN is 0, the clock set by USARTn_PR.PSC[1:0] bits (PCLK1, PCLK1/4, PCLK1/16, PCLK1/64) Unit: MHz

When USARTn_PR.ULBREN is 1, the clock set by USARTn_PR.PSC[1:0] bits (PCLK1/128, PCLK1/256, PCLK1/512, PCLK1/1024) Unit: MHz

DIV _ Integer: USARTn _ BRR.DIV _ Integer setting

When the internal clock source is used, when C is PCLK1 and DIV_Integer=1, the maximum baud rate is PCLK1/8 (Mbps). Note that the DIV _ Integer disable is set to 0 in synchronous mode.

When the clock source is external, the maximum frequency of the external input clock is required to be PCLK1/6, so the maximum baud rate is PCLK1/6 (Mbps).

It should be noted that in addition to the calculation method based on PCLK1 described above, the maximum communication baud rate in synchronous mode also needs to refer to the maximum communication baud rate specified in the Section **Electrical Features of Datasheet.**

26.3.6.2 Data Format

One frame of data in clock synchronization mode is composed of eight bits, and eight synchronization clock pulses are needed to send and receive one frame of data. Data is sent at the falling edge of the synchronization clock when data is sent and sampled at the rising edge of the synchronization clock when data is received.

The synchronization clock is fixed at a high level when there is no data transfer. After the last bit is sent, the communication line keeps the last bit value.

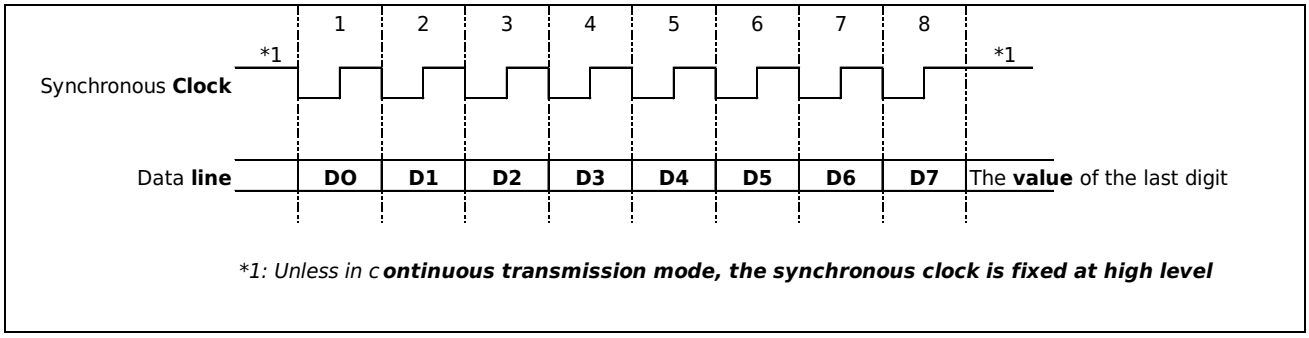


Figure 26-24 Clock Synchronization Mode Data Format

26.3.6.3 Modem Operation

Modem operations include CTS and RTS functions. When USARTn_CR3.RTSE=1, the RTS function is valid, and when USARTn_CR3.CTSE=1, the CTS function is valid.

CTS Function

The CTS function controls the sending of data through the input of the USARTn_CTS pin. The data can be sent only when the USARTn_CTS pin enters the low level. The data being sent is not affected if the USARTn_CTSpin enters the high level.

RTS Function

The RTS feature is a low level output from the USARTn_RTS pin to request the other party to send data.

TheUSARTn_RTS pin output low level needs to meet all of the following conditions:

- Receive enable (USARTn_CR1.RE=1) and not receiving data
- There is no unread data in the USARTn_RDR.RDR register (when USARTn_CR1.RE=1)
- USARTn_TDR.TDR update end (when USARTn_CR1.TE=1)
- No reception error

If all the above conditions are not met simultaneously,USARTn_RTS outputs a high level.

26.3.6.4 Transmitter

Transmitter enable bit (USARTn _ CR1.TE) When set to 1, the data in the transmit shift register is serially output in the USARTn _ TX pin and the corresponding clock pulse is output in the USARTn _ CK pin.

The transmit data register USARTn_TDR.TDR and the internal transmit shift register form a double buffer structure, which can send data continuously.

To ensure the correctness of sending data, a request can only be written once by sending data register empty interrupt or DMA.

Sending Data Setting Procedure

1. Set USARTn_CR1, USARTn_SR1 registers to reset values;
2. Set the pins to be used;
3. Select the clock source through USARTn_CR2.CLKC[1:0] bits;
4. Set USARTn_CR1, USARTn_CR2, USARTn_CR3 registers;
5. Set USARTn_PR to select the pre-scaling value, and USARTn_BRR register to set the communication baud rate (no need to set when the clock source is an external clock source);
6. Enable the sendter (USARTn_CR1.TE=1). If you need to use the send data register null interrupt, set USARTn_CR1.TXEIE=1.
7. Wait for the transmit data register to be empty, write the communication data to USARTn_TDR.TDR, transfer the data to the transmit shift register, and start sending;

   (When the CTS function is valid, the data is transferred to the transmit shift register when the USARTn_CTS input is low, and the transmission starts)

8. If you need to send data continuously, repeat step 7;

9. Confirm that the send is complete by confirming the USARTn _ SR.TC bit. When sending data continuously and using the send interrupt, the last send data can be written through the TI interrupt, and USARTn_CR1.TXEIE is written to 0 and USARTn_CR1.TCIE is written to 1. After the last data is sent, a send completion interrupt is generated.

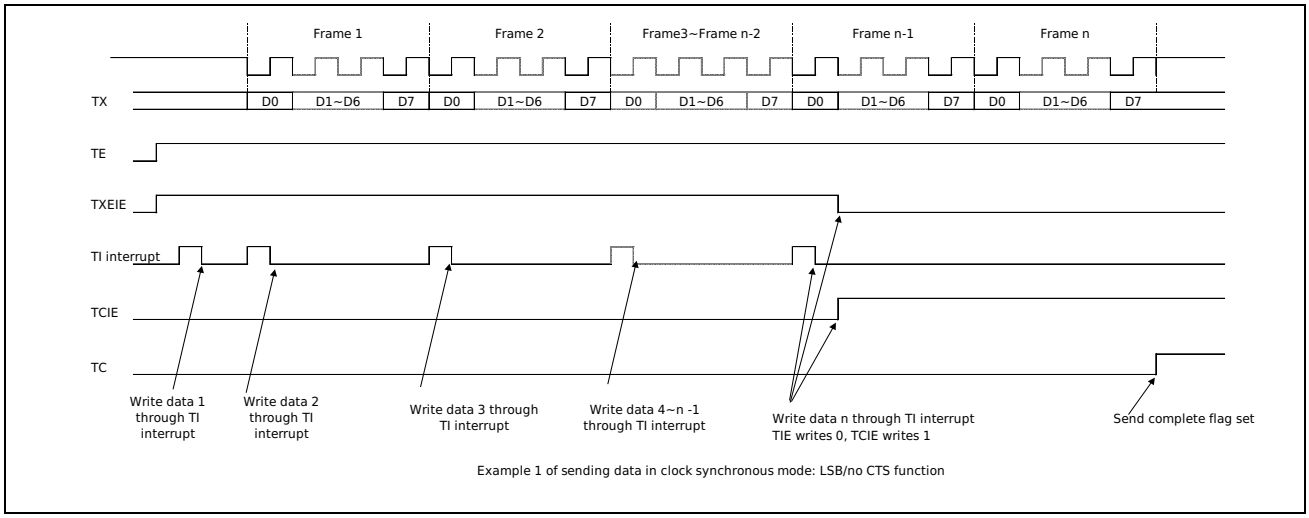


Figure 26-25 Diagram of Clock Synchronization Mode Sending Data 1

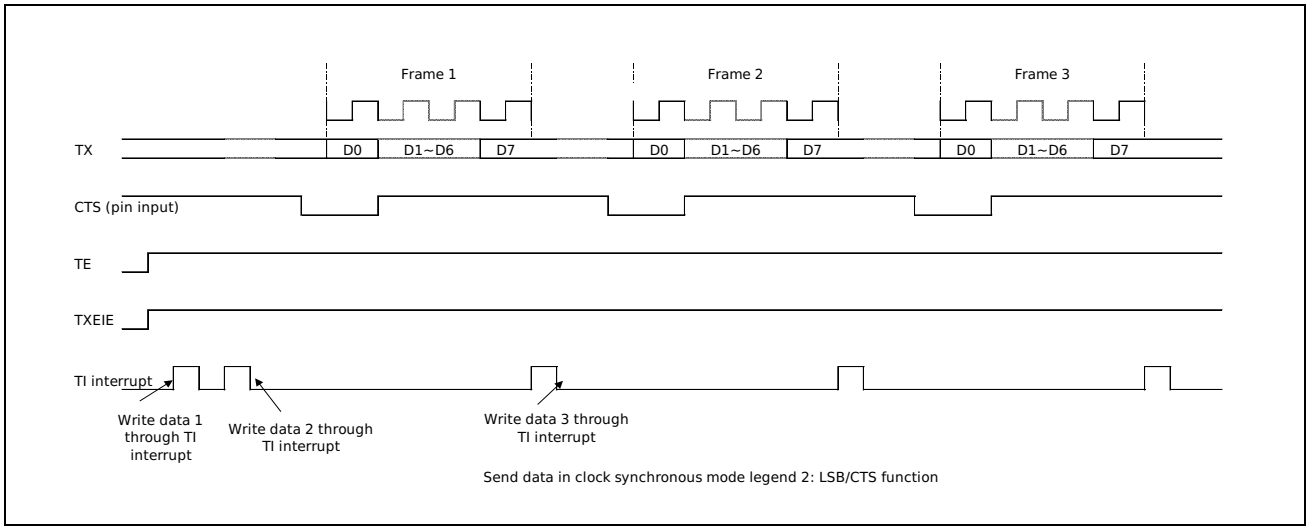


Figure 26-26 Clock Synchronization Mode Sending Data Legend 2

Transmitter Interrupt

The Clock Synchronization Mode Transmitter supports two types of interrupts, sending data register null interruptTI and sending complete interruptTCI.

TXEIE=1, TI interrupt occurs when the value of USARTn_TDR.TDR register is transferred to the transmit shift register.

TCIE=1, when the last bit of data is sent, if the USARTn_TDR.TDR register is not updated, a TCI interrupt occurs.

26.3.6.5 Receiver

Receiving Data Setting Procedure

1. Set USARTn _ CR1, USARTn _ SR register to reset
2. Set the pins to be used;
3. Select the clock source through USARTn_CR2.CLKC[1:0] bits;
4. Set USARTn_CR1, USARTn_CR2, USARTn_CR3 registers;
5. Set USARTn_PR to select the pre-scaling value, and USARTn_BRR register to set the communication baud rate (no need to set when the clock source is an external clock source);
6. Enable the receiver (USARTn_CR1. RE=1), if you need to use receive interrupt, set USARTn_CR1. RIE=1

   (When using the RTS function, USARTn_RTS outputs low level after RE=1)
7. Synchronize with the input synchronization clock or internally generated synchronization clock to start receiving data, receiving data to the receive shift register.

1) When an overflow error occurs, the data is lost and set to the USARTn _ SR.ORE flag
2) 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 of data is received, and then step 7 is repeated to achieve continuous Receive data function

   (When using the RTS function, USARTn_RTS outputs a low level after data is read)

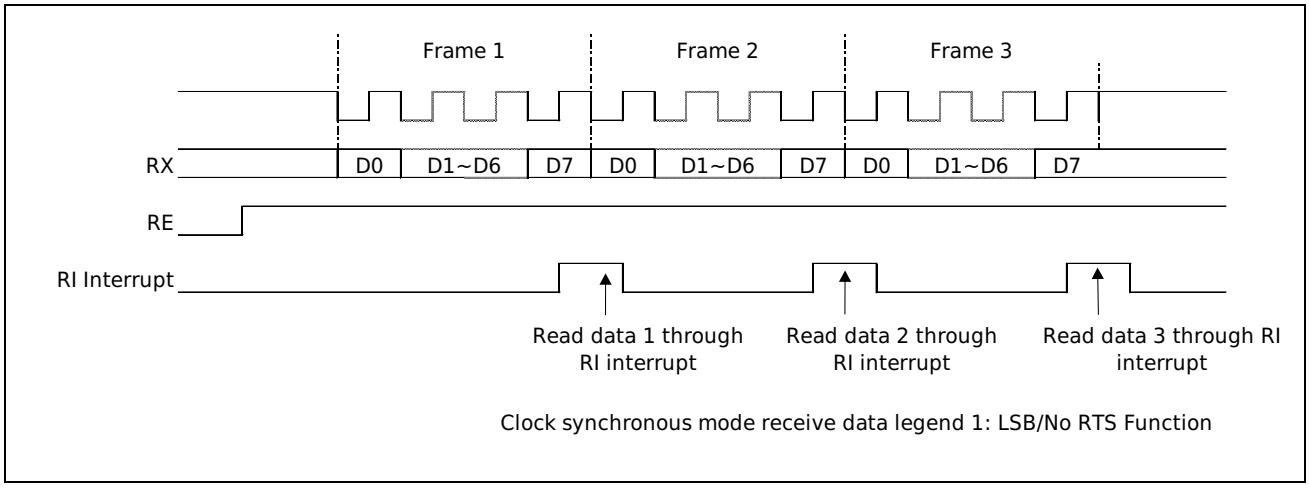


Figure 26-27 Diagram of Clock Synchronization Mode Receive Data 1

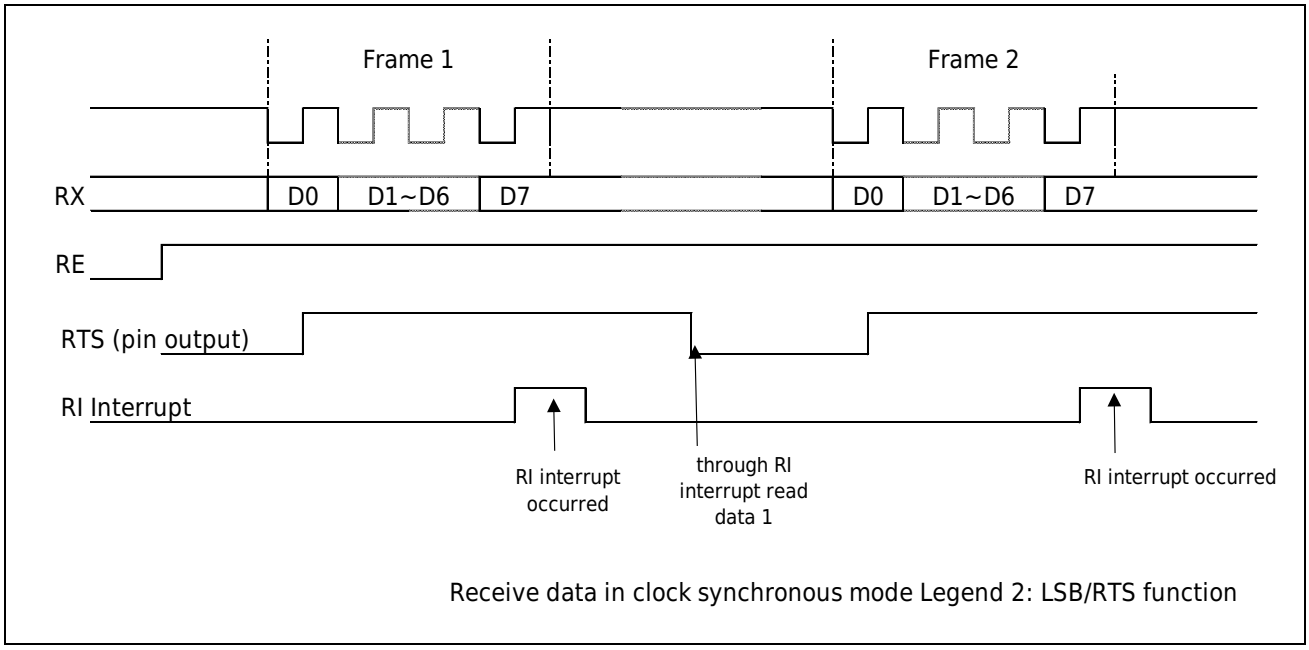


Figure 26-28 Diagram of Clock Synchronization Mode Receive Data 2

Error Handling

An overflow error (USARTn _ SR.ORE) is received while receiving data in clock synchronization mode. Receiving errors can no longer be received and sent. You can restart the data transfer by clearing the error flag by writing the corresponding zeroing register.

The overflow error occurs when the USARTn_RDR.RDR value is not read and new data is received, so the data received in the previous frame should be read before the last bit of the current frame is received. Data loss received when an overflow error occurs, RIinterrupt does not occur.

Receiver Interrupt

The clock synchronization mode receiver supports two types of interrupts, receive data register full interruptRI and receive error interrupt EI.

RIE = 1, RIinterrupt occurs when data is transferred from the receiving shift register to the receiving data register.

RIE = 1, EI interrupt occurs when an error (overflow error) occurs in the received data.

26.3.6.6 Send and Receive Data at the Same Time

The USART clock synchronization mode supports full-duplex operation while sending and receiving data. A command is needed to write RE, TE, RIE and TXEIE into one while transmitting and receiving data. The other setting flow is the same as that of transmitter and receiver.

26.3.6.7 Clock Synchronization Mode Interrupt and Events

Table 26-8 Clock Synchronization Mode Interrupt/Event Table

Interrupt name	Enable bit (interrupt only)	Flag bit	Can be used as an event source
Reception error interrupt	RIE	ORE	Yes
Received data register full interrupt	RIE	RXNE	Yes
Send data register empty interrupt	TXEIE	TXE	Yes
Send complete interrupt	TCIE	TC	Yes
Send complete interrupt	TENDIE	TEND	Yes

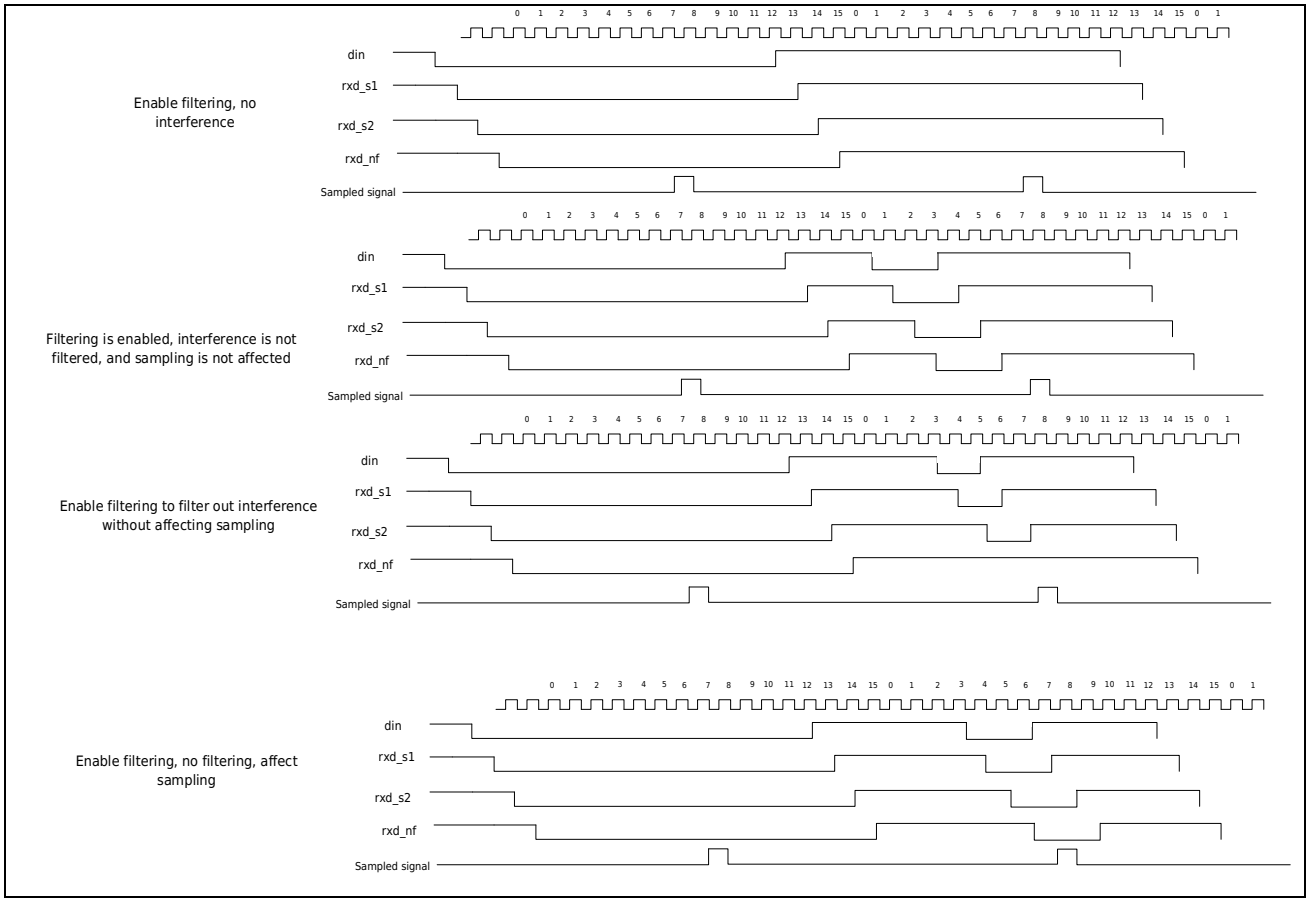
26.3.7 Digital Filtering Function

USARTn_CR1.When NFE = 1, the built-in digital filter function is effective. The digital filter is only effective in UART mode and removes noise from the received data line RX.

The built - in digital filter can filter 3/16 (USARTn _ CR1.OVER8 = 0 or 3/8 width (USARTn _ CR1.OVER8 = 1) noise.

If the clock of the digital filter stops before it starts, the digital filter continues to work from the time the clock stops.

USARTn _ CR.TE = 0 and USARTn _ CR.RE = 0 reset the Flip-Flop state inside the digital filter to 1.



Turning on the NF function actually delays the input rd signal twice to obtain three signals. Through these two delays, the input signal and the two delayed signals must be consistent before they can be flipped inside the module.

26.3.8 Interrupt

The following table gives the USART overall interrupt:

Table 26-9 USART Overall Interrupt List

Interrupt Name	Symbol	Description
Error interrupt	USART_n_EI n=1~4	UART/Multiprocessor/Smart Card/Clock Synchronous Mode Receive Error Interrupt LIN Error Interrupt
Received data register full interrupt	USART_n_RI n=1~4	UART/Multiprocessor/LIN/Smart Card/Clock Synchronous Mode Receive Data Register Full Interrupt LIN Error Interrupt
Send data register empty interrupt	USART_n_TI n=1~4	UART/Multiprocessor/LIN/Smart Card/Clock Synchronous Mode send Data Register null Interrupt
Send complete interrupt	USART_n_TCI n=1~4	UART/Multiprocessor/LIN/Smart Card/Clock Synchronous Mode send Complete Interrupt (Writing send data to the send data register can clear the interrupt)
Send complete interrupt	USART_n_TENDI n=1~4	UART/Multiprocessor/LIN/Smart Card/Clock Synchronous Mode send Complete Interrupt (Write 1 to register CR1.CTEND bit or write send data to send data register to clear interrupt)
Wake-up signal/break detection interrupt	USART_n_BRKWKPI n=2~4	LIN wake-up signal/break detection interrupt
UART receive TIMEOUT interrupt	USART_n_RTO n=1~4	UART receive TIMEOUT interrupt
RX Stop Mode wake-up interrupt	USART_1_WUPI	USART_1 RX line wake-up Stop Mode interrupt

26.4Register Description

This chapter will describe the USART module control status and other related registers in detail. It should be noted that the function register bit is only valid if the channel is configured with the corresponding function, otherwise it defaults to 0.

Table 26-10 USART Base Address

Name	Base Address	Description
USART1	0x4001 CC00	USART1 base address
USART2	0x4001 D000	USART2 base address
USART3	0x4001 D400	USART3 base address
USART4	0x4001 D800	USART4 base address
USART1_NFC	0x4005 5400	USART1_NFC base address

Table 26-11 USART Register List

Register	Description	Offset Address	Bit Width	Reset Value
USART_SR	Status register	0x00	32	0x0000 00C0
USART_TDR	Send Data	0x04	16	0x01FF
USART_RDR	Receive Data Register	0x06	16	0x0000
USART_BRR	Baud rate register	0x08	32	0x0000 FFFF
USART_CR1	Control register 1	0x0C	32	0x8000 0000
USART_CR2	Control register 2	0x10	32	0x0000 0600
USART_CR3	Control register 3	0x14	32	0x0000 0000
USART_PR	Prescaler Register	0x18	32	0x0000 0000
USART_LBMC	LIN Baud Rate Measurement Count Register	0x1C	32	0x0000 0000

Table 26-12 USART1_NFC Register List

Register	Description	Offset Address	Bit Width	Reset Value
USART1_NFC	USART1 Filter Control Register	0x1C	32	0x0000 0000

26.4.1 Status Register (USART _ SR)

Reset value: 0x0000 00C0

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															MPB

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res				TENDF	LBD	WKUP	RTOF	TXE	TC	RXNE	BE	ORE	Res	FE	PE

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b17	Res	Reserved Bit	Read at “0”, write at “0”	RW
b16	MPB	Multiprocessor position	Multi-processor bit flag 0: Current received data is communication data 1: Current received data is ID **Note:** The MPB bit is only valid in multi-processor mode.	R
b15~b12	Res	Reserved Bit	Read at “0”, write at “0”	RW
b11	TENDF	Send completion mark	Send completion flag bit 0: Send data 1: Completion of data transmission UART mode, Clock synchronization mode TEND setting condition • When the last bit of a frame of data is sent, the value of the sent data register is not updated TEND zeroing condition • When TE = 1, write the sending data to the sending data register • Clear register CR1.CTEND bit write 1 Smart card pattern TEND setting condition • After a certain time has elapsed after the last 1 byte of data is sent, FE=0 and the value of the send data register is not updated. The specific timing of TEND setting is: 2.5 digits after the check digit is sent TEND zeroing condition • When TE = 1, write the sending data to the sending data register • Clear register CR1.CTEND bit write 1	R
b10	LBD	LIN interval detection flag	LIN interval detection flag (only valid for USART_2/USART_3/USART_4) 0: No interval detected 1: Interval segment detected LBD setting condition When CR2.WKUPE=0, after detecting that the low level width of the RX line is greater than or equal to the width set by the CR2.LBDL register, and the break delimiter is detected Clearing conditions Clear register CR1.CLBD bit write 1 **Note:** RE=0 does not reset the LBD bit	R
b9	WKUP	LIN Wakeup Signal Flag	LIN Wakeup signal detection flag (valid only for USART_2/USART_3/USART_4) 0: No wakeup signal detected 1: Wakeup signal detected WKUP setting condition When CR2.WKUPE=1, the low level width of the RX line is detected to be greater than or equal to 2.5 bits of data width (130uS at 19.2Kbps) Clearing conditions Clear register CR1.CWKUP bit write 1 **Note:** RE=0 does not reset the WKUP bit After detecting that the WKUP signal is 1, it indicates that the LIN node has exited from Sleep Mode. After setting register CR2.WKUPE=0, subsequent operations such as interval field detection are performed.	R
b8	RTOF	UART receive TIMEOUT Flag bit	UART receive TIMEOUT flag (all channels) 0: no UART receive TIMEOUT 1: Occurrence UART receive TIMEOUT RTOF setting condition • No new received data is detected after the set time has elapsed since the STOP bit of the last frame data was detected RTOF zeroing condition • Clear register CR1.CRTOF bit write **Note:** RTOF is set to 1 by hardware, and it is set to 1 only when CR1.RE=1 and	R

			CR1.RTOE=1. When CR1.RE=0, the TIMEOUT function is valid, but RTOF is not set to 1.	
b7	TXE	Send data register empty	Send data register null flag The TXE bit valid in UART/clock synchronization mode/LIN mode. 0: Data not transmitted to shift register, send data register is not empty 1: Data transfer to shift register, send data register empty **Note:** TXE bits are set by hardware 1 and clear 0, TXE is cleared by hardware when data is not transmitted to shift register, TXE is cleared by hardware when data is transmitted to shift register 1	R
b6	TC	Send completion mark	Send completion flag bit 0: Send data 1: Completion of data transmission UART mode, Clock synchronization mode TC setting condition • TE=0: Sending is prohibited • When the last bit of a frame of data is sent, the value of the sent data register is not updated TC zeroing condition • When TE = 1, write the sending data to the sending data register Smart card pattern TC setting condition • TE=0: Sending is prohibited • After a certain time has elapsed after the last 1 byte of data is sent, FE=0 and the value of the send data register is not updated. The specific timing of TC setting is: 2.5 digits after the check digit is sent TC zeroing condition • When TE = 1, write the sending data to the sending data register **Note:** When the TE bit changes from 0 to 1, TC remains at 1.	R
b5	RXNE	Received data register is not empty	Received data register is not null 0: No data received 1: Ready to read received data **Note:** The RXNE bit is set by hardware 1 and cleared by 0, the RXNE bit is set by hardware 1 when ready to read the received data, and the RXNE bit is cleared by hardware 0 after reading the received data.	R
b4	BE	LIN bus error detection flag	LIN bus error detection flag (only valid for USART_2/USART_3/USART_4) 0: No bus error 1: A bus error occurred BE setting condition • When sending data, the bus data and the sent data are detected to be inconsistent. The detection position is 13/16 of each bit (16-bit oversampling) BE zeroing condition • Clear register CR1.CBE bit write 1 **Note:** TE=0 does not reset the BE bit When BE=1, after the data being sent is completed, no new data will be sent. The software needs to clear the BE flag and restart the sending action.	R
b3	ORE	Reception overflow error	Reception overflow error flag bit 0: No Receive Overflow Error 1: Reception overflow error ORE setting condition • A new frame of data was received when the received data register was not read ORE zeroing condition • Clear register CR1.CORE bit write 1 **Note:** RE=0 does not reset the ORE bit Data received before ORE = 1 is maintained, and data received when ORE = 1 is lost ORE = 1 does not continue to receive data and cannot send data in clock synchronization mode	R
b2	Res	Reserved Bit	Read at “0”, write at “0”	RW
b1	FE	Received frame error	Received frame error flag bit 0: No received frame error 1: Receive frame error UART mode Fe setting condition • The stop bit of the received data frame is low, and only the first stop bit is checked in the case of two stop bits. • FE zeroing condition • Clear register CR1.CFE bit write 1 Note: In UART mode, Re = 0 does not reset the PE position Data received at FE = 1 will be retained but RIinterrupt will not occur and cannot continue to receive data after FE = 1 Smart Card Pattern Fe setting condition	R

			Sampled low error signal flag • FE zeroing condition • Clear register CR1.CFE bit write 1 **Note:** In smart card mode, RE=0 cannot reset the FE bit	
b0	PE	Received data verification	Received data check error flag 0: No received data check error 1: Received data check error PE setting condition When a parity error occurs in the received data PE zeroing condition • Clear register CR1.CPE bit write 1 **Note:** RE=0 does not reset the PE bit Data received at PE = 1 will be retained but RIinterrupt will not occur and cannot continue to receive data after PE = 1	R

26.4.2 Data Register (USART_TDR)

Reset Value: 0x01FF

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res						MPID	TDR[8:0]								

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b15~b10	Res	Reserved Bit	Read at “0”, write at “0”	RW
b9	MPID	Multiprocessor mode ID bit	Send communication data or select bits for sending IDs in multiprocessor mode 0: Send data 1: Sending ID **Note:** MPID bits are only valid in multiprocessor mode and other modes must be set to reset	RW
b8~b0	TDR[8:0]	Send Data	Send Data **Note:** The highest TDR [8] is only valid in UART mode and the data length is set to 9 bits	RW

26.4.3 Data Register (USART_RDR)

Reset value: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res							RDR[8:0]								

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b15~b9	Res	Reserved Bit	Read at “0”, write at “0”	RW
b8~b0	RDR[8:0]	Receive Data Register	Receive Data Register **Note:** The highest RDR [8] is only valid in UART mode and the data length is set to 9 bits	R

26.4.4 Baud Rate Register (USART _ BRR)

Reset value: 0x0000 FFFF

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
DIV_ Integer [7:0]								Res	DIV_Fraction[6:0]						

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b16	Res	Reserved Bit	Read at “0”, write at “0”	RW
b15~b8	DIV_Integer[7:0]	Integer frequency division register	Integer frequency division register **Note:** IV _ Integer [7: 0] can only be set at TE = 0 & RE = 0 (Send/Receive Disabled)	RW
b7	Res	Reserved Bit	When writing, you need to write the reset value	RW
b6~b0	DIV_Fraction[6:0]	Fractional divider register	Fractional divider register **Note:** DIV_Fraction[6:0] can only be set when TE=0&RE=0 (transmit/receive disabled), and the set value is only valid when FBME=1.	RW

Table 26-13 Baud Rate Calculation Formula (Invalid Fractional Baud Rate FBME=0)

Mode	Formula for calculating baud rate	Error E (%) calculation formula
UART mode Multiprocessor mode		
Clock synchronization mode		-
Smart Card Pattern		

B: Baud rate unit: Mbps

C: Clock unit set by PR.ULBREN and PR.PSC[1:0] bits: MHz

BCN: CR3.BCN register setting value

Table 26-14 Baud Rate Calculation Formula Valid Fractional Baud Rate FBME=1)

Mode	Formula for calculating baud rate	Error E (%) calculation formula
UART mode Multiprocessor mode		
Clock synchronization mode		-
Smart Card Pattern		

B: Baud rate unit: Mbps

C: Clock unit set by PR.ULBREN and PR.PSC[1:0] bits: MHz

BCN: CR3.BCN register setting value

26.4.5 Control Register 1 (USART_CR1)

Reset value: 0x8000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
SBS	NFE	FBME	ML	Res		CTEND	MS	CLBD	CWKUP	CBE	CRTOF	CORE	Res	CFE	CPE
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
OVER8	Res		M	Res	PCE	PS	TENDIE	TXEIE	TCIE	RIE	SLME	TE	RE	RTOIE	RTOE

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31	SBS	Location of UART Mode Receiving Data Begin Detection	Location of start bit detection mode when UART mode receives data 0: Start bit detection mode is RX pin low level 1: Starting position detection method is RX pin falling edge **Note:** The SBS bit must be kept at the reset value when not in UART mode. SBS bits can only be set at TE = 0 & RE = 0 (Send/Receive Disabled)	RW
b30	NFE	Digital filter enable bit	Digital filter enable bit 0: Disable digital filtering 1: Enable data filtering **Note:** The NFE bit must be kept at the reset value when not in UART mode. NFE bits can only be set at TE = 0 & RE = 0 (Send/Receive Disabled)	RW
b29	FBME	Fractional baud rate function enable	Fractional baud rate function enable bit 0: Prohibited 1: Enable **Note:** FBME bits can only be set at TE = 0 & RE = 0 (Send/Receive Disabled)	RW
b28	ML	MSB/LSB selection bit	MSB/LSB mode selection bit in UART mode/clock synchronization smart card mode 0: LSB mode 1: MSB mode **Note:** ML bits can only be set at TE = 0 & RE = 0 (Send/Receive Disabled)	RW
b27~b26	Res	Reserved Bit	Read at “0”, write at “0”	RW
b25	CTEND	TEND clear bit	TEND clear bit 0: Not Zero TEND Mark 1: Clear TEND logo **Note:** Writing 1 to the CTEND bit clears the TEND flag, and reading it returns 0.	
b24	MS	Communication mode selection bit	Communication mode selection bit 0: UART mode 1: Clock synchronization mode **Note:** The MS bit can only be set when TE=0&RE=0 (send/receive disabled), the smart card mode MS needs to write the reset value	RW
b23	CLBD	LBD clear bit	LBD clear bit (only valid for USART_2/USART_3/USART_4) 0: Not zero LBD mark 1: Clear LBD mark **Note:** Writing 1 to the CLBD bit clears the LBD flag, and reading it returns 0.	RW
b22	CWKUP	WKUP clear bit	WKUP clear bit (valid only for USART_2/USART_3/USART_4) 0: Not zero WKUP mark 1: Clear WKUP mark **Note:** Writing 1 to the CWKUP bit clears the WKUP flag, and reading it returns 0.	RW
b21	CBE	BE clear bit	BE clear bit (valid only for USART_2/USART_3/USART_4) 0: Not zero BE mark 1: Clear BE mark **Note:** Writing 1 to the CBE bit clears the BE flag, and reading it returns 0.	RW
b20	CRTOF	RTOF clear bit	RTOF clear bit 0: Not zero RTOF mark 1: Clear RTOF mark **Note:** Writing 1 to the CRTOF bit clears the RTOF flag, and reading it returns 0.	RW
b19	CORE	ORE flag zeroing	ORE flag zeroing 0: Not zeroed ORE logo 1: Clear ORE logo **Note:** Writing 1 to the CORE bit clears the ORE flag, and reading it returns 0.	RW

b18	Res	Reserved Bit	Read at “0”, write at “0”	RW
b17	CFE	Fe mark zeroing	Fe mark zeroing 0: Unzeroed FE logo 1: Reset FE logo **Note:** Writing 1 to the CFE bit clears the FE flag, and reading it returns 0.	RW
b16	CPE	PE mark zeroing	PE mark zeroing 0: Not Zero PE Mark 1: Clear PE logo **Note:** Writing 1 to the CPE bit clears the FE flag, and reading it returns 0.	RW
b15	OVER8	UART over-sampling mode	UART over-sampling mode setting, i.e. the basic number of clocks during one-bit data transmission 0: 16 bits 1: 8 bits **Note:** The OVER8 bit must be kept at the reset value when not in UART mode. OVER8 bits can only be set at TE = 0 & RE = 0 (Send/Receive Disabled)	RW
b14~b13	Res	Reserved Bit	Read at “0”, write at “0”	RW
b12	M	Data length setting	In UART mode, transmit/receive data length setting 0: 8 bits 1: 9 bits **Note:** The M bit must be kept at the reset value when not in UART mode M bits can only be set when TE = 0 & RE = 0 (Send/Receive Disabled)	RW
b11	Res	Reserved Bit	Read at “0”, write at “0”	RW
b10	PCE	Check enable bit	Parity enable bit in UART mode 0: No check 1: Verification **Note:** The PCE bit must be 1 in smart card mode, and the PCE bit must maintain the reset value in clock synchronous mode PCE bits can only be set at TE = 0 & RE = 0 (Send/Receive Disabled)	RW
b9	PS	Check Bit	Parity selection bit in UART mode 0: Even check 1: Parity **Note:** PS bits can only be set at TE = 0 & RE = 0 (send/receive disabled), and PS bits are valid at PCE = 1	RW
b8	TENDIE	Send complete interrupt	Send complete interrupt 0: TENDI interrupt request invalid, TENDI interrupt does not occur 1: TENDI interrupt request enabled, TENDI interrupt occurs	RW
b7	TXEIE	Send data register empty interrupt enable bit	Send data register empty interrupt enable bit 0: Tlinterrupt request is invalid, Tlinterrupt does not occur 1: Tlinterrupt request is valid, Tlinterrupt occurs **Note:** When TE=0, write TXEIE=1, then you need to wait for TI interrupt to occur when TE=1. If TXEIE=1 is written when TE=1, it is necessary to wait until SR.TC=1 before writing	RW
b6	TCIE	Send complete interrupt	Send complete interrupt 0: TClinterrupt request is invalid, TClinterrupt does not occur 1: TClinterrupt request enabled, TClinterrupt occurs **Note:** Reading in non-smart card mode valid, reading in smart card mode is 0	RW
b5	RIE	Receive interrupt	Receive interrupt 0: Invalid receive interrupt request, RI and EI interrupt do not occur 1: Receive interrupt request is valid, RI and EI interrupt occur	RW
b4	SLME	Silent mode enable bit	Silent mode enable bit when multiprocessor operation 0: Normal mode 1: Silent mode When SLME = 1, the communication data whose MPB bit is 0 will not be read from the receiving shift register to the receiving register, and the error flags ORE and FE bit are not set. When receiving ID data with MPB 1, SLME automatically clears and starts normal data receiving action. **Note:** The SLME bit is only valid in UART multiprocessor mode and must remain reset in other modes.	RW
b3	TE	Transmitter energy position	Transmitter energy position 0: Transmitter prohibited 1: Transmitter enable **Note:** TE bits in clock synchronization mode can only be written 1 at TE = 0 & RE = 0 (send/receive disabled).	RW

b2	RE	Receiver enable position	Receiver enable position 0: Receiver disabled 1: Receiver enable **Note:** In clock synchronization mode RE bits can only be written at TE = 0 & RE = 0 (send/receive disabled) 1	RW
b1	RTOIE	UART TIMEOUT Interrupt enable bit	UART TIMEOUT Interrupt enable bit 0: UART TIMEOUT interrupt request is invalid, RTOI interrupt does not occur 1： UART TIMEOUT interrupt request is valid, RTOI interrupt occurs	RW
b0	RTOE	UART TIMEOUT function enable bit	UART TIMEOUT Function enable bit 0: UART TIMEOUT Function Prohibition 1: UART TIMEOUT Function enable	RW

26.4.6 Control Register 2 (USART_CR2)

Reset value: 0x0000 0600

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res														SBKM	SBK

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res	LINEN	STOP	CLKC[1:0]		Res		WKUPE	SBKL[1:0]		LBDL	LBDIE	BEE	BEIE	WKUPIE	MPE

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b18	Res	Reserved Bit	Read at “0”, write at “0”	RW
b17	SBKM	Interval sending mode	Interval segment transmission mode selection (only valid for USART_2/USART_3/USART_4) 0: SBK bit write 1 automatically sends the interval segment 1: After writing 1 to the SBK bit, write 0x00 to the USARTn_TDR.TDR register to start sending the interval segment **Note:** In clock synchronization mode RE bits can only be written at TE = 0 & RE = 0 (send/receive disabled) 1	RW
b16	SBK	Interval segment transmission enable bit	Interval segment transmission enable bit (valid only for USART_2/USART_3/USART_4) 0: Do not send interval segments 1: Set the sending interval according to the SBKM bit SBK SBKM Interval segment sending 0 0 Not send 0 1 Not send 1 0 When SBK is written as 1, the interval segment is automatically sent. After the sending is completed, SBK is automatically cleared 1 1 Write data 0x00 to USARTn_TDR.TDR, start sending the interval segment, after sending, SBK is automatically cleared	RW
b15	Res	Reserved Bit	Read at “0”, write at “0”	RW
b14	LINEN	LIN function enabled	In UART mode, LIN function enable bit (valid only for USART_2/USART_3/USART_4) 0: Disable LIN function 1: Enable LIN function **Note:** LINEN bits can only be set at TE = 0 & RE = 0 (Send/Receive Disabled)	RW
b13	STOP	Stop positioning	Stop bit length setting in UART mode 0: 1 stop bit 1: 2 stop bit **Note:** The STOP bit must be kept at the reset value when not in UART mode. STOP bits can only be set at TE = 0 & RE = 0 (Send/Receive Disabled)	RW
b12~b11	CLKC[1:0]	Clock control bit	**UART Mode** 0b00: Clock generated by timer for internal baud rate generator, clock not output to USATRTn _ CK pin, USARTn _ CK pin can be used as ordinary IO 01: Clock generated by timer for internal baud rate generator, clock output to USARTn _ CK pin, output clock frequency and baud rate same 10, 11: The clock source is an external input clock. The frequency of the input clock is 16 times (OVER8=0) or 8 times (OVER8=1) the baud rate. **Clock synchronization mode** 0b00 or 0b01: Clock generated by timer for internal baud rate generator, output to USARTn _ CK pin 0b10 or 0b11: Timer is an external input clock with the same frequency and baud rate **Smart Card Pattern** 00: The clock source is the clock generated by the internal baud rate generator, the clock is not output to the CK pin, and the CK pin can be used as a normal IO 01: The clock source is the clock generated by the internal baud rate generator, and the clock is output to the CK pin Other: prohibitions **Note:** CLKC [1: 0] bits can only be set at TE=0&RE=0 (Send/Receive Disabled)	RW
b10~b9	Res	Reserved Bit	The reset value 0b11 needs to be maintained, and 0b11 is written when writing	RW
b8	WKUPE	LIN wake-up signal detection enable bit	LIN wake-up signal detection enable bit (valid only for USART_2/USART_3/USART_4)	RW

			0: Disable wake-up signal detection 1: Enable wake-up signal detection **Note:** Only when the system needs to detect a wake-up signal, set WKUPE to 1. After detecting the wake-up signal, the software needs to clear WKUPE	
b7~b6	SBKL[1:0]	LIN interval low level width setting bit	LIN interval low level width setting bit (only valid for USART_2/USART_3/USART_4) 00: 10 bits 01: 11 bits 10: 13 bits 11: 14 bits **Note:** The SBKL bit must be kept at the reset value when not in UART-LIN mode. SBKL bits can only be set at TE = 0 & RE = 0 (Send/Receive Disabled)	RW
b5	LBDL	LIN interval low level width detection threshold setting bit	LIN interval low level width detection threshold setting bit (only valid for USART_2/USART_3/USART_4) 0: ≥10 digits 1: ≥11 digits **Note:** The LBDL bit must be kept at the reset value when not in UART-LIN mode. LBDL bits can only be set at TE = 0 & RE = 0 (Send/Receive Disabled)	RW
b4	LBDIE	LIN Break Segment Detection Interrupt Enable Bit	LIN Break Segment Detection Interrupt Enable Bit (valid only for USART_2/USART_3/USART_4) 0: LIN break segment detection interrupt request invalid and no interrupt occurs 1: LIN interval detection interrupt request valid and an interrupt occurs **Note:** The LBDIE bit must be kept at the reset value when not in UART-LIN mode.	RW
b3	BEE	LIN bus error detection function enable bit	LIN bus error detection function enable bit (only valid for USART_2/USART_3/USART_4) 0: Disable LIN bus error detection function 1: Enable LIN bus error detection function **Note:** The BEE bit must be kept at the reset value when not in UART-LIN mode. BEE bits can only be set at TE = 0 & RE = 0 (Send/Receive Disabled)	RW
b2	BEIE	LIN Bus Error Interrupt Enable Bit	LIN Bus Error Interrupt Enable Bit (valid only for USART_2/USART_3/USART_4) 0: LIN bus error interrupt request invalid and no interrupt occurs 1: LIN bus error interrupt request valid and an interrupt occurs **Note:** The LBDIE bit must be kept at the reset value when not in UART-LIN mode.	RW
b1	WKUPIE	LIN wake-up signal detection Interrupt Enable Bit	LIN wake-up signal detection Interrupt Enable Bit (valid only for USART_2/USART_3/USART_4) 0: LIN wake-up signal detection interrupt request invalid and no interrupt occurs 1: LIN wake-up signal detection interrupt request valid and an interrupt occurs **Note:** In non-UART-LIN mode, the WKUPIE bit must be kept at the reset value.	RW
b0	MPE	Multi-processor function	In UART mode, the multiprocessor function enables the bit 0: Prohibited 1: Enable **Note:** The MPE bit must be kept at the reset value when not in UART mode. MP bits can only be set at TE = 0 & RE = 0 (Send/Receive Disabled)	RW

26.4.7 Control Register 3 (USART_CR3)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
DEP	DEM	Res	DEA[3:0]				Res	BCN[2:0]			Res				
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b15	b14
Res		DED[3:0]				CTSE	RTSE	Res		SCEN	LOOP	HDSEL	Res		

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31	DEP	DE Functional Polarity	Driver enable polarity mode 0: The DE signal is high when usart is in IDLE 1: The DE signal is low when usart is in IDLE This bit field cannot be written when USART TE or RE is enabled.	RW
b30	DEM	DE function enabled	Enable driver enable mode This bit is used to activate the external transceiver control through the DE signal 0: DE function disable 1: DE Function Enable When USART TE or RE is enabled, writing to this bit field is prohibited.	RW
b29	Res	Reserved Bit	Read at “0”, write at “0”	RW
b28~b25	DEA[3:0]	DE Assertion Enable Time	Driver Enable Assertion Time These bits are used to define the time between the activation of the DE (Driver Enable) signal and the start of the start bit. It is expressed in sampling time units (1/8 or 1/16 bit time) and is configured by the OVER8 bit, which can be configured to a maximum of 0x7 or 0xF. Writing to this bit field is prohibited when USART TE or RE is enabled. The minimum configuration is 0. When configured as 0, the DE signal occurs at the same time as the start bit; when configured as 0x7 or 0xF, the DE signal occurs 7 or 15 time units before the start bit	RW
b24	Res	Reserved Bit	Read at “0”, write at “0”	RW
b23~b21	BCN[2:0]	base clock count	In smart card mode, sets the number of base clocks during a one-bit data transfer BCN[2: 0] setting Value the basic number of clocks during one-bit data transmission 000 32 001 64 010 Setting prohibited 011 128 100 Setting prohibited 101 256 110 372 111 Setting prohibited **Note:** The BCN [2: 0] bit must remain reset in non- Smart Card mode BCN [2: 0] bits can only be set at TE=0&RE=0 (Send/Receive Disabled) USART_2/USART_3/USART_4 do not support this function, and the output is 0	RW
b20~b14	Res	Reserved Bit	Read at “0”, write at “0”	RW
b13~b10	DED	DE Assertion Cancel Time	Driver Enable Deassertion Time These bits are used to define the time between the end of the last stop bit in a sent message and the deactivation of the DE (Drive Enable) signal. It is expressed in sampling time units (1/8 or 1/16 bit time) and is configured by the OVER8 bit, with a maximum configurable value of 0x7 or 0xF When USART TE or RE is enabled, writing to this bit field is prohibited. The minimum configurable value is 0. When configured as 0, the DE signal change occurs 2 time units after the end bit is completed; when configured as 0x7 or 0xF, the DE signal occurs 9 or 17 time units after the end bit is completed.	RW
b9	CTSE	CTS function	CTS function 0: CTS function Invalid 1: CTS function valid **Note:** CTSE bits can only be set at TE = 0 & RE = 0 (Send/Receive Disabled)	RW
b8	RTSE	RTS function	RTS function 0: RTS function Invalid 1: RTS function valid **Note:**	RW

			RTSE bits can only be set at TE = 0 & RE = 0 (Send/Receive Disabled)	
b7~b6	Res	Reserved Bit	Read at “0”, write at “0”	RW
b5	SCEN	Smart card pattern enable bit	Smart Card Pattern Enable bit 0: Prohibited Smart Card Pattern 1: Enable Smart Card Pattern **Note:** The SCEN bit must be kept at the reset value when not in smart card mode. SCEN bits can only be set at TE = 0 & RE = 0 (Send/Receive Disabled) USART_2/USART_3/USART_4 do not support this function, and the output is 0	RW
b4	LOOP	LIN loopback mode enable bit	LIN loopback mode enable bit (only valid for USART_2/USART_3/USART_4) 0: Normal mode 1: Loopback mode **Note:** LOOP bits can only be set at TE = 0 & RE = 0 (Send/Receive Disabled)	RW
b3	HDSEL	UART Single Line Half Duplex Mode Enable Bit	UART Single Line Half Duplex Mode Enable Bit 0: UART full duplex mode 1: UART half-duplex mode **Note:** HDSEL bits can only be set at TE = 0 & RE = 0 (Send/Receive Disabled)	RW
b2~b0	Res	Reserved Bit	Read at “0”, write at “0”	RW

26.4.8 Pre-divider Register (USART _ PR)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res											ULBREN	LBMPSC[1:0]		PSC[1:0]	

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b5	Res	Reserved Bit	Read at “0”, write at “0”	RW
b4	ULBREN	Ultra low baud rate enable bit	When the internal clock source is used, the ultra-low baud rate function is enabled. 0: Ultra-low baud rate function invalid 1: Ultra-low baud rate function valid And according to the value of PSC[1:0], the prescaler division value is: 00：PCLK1/128 01：PCLK1/256 10：PCLK1/512 11：PCLK1/1024 **Note:** ULBREN bits can only be set at TE = 0 & RE = 0 (Send/Receive Disabled)	
b3~b2	LBMPSC[1:0]	LIN baud rate measurement counter clock source selection	LIN baud rate measurement counter clock source selection (only valid for USART_2/3/4) 00：PCLK1 01：PCLK1/2 10：PCLK1/4 11：PCLK1/8 **Note:** LBMPSC [1: 0] bits can only be set at TE=0&RE=0 (Send/Receive Disabled)	RW
b1~b0	PSC[1:0]	Pre-divider value	Pre-divider divider selection bit when internal timer 00：PCLK1 01：PCLK1/4 10：PCLK1/16 11：PCLK1/64 **Note:** PSC [1: 0] bits can only be set at TE=0&RE=0 (Send/Receive Disabled)	RW

26.4.9 LIN Baud Rate Measurement Count Register (USART_LBMC)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
LBMC[15:0]															

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b16	Res	Reserved Bit	Read at “0”, write at “0”	RW
b15~b0	LBMC[15:0]	LIN baud rate measurement counter	LIN baud rate measurement result counter (only valid for USART_2/USART_3/USART_4) The LIN slave node uses the clock source selected by the USARTn_PR.LBMPSC register to measure the frequency of the synchronization segment sent by the master node. The measured LIN communication baud rate is obtained by dividing the counting clock frequency selected by USARTn_PR.LBMPSC by the value of LBMC. **Note:** The value of LBMC is meaningful only after the synchronization segment is received.	R

26.4.10 USART1 Filter Control Register (USART1_NFC)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res													USART1_NFE	USASRT1_NFS[1:0]	

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b3	Res	Reserved Bit	Read at “0”, write at “0”	RW
b2	USART1_NFE	USART1 Filter Enable Register	USART_1 Filter Enable Register This register is used to control the switch of the analog filter on the RX line in Stop Mode. 0: Analog filter off 1: Analog filter is on. For the filter range, please refer to the setting of USART1_NFS[1:0].	RW
b1~b0	USART1_NFS	USART1 Filter Selection Register	USART_1 Filter Selection Register This register is used to control the filtering range of the analog filter on the RX line in Stop Mode. 00: Filter Width Level 1 01: Filter Width Level 2 10: Filter Width Level 3 11: Filter Width Level 4 For specific values of each gear, please refer to the RX filtering features in USART1 Stop Mode section in the **Data Sheet of Electrical Features.**	RW

26.5Precautions

26.5.1 UART Considerations

Transmitter

UART mode transmitter send disable (USARTn _ CR1.TE = 0), then the TX pin can be used as a common IO and the output value and direction can be set. If output is 0, frame errors are generated by the receiver, which interrupts the data transfer. If output 1, the receiver cannot detect the start bit to start the data transfer.

Receiver

When a frame error occurs in UART mode, the software detects whether the subsequent RX line is at a low level, thus determining whether the sender wants to interrupt the transmission.

If the receiving data start bit detection mode is low level detection, the receiving error continues to receive all low level data after erasing the error flag, and the receiving error will occur again.

26.5.2 Clock Synchronization Mode Considerations

1) When using an external input clock to send data, the update of USARTn_TDR.TDR needs to be completed before the clock is entered. After data is written, at least one bit of data time needs to be waited before the clock is entered.
2) When data is sent consecutively, the next frame of data needs to be updated before the last bit of the current frame is sent.

26.5.3 Other Considerations

1) In order to prevent the TX communication line Hi-Z from being sent in the Disable state, the following methods can be used:
   - Cable pull-up
   - At the end of sending data, before USARTn_CR1.TE=0, set the TX pin as a normal IO output
   - Before sending data, after USARTn_CR1.TE=1, set IO to TX function

27 Inter-Integrated Circuit (I2C)

27.1 Overview

I2C (Integrated Circuit Bus) is 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, fast mode and FM+ mode are supported. SMBus is also supported.

27.2 Main 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, FM+ mode up to 1Mbps
- 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.
- A maximum of 128 slave mode addresses are supported. Supports 7-bit address format and 10-bit address format. Broadcast call address, SMBus master address, SMBus device default address, SMBus alarm address can be detected
- Answer bits can be automatically determined when sent. Answer bits can be sent automatically upon 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 unanimously interrupt
- 2-stage transmit FIFO and 2-stage receive FIFO

27.3Functional Description

27.3.1 I2C System Block Diagram

27.3.1.1 System Block Diagram

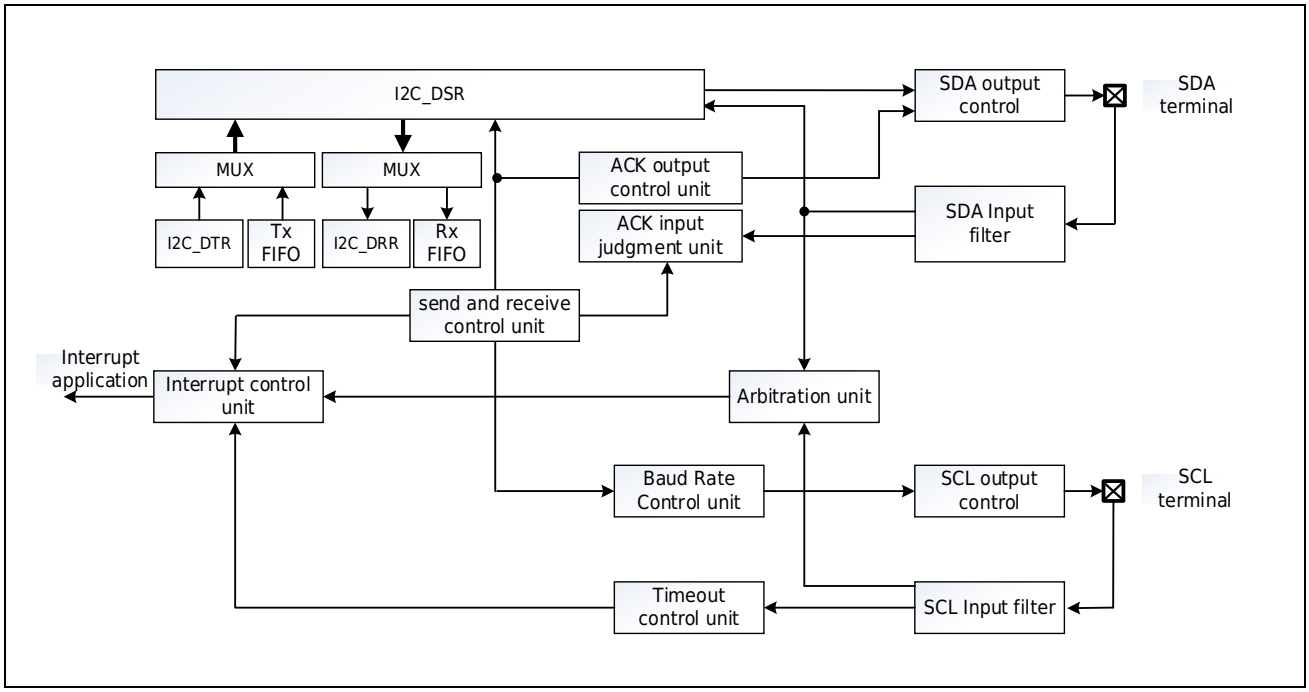


Figure 27-1 I2C System Block Diagram

27.3.1.2 Structural Diagram

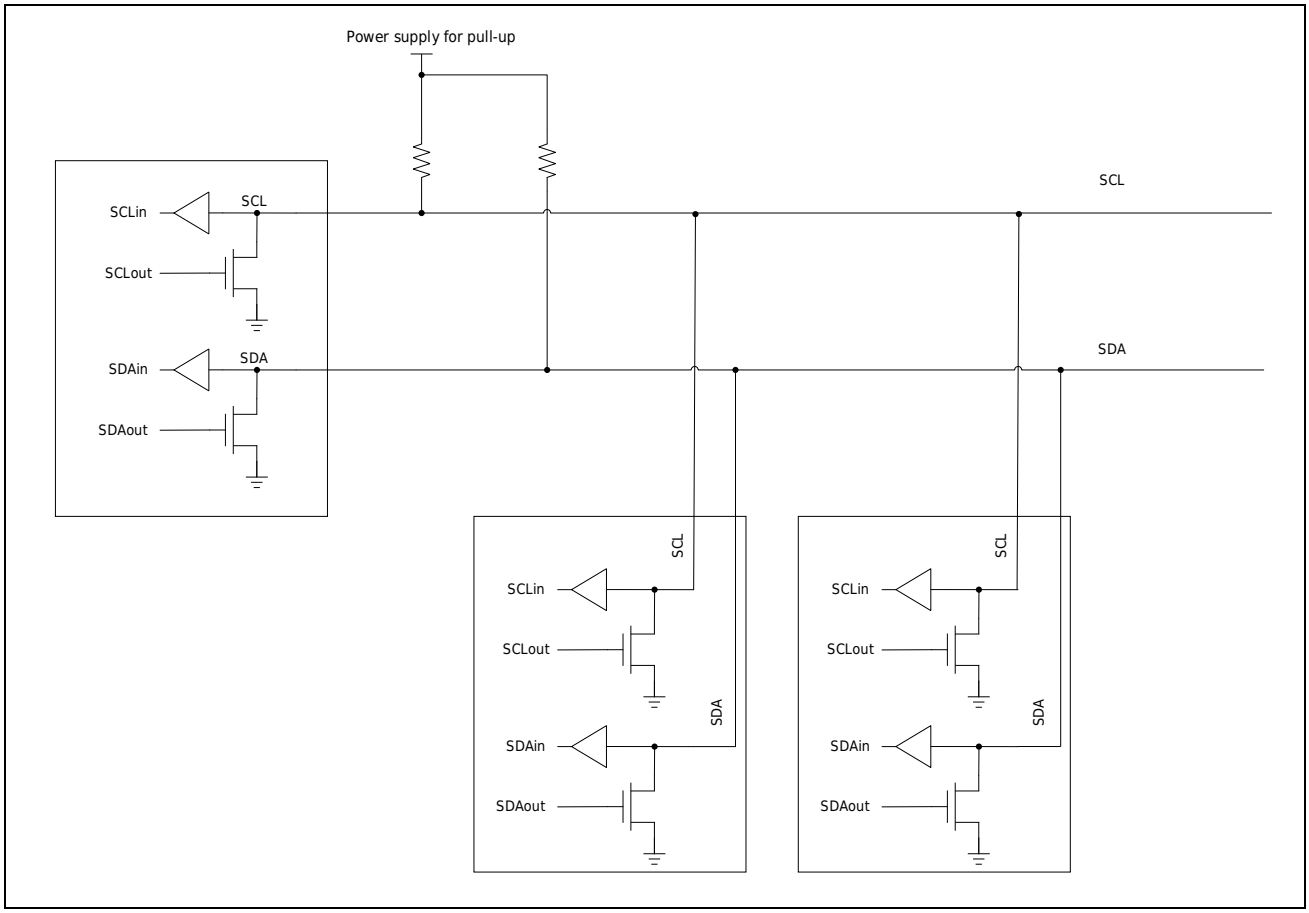


Figure 27-2 Example of I2C Bus Structure

Table 27-1 Input/Output Pin

Pin Name	Input/Output	Function
SCL	Input/Output	Input/output pin of serial clock
SDA	Input/Output	Serial data input/output pin

When the I2C bus is selected, the SCL/SDA input level is the Schmitt level (compatible with CMOS level). When SMBus is selected, please set the SCL/SDA input level to CMOS level by setting the PCRxy register (PCRxy.CINSEL) in the [Universal IO (GPIO)] section.

27.3.2 I2C protocol

I2C bus consists of a clock line (SCL) and a data line (SDA). All connection devices must be drain open output. SCL, external pull-up resistance of SDA line. Resistance depends on system application.

In universal, a complete communication process consists of the following four parts:

1) Start condition
2) Address transfer
3) Data transfer
4) Stop condition

The following figure is a Sequence Diagram of the I2C bus.

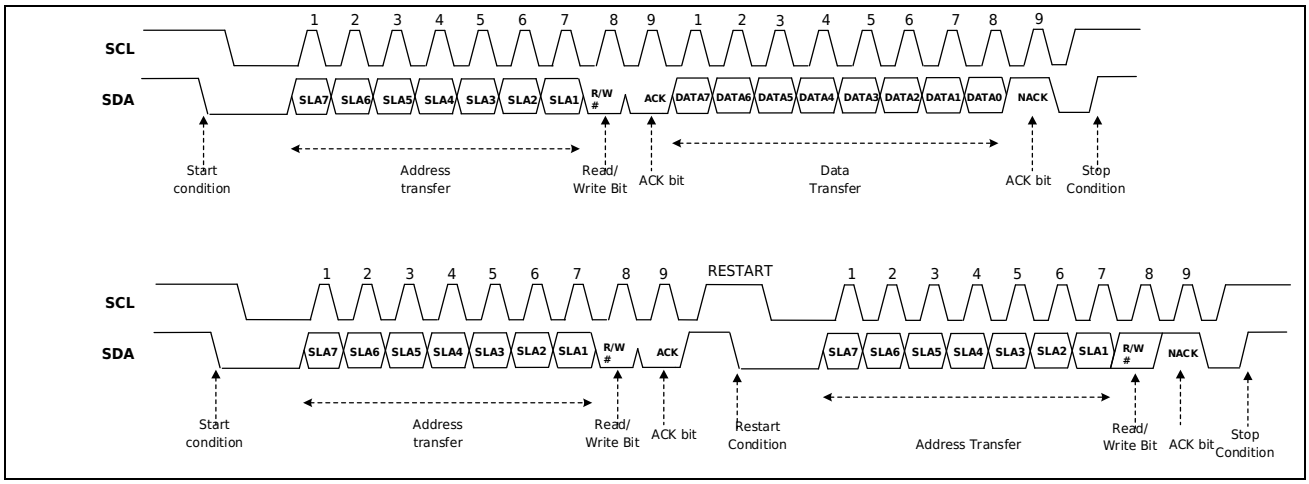


Figure 27-3 I2C Bus Timing Diagram

27.3.2.1 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. Devices on the bus in idle state can start the communication by sending the start condition.

If the START bit is set to 1 while the I2C_SR.BUSY flag is 0 (bus idle), a start condition is sent. If a start condition is detected, the I2C_SR.BUSY flag and the I2C_SR.STARTF flag are automatically set to 1, and the START bit is automatically cleared to 0. At this time, if the SDA signal sent when the START bit is 1 is the same as the signal state of the SDA line, and the start condition is detected, it is considered that the start condition is correctly sent through the START bit, and then the I2C_SR.MSL bit and I2C_SR.TRA bit are automatically set to 1 and then changed to the master control transmission mode. In addition, I2C_SR.TEMPTYF automatically becomes 1 because the TRA bit is 1. Next write the slave address to the I2C _ DTRregister and send the address.

27.3.2.2 Address Transfer

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 valid 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 pattern 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 pattern 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 lower octet slave address.

   Master accept 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 lower octet 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).

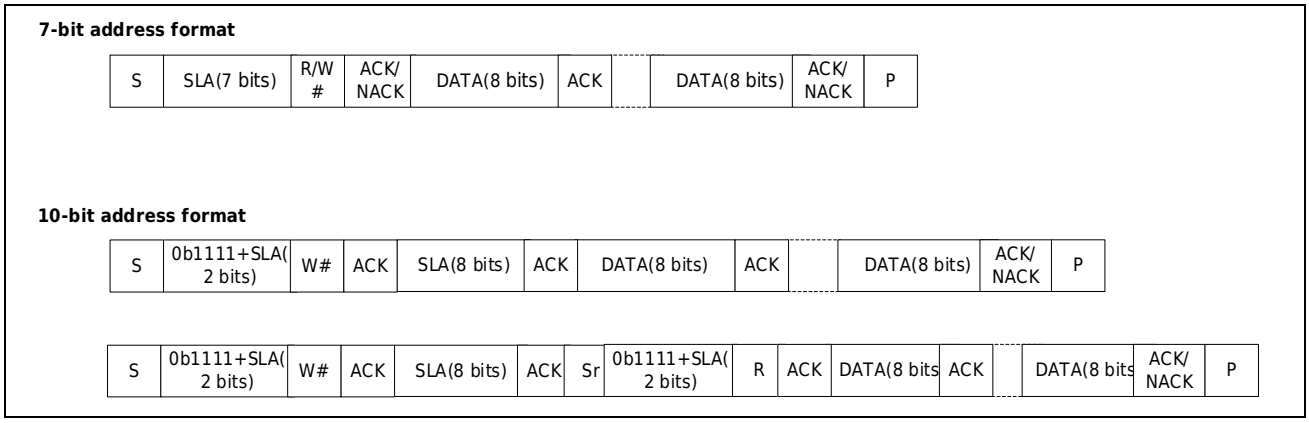


Figure 27-4 I2C Data Format of I²C bus

S: Indicates the start condition.

SLA: Indicates the slave address.

RW#: Indicates the direction of sending and receiving. When RW# is 1, data is sent from the slave to the master; when RW# is 0, data is sent from the master to the slave

Sr: Restarting condition.

DATA: Data sent and received

P: Stop condition.

27.3.2.3 Data Transfer

After the addresses match, the master on the bus transmits data frame by frame according to the direction defined by RW.

The data transmitted after all address frames are treated as data frames. Even a lower 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 cycle. 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 computer on the bus and does not respond to the master in the ninth clock cycle, the slave computer must send NACK. If the master receives data on the bus, NACK is sent in the 9th cycle, 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 conditional release bus
2) Send the restart condition to start a new communication.

Master Sends Data

In master transmit mode, the master outputs SCL clock and transmit data, receives data from the master and returns an answer. The sequence of master sending data is shown in the following figure.

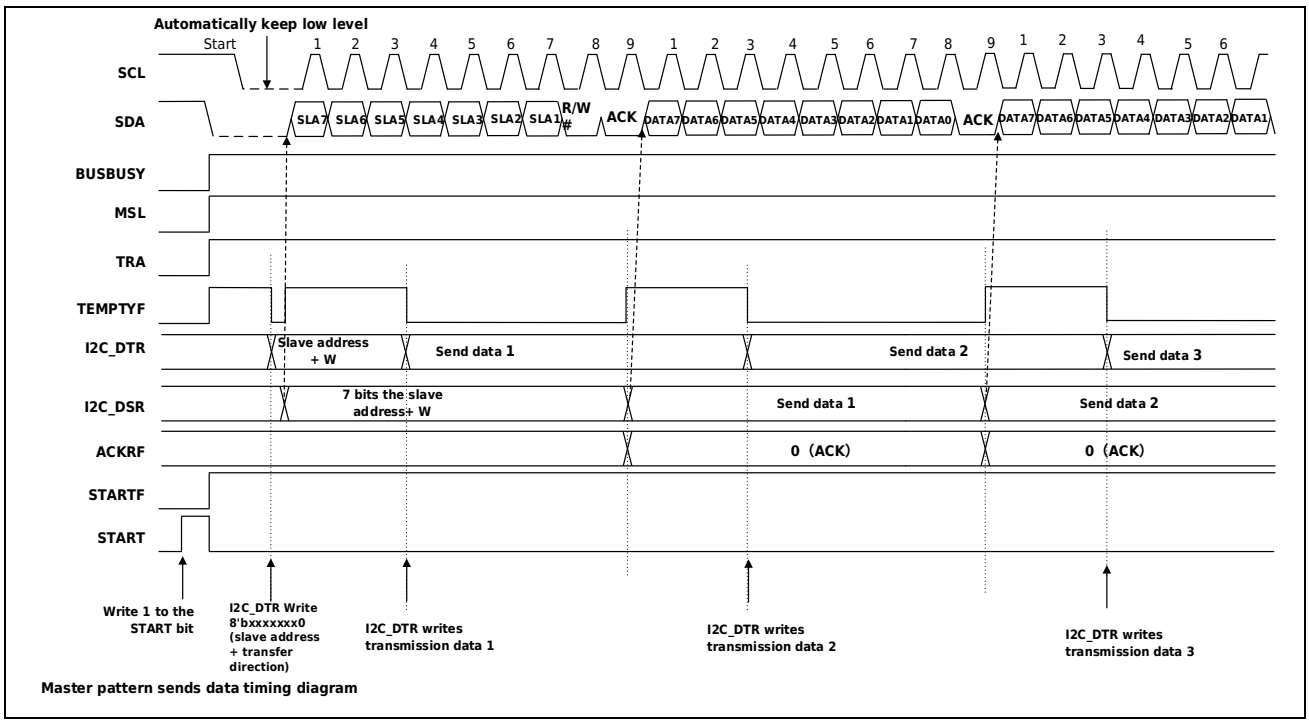


Figure 27-5 Sequence Diagram of Master Sending Data in 7-bit Address Format (Example)

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.

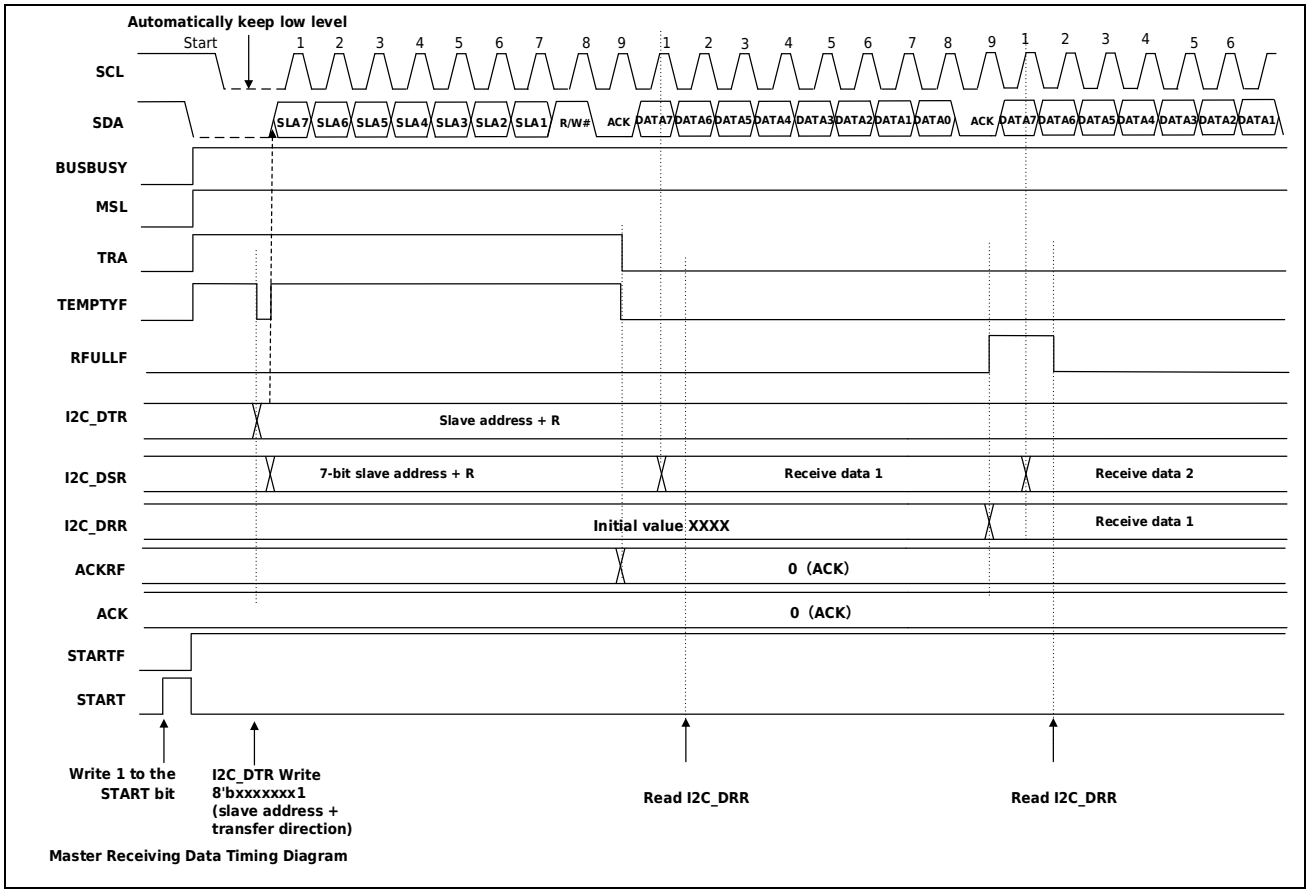


Figure 27-6 Sequence Diagram of Received Data for Master in 7-bit Address Format (Example)

Sending Data from a Slave Computer

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 machine is shown in the following figure.

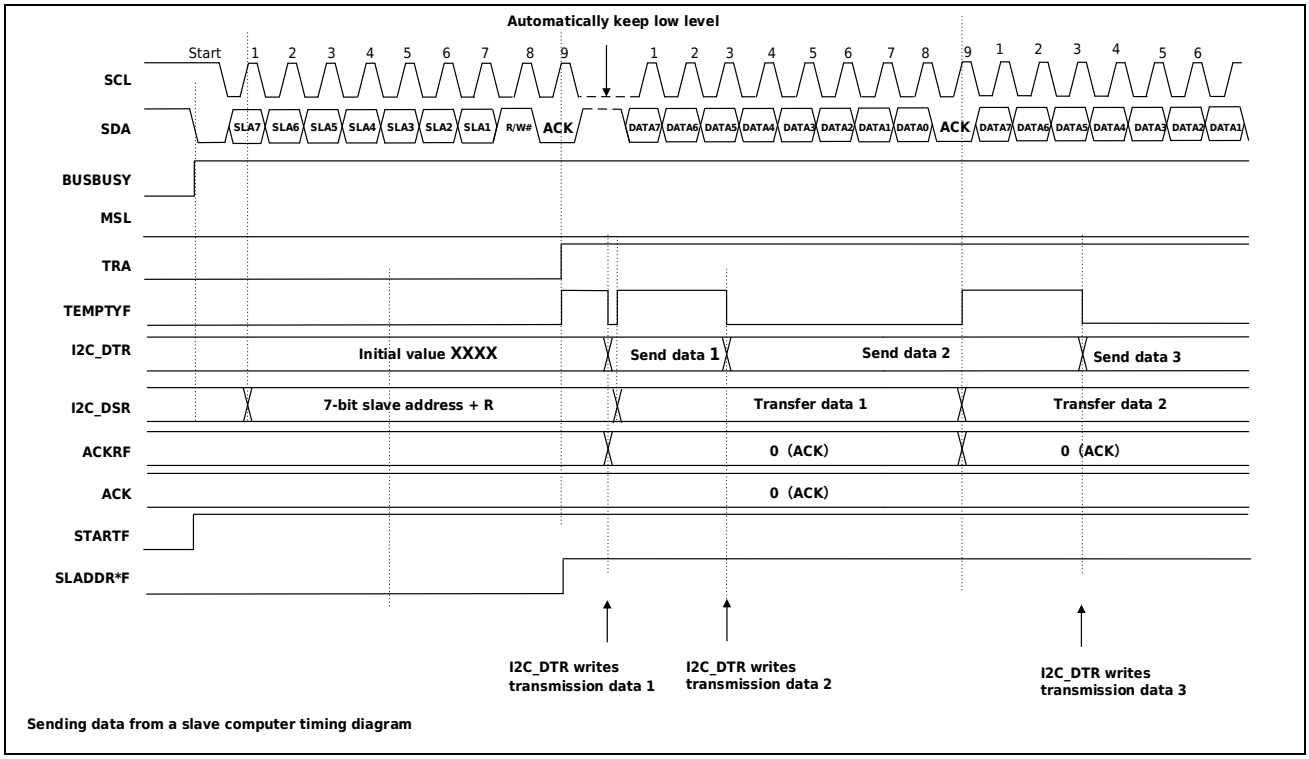


Figure 27-7 Sequence Diagram of Slave Send Mode in 7-bit Address Format (Example)

Receive Data from Machine

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 machine is shown in the following figure.

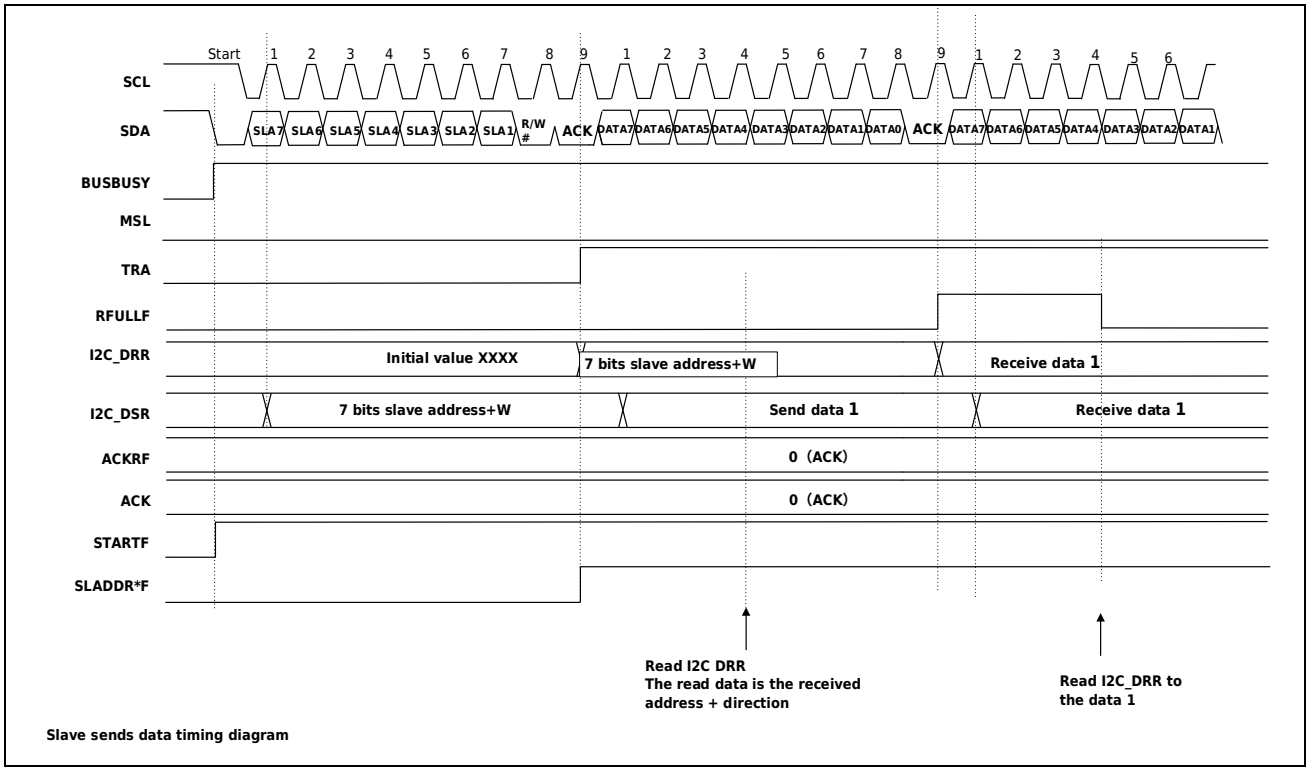


Figure 27-8 Sequence Diagram of 7-bit Address Format Slave Receive Mode (Example)

27.3.2.4 Stop Condition

A stop condition is sent via the I2C_CR1.STOP bit.

Setting the STOP bit to 1 while the I2C_SR.BUSY flag is 1 (bus busy) and the I2C_SR.MSL bit is 1 (master mode) sends a stop condition.

27.3.2.5 Restart Condition

Through I2C _ CR1.The RSTART bit generation restart condition.

When the RSTART bit is set to 1 while the I2C_SR.BUSY flag is 1 (bus busy) and the I2C_SR.MSL bit is 1 (master mode), a row restart condition is generated.

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.

27.3.2.6 SCL Clock Synchronization

When using the I2C bus in multi-master mode, there may be a conflict in the SCL clock 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.SHIGHWregister 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, the low level drive of the SCL line is interruptd 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.

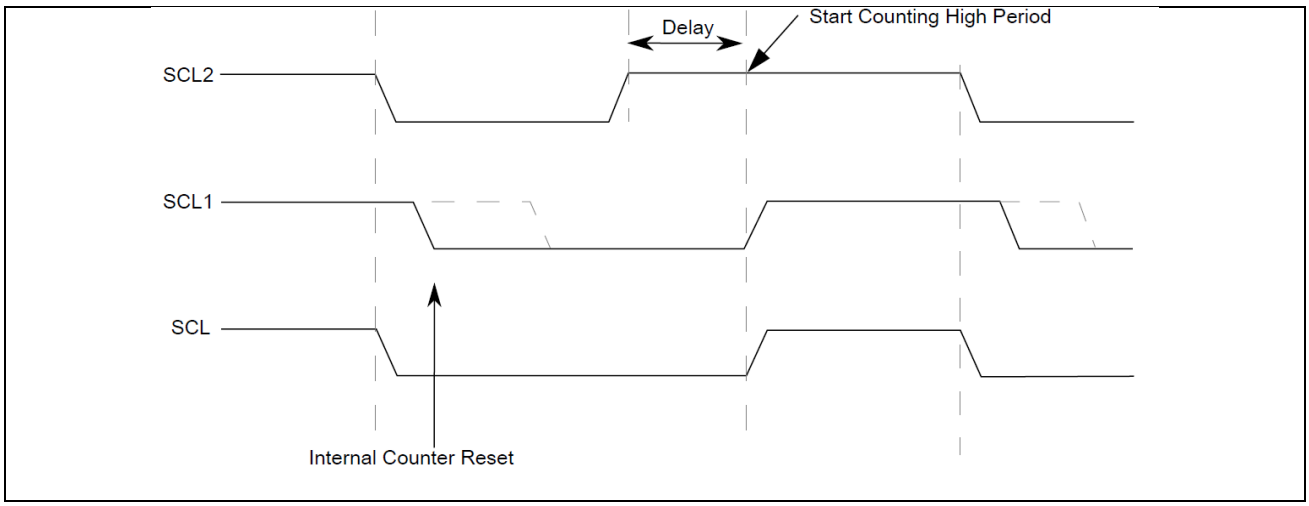


Figure 27-9 SSCL Synchronization Time Series

27.3.2.7 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 cycle of the bus clock depends on the longest low-level clock and the high cycle 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 hardware "1". If a master arbitration fails, it is immediately transferred to the slave receiving mode. At this point, if the slave addresses, including broadcast addresses, match, the slave mode continues to run.

27.3.2.8 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 master.

[Slave Send Mode]

1) In transmit mode (I2C _ SR.TRA bit = 1), if the shift register (I2C _ DSRregister) is empty and the transmit data (I2CDTregister) is not written, the low level of the SCL line is automatically maintained in the low level range of the 9th clock and the next clock, as shown in the following figure.

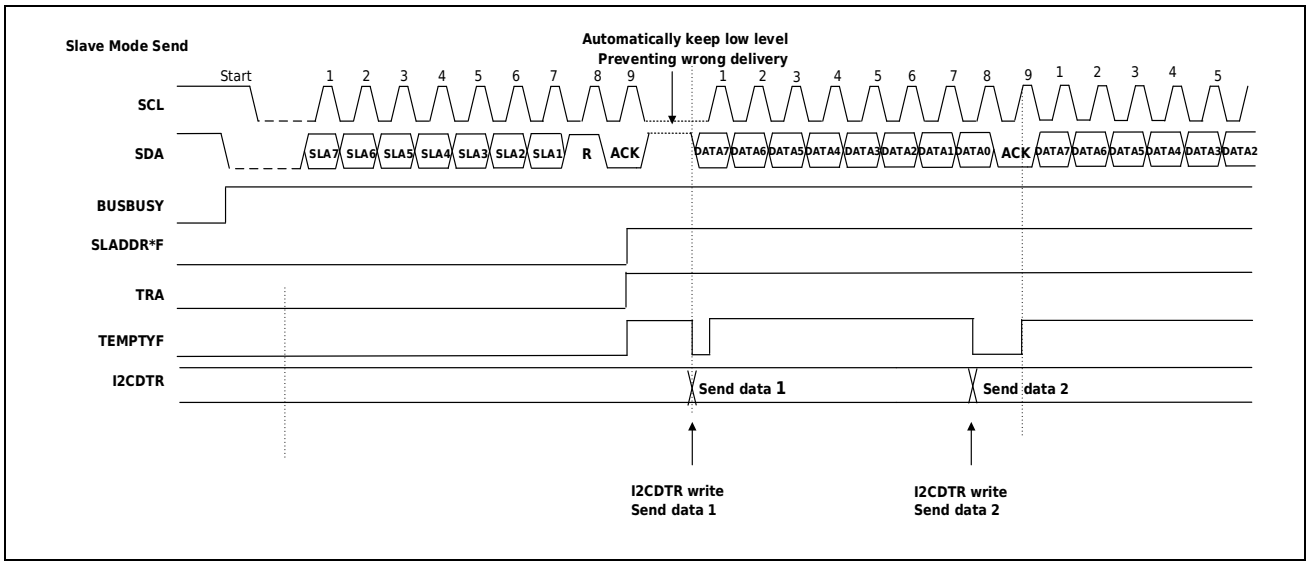


Figure 27-10 Sending Sequence Diagram of Slave (1)

2) After the I2C_SR.NACKF flag becomes 1 or the last transmit data is written to the I2C_DTR register, wait until the I2C_SR.TENDF flag becomes 1 while the I2C_SR.TEMPTYF flag is 1. When the IC_SR.NACKF flag or the TENDF flag is 1, the SCL line is kept low after the 9th clock falls. At

this point, the communication must be interruptd by reading I2C _ DRRregister, thereby releasing the SCL line.

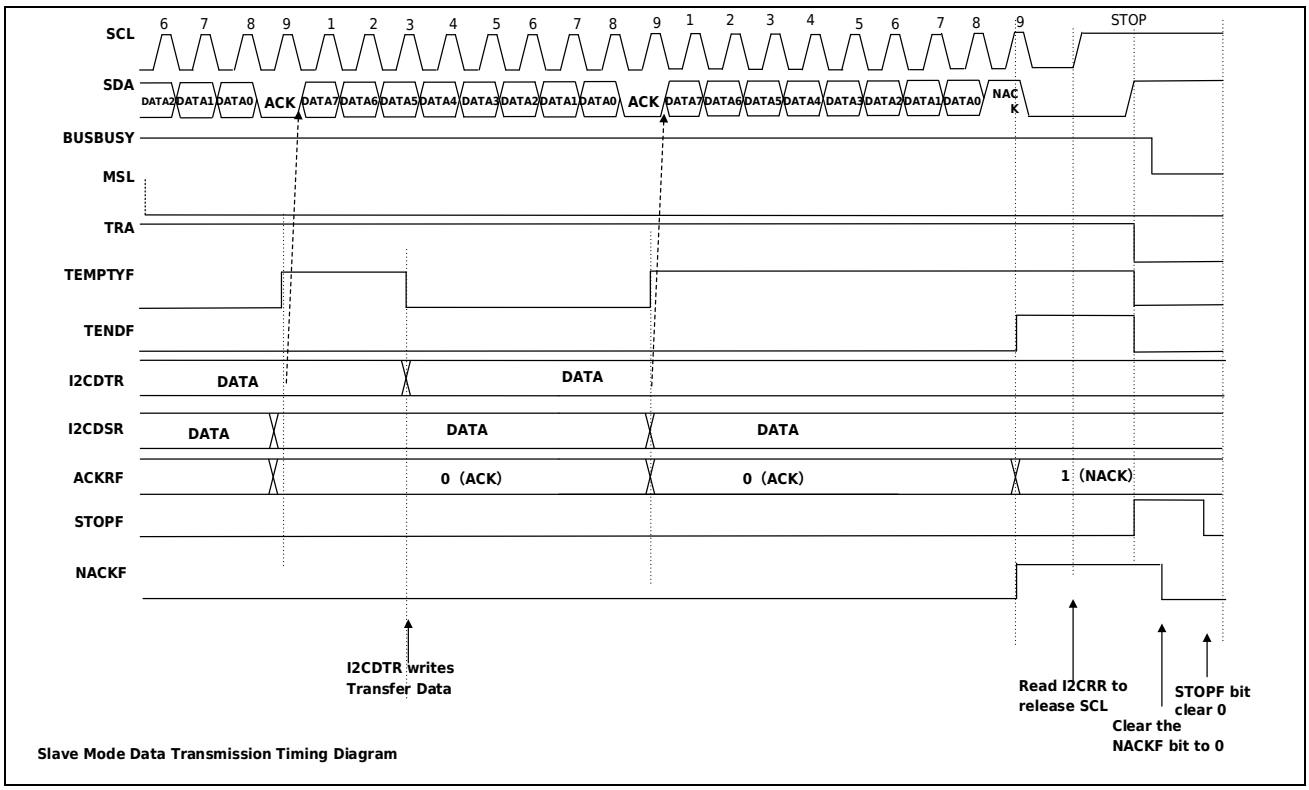


Figure 27-11 Sending Sequence Diagram of Slave (2)

[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 between the 8th SCL and the 9th SCL clock, as shown in the following figure.

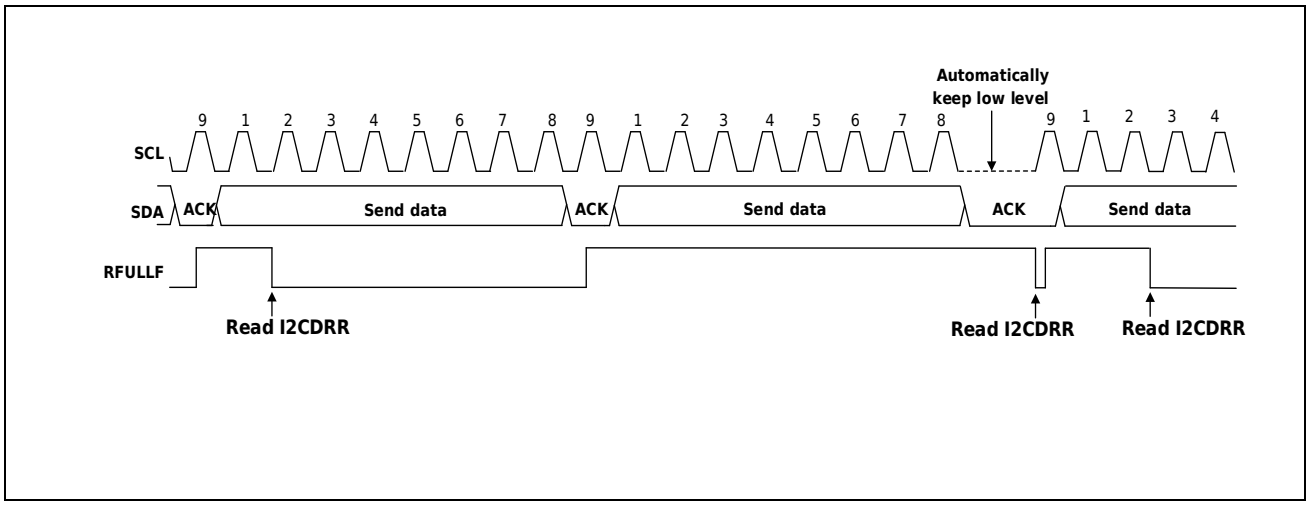


Figure 27-12 Slave Receiving Timing Diagram

[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, the last byte sends an ACK or NACK based on matching. 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. Fast ACK/NACK passes through I2C _ CR3.FACKEN bit control, action sequence is shown in the following figure.

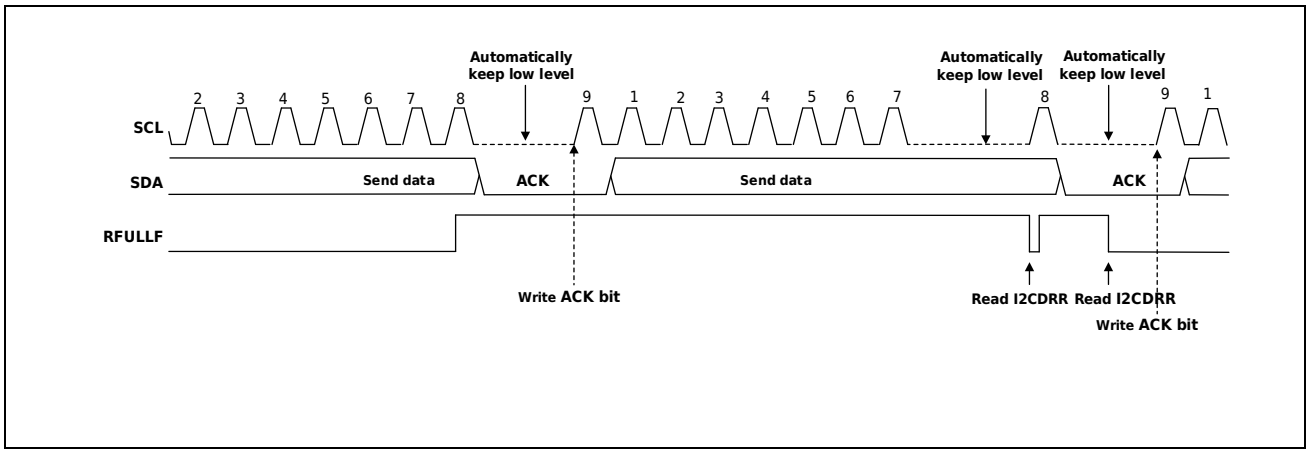


Figure 27-13 Fast ACK/NACK Timing Diagram

27.3.3 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.

27.3.3.1 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, I2C _ SLR1 and I2C _ SLR0register subordinate addresses can be detected.

If the set slave addresses match, set the corresponding SLADDR1F, SLADDR0F to 1 on the rising edge of the 9th clock of the SCL clock, and then set the I2C_SR.RFULLF flag or I2C_SR.TEMPTYF flag to 1 according to the subsequent R/W # bits. As a result, I _ 2C _ SR can be confirmed by either receiving full interrupt or sending null interrupt.The SLADDR1F and SLADDR0F flags determine which slave address is specified.

The timings of I2C_SR. SLADDR1F and SLADDR0F flags changing to 1 are shown in the figure below.

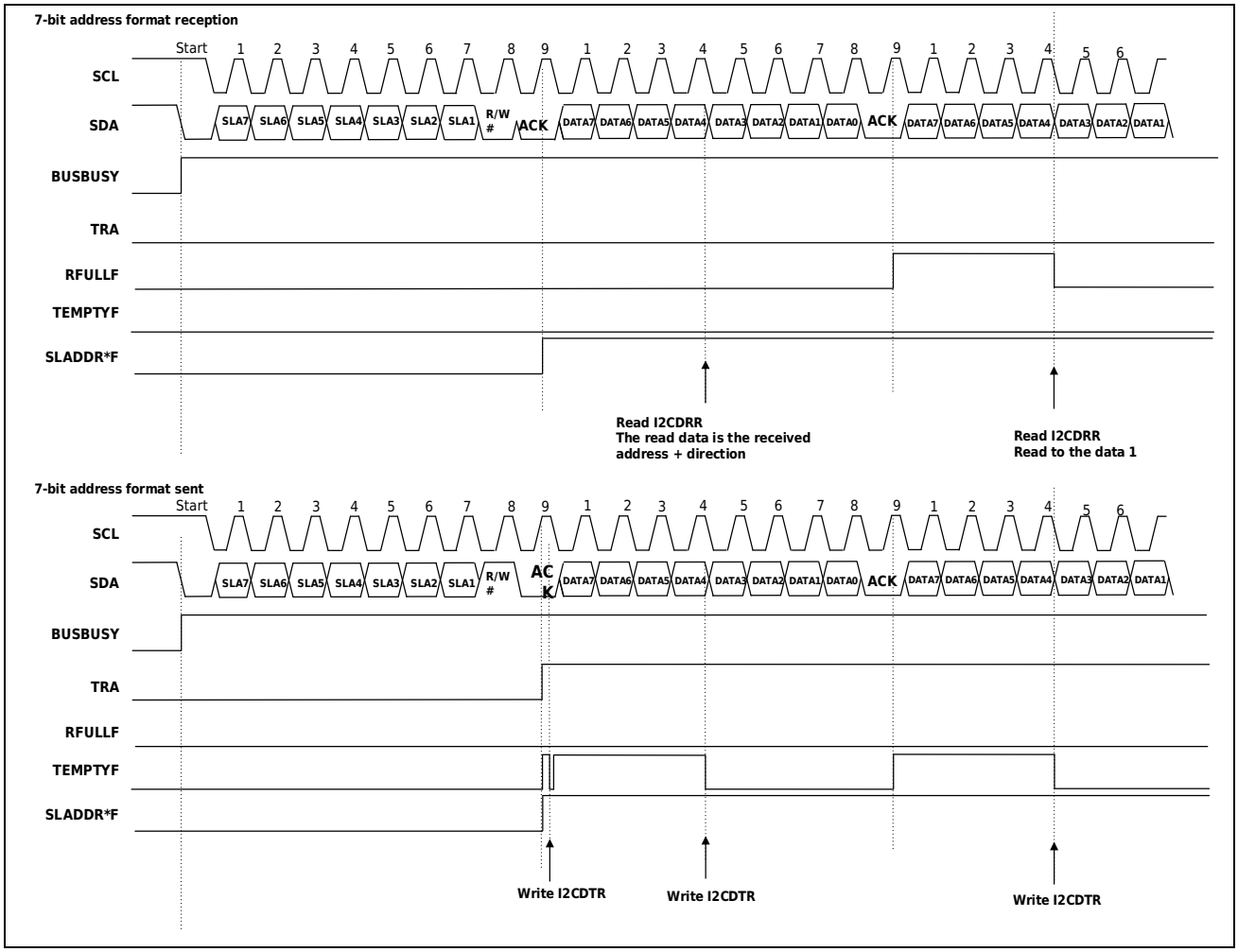


Figure 27-14 Timing when Selecting 7-bit Address Format

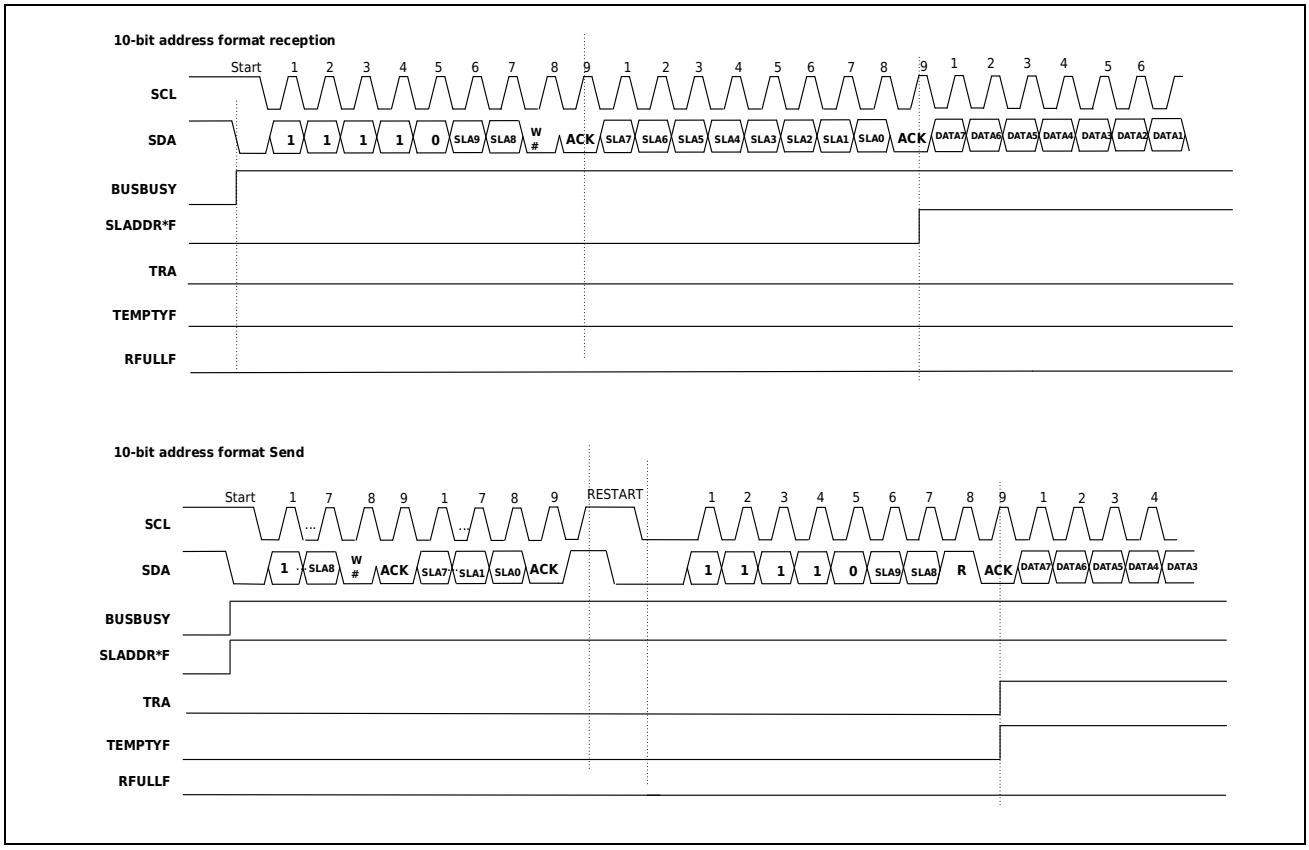


Figure 27-15 Timing when Selecting 10-Bit Address Format

27.3.3.2 Broadcast Address Matching

When the I2C_CR1.GCEN bit is 1, the broadcast address (0b0000 000+0[W]) can be detected.

But after the start condition or restart condition, the address is 0b0000 000+1[R], which is considered the slave address of All "0" and not the broadcast address.

If a broadcast address is matched, the I2C_SR.GENCALLF flag is set to 1 on the rising edge of the ninth clock of the SCL clock.

The broadcast address matches the same run as the normal slave receive run.

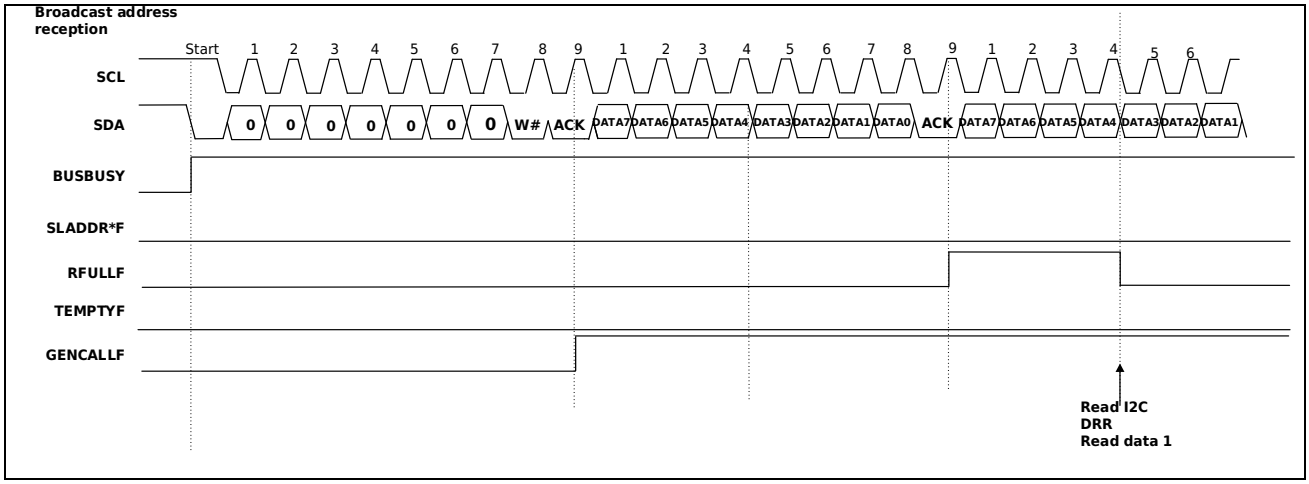


Figure 27-16 Broadcast Address Matching Timing Diagram

27.3.3.3 SMBus Master Address Matching

This product has the master address detection function when SMBus runs. If the I2C_CR1.SMBMASTEREN bit is set to 1 when the I2C_CR1.SMBUS bit is 1, the master address (0b0001 000) can be detected in slave receive mode (I2C_SR.MSL bit TRA is 0b00).

If the SMBUS Master address is detected, set the I2C_SR.SMBHOSTF flag to 1 on the rising edge of the ninth clock of the SCL clock.

Even if the bit following the SMBUS master address (0b0001 000) is the Rd bit (RW# bit receives 1), the SMBUS master address can be detected. The running of SMBUS master address detection is the same as that of normal slave mode.

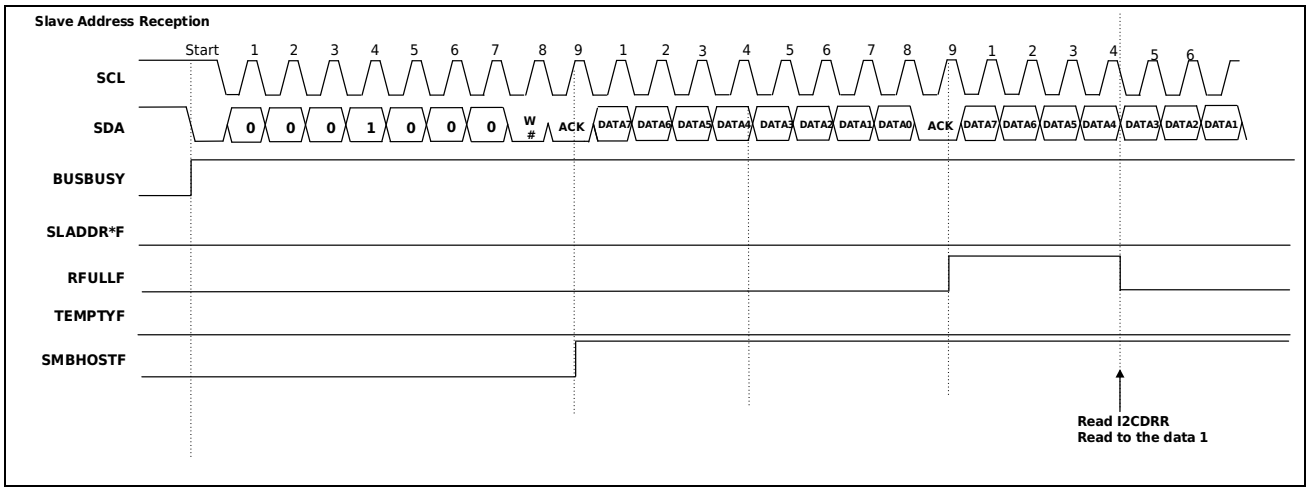


Figure 27-17 SMBus Master Address Matching Timing Diagram

27.3.3.4 SMBus Alarm Response Address Matching

This product has the function of alarm response address detection when SMBus is running. If the I2C_CR1.SMBARLERTEN bit is set to 1 when the I2C_CR1.SMBUS bit is 1, the SMBUS alert response address (0b0001 100) can be detected in slave receive mode (I2C_SR.MSL bit TRA is 0b00).

If an SMBUS alarm response address is detected, the I2C _ SR. SMBALRTF flag is set to 1 on 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.

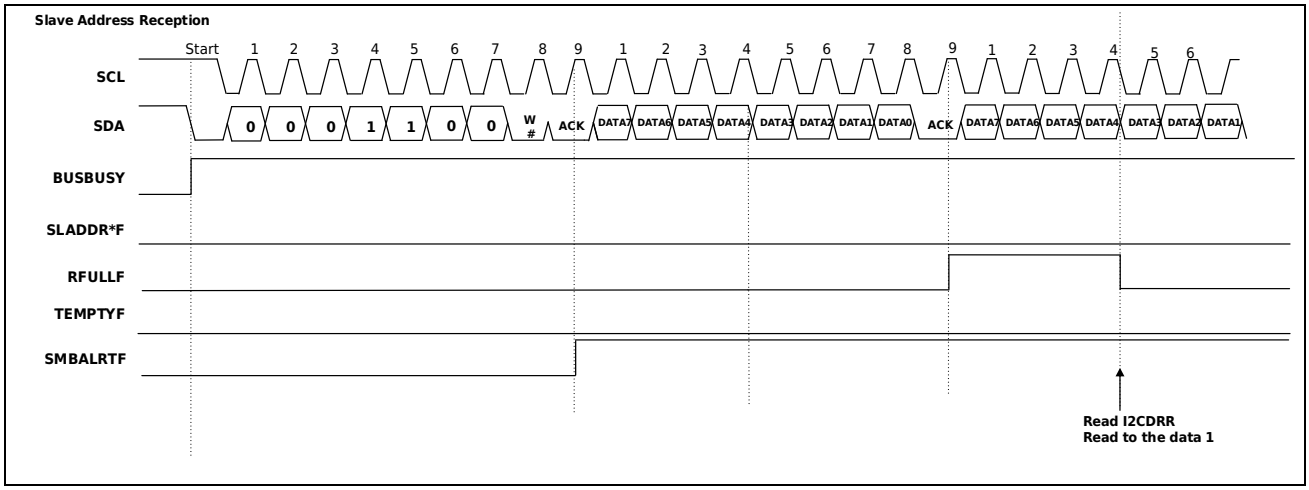


Figure 27-18 SMBus Alarm Response Address Matching Timing Diagram

27.3.3.5 SMBus Default Address Match

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 (0b1100 001) can be detected in slave receive mode (I2C_SR.MSL bit TRA is 0b00).

If the SMBUS default address is detected, the I2C_SR.SMBDEFAULTF flag is set to 1 on the rising edge of the 9th SCL clock.

The SMBUS default address detection runs the same as the normal slave mode.

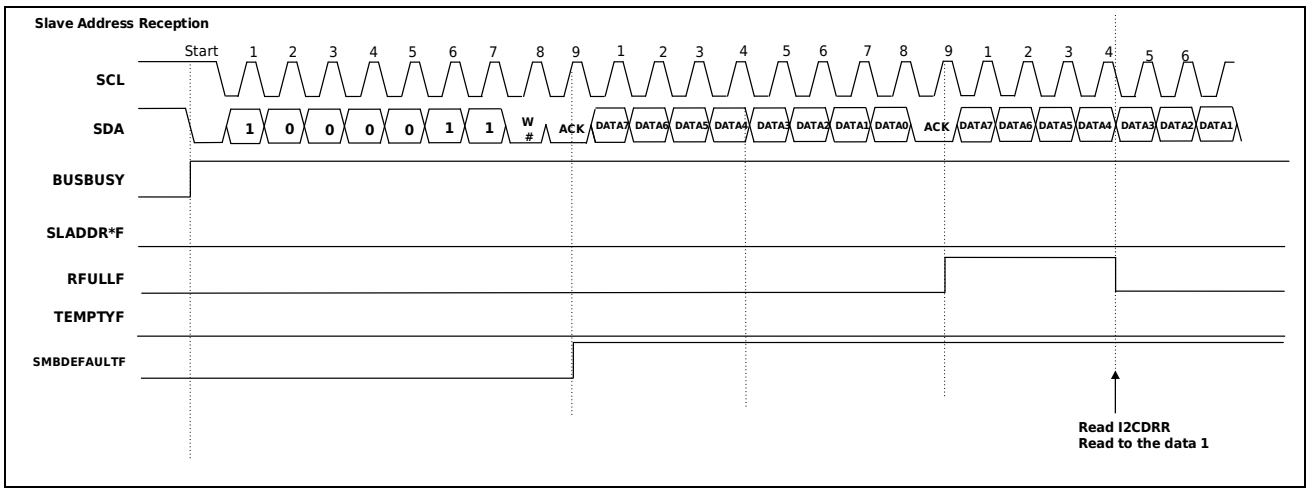


Figure 27-19 SMBus Default Address Timing Diagram

27.3.4 SMBus Action

This I2C interface can be implemented with SMBus (Ver.2.0) communication as a reference. I2C_CR1.SMBUS is required for SMBus communication.SMBUS position 1. The transmission speed must be set within the SMBus specification range of 10Kbps to 100Kbps by setting the I2C_CCR register.

27.3.4.1 SMBus Timeout Measurement

1) SCL Level When 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 period of time, and the abnormal state of the bus is detected.

When operating in the master mode, after the start condition is generated, it is detected that the low level or high level of the SCL line is fixed for more than a certain period of time, and an abnormal state of the bus is detected.

The overtime 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 time-out is detected and an abnormal state of the bus is notified.

For the counting of the internal counter, you 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.

2) Timeout measurement of slave machine

The slave device of SMBus communication needs to measure the interval shown below (timeout interval: $TLOW_{SEXT}$).

- 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] $TLOW_{SEXT}$: 25ms (max).

If the timer measures longer than the timeout of the SMBus specification clock low level detection TIMEOUT: 25 ms (min), the slave computer needs to release the bus.

3) Timeout Measurement of Master

The main control unit of SMBus communication needs to measure the interval shown below (timeout interval: $TLOW_{MEXT}$).

- 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 zones. The timeout time must be within 10 ms (max) of the

accumulated time [Master] TLOW: MEXT: 10 ms (max) of the clock low level in the SMBus specification, and the cumulative time from the start condition to the stop condition must be within 25 ms (max) of TLOW: SEXT: 25 ms (max).

If the time measured by the timer exceeds the accumulated time of the clock low level in the SMBus specification [main control device] TLOW: MEXT: 10ms (max), or the accumulated time exceeds the timeout of the clock low level detection in the SMBus specification (TIMEOUT: 25ms (min)), the Master needs to abort processing. When sending a master, you must immediately abort the sending (write I2C _ DTRregister). Abort the master transaction by sending a stop condition.

27.3.4.2 Packet Error Code (PEC)

In communication, the CPU calculates the CRC, sends the packet error code (PEC) of the SMBus, or checks the received data.

27.3.5 Reduction

It has the function of resetting the communication module. There are two kinds of resets, one for ICCR2.All registers, including the BBSY flag, are initialized and reset. The other is to remove the slave address matching state and to initialize the internal counter while maintaining various set values.

After reset, the I2C_CR1.SWRST bit must be set to 0.

Because either way, the SCL pin/SDA pin output is released to a high impedance state, it can also be used to unplug the bus from unplanned downtime.

The reset in the slave mode causes disparity with the master device, so try to avoid using it. Note: At reset (I2C_CR1.PE and I2C_CR1.SWRST bit is 0b01) The bus status such as start condition cannot be monitored during the process.

27.3.6 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.

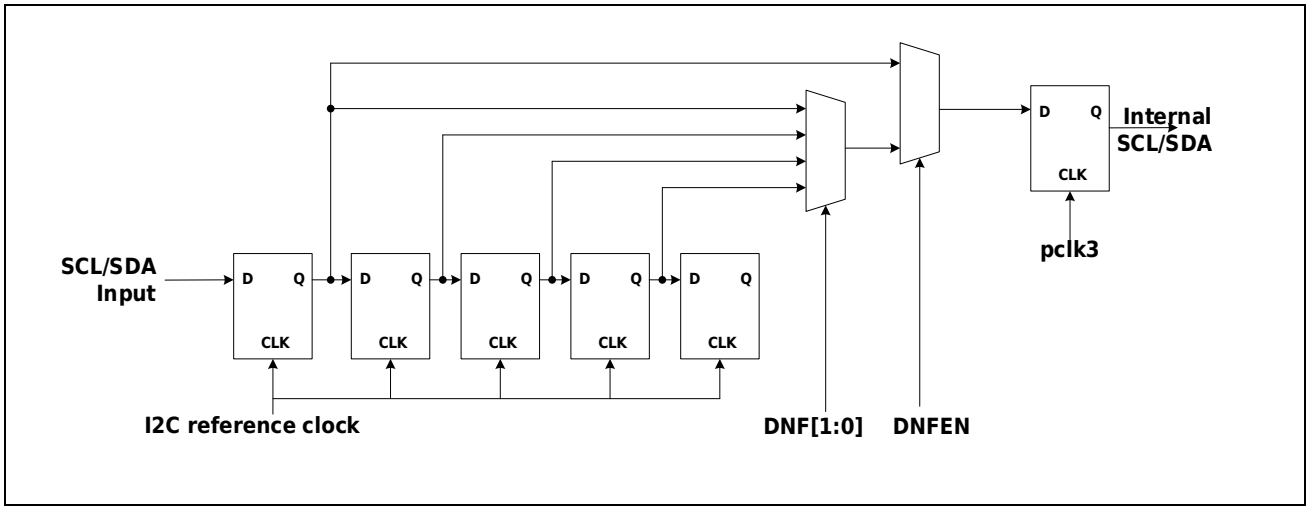


Figure 27-20 Digital Filtering Circuit Block Diagram

The internal digital filter circuit is composed of 4-segment product family-connected trigger circuit and matching detection circuit.

The number of effective segments of the digital filter is selected through the I2C_FLTR.DNF bit. According to the selected effective segment number, the noise elimination capability is 1 to 4 I2C cycles.

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 effective segments set by the I2C_FLTR.DNF bit all matches, the level is used as the internal The signal is transmitted, otherwise it keeps the original value.

27.4Interrupt and Events

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, sending data empty, end of sending.

The interrupt list is as follows.

Table 27-2 Interruption List

Name	Interrupt Source	Interrupt Mark	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
		SMBMASTERF	SMBMASTERF=1& SMBMASTERFIE=1
		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	Received data full	RFULLF	RFULLF=1&RFULLIE=1&~FEN
	Rx FIFO data is not null	RFREQF	RFREQF=1&RFREQIE=1&FEN
I2C_TXI	Send data null	TEMPTYF	TEMPTYF=1&TEMPTYIE=1
I2C_TEI	End of transmission	TENDF	TENDF=1&TENDIE=1

The following table shows the output of the event signal.

Table 27-3 Event Signal Output List

Name	Event source	Event conditions
I2C_EEI	Communication error/communication event	ARLOF=1
		SLADDR0F=1
		SLADDR1F=1
		SMBALRTF=1
		SMBMASTERF=1
		SMBDEFAULTF=1
		GENCALLF=1
		NACKF=1
		TMOUTF=1
		STARTF=1

Name	Event source	Event conditions
		STOPF=1
I2C_RXI	Received data full	RFULLF=1
I2C_TXI	Send data null	TEMPTYF=1
I2C_TEI	End of transmission	TENDF=1

27.5Application Examples

27.5.1 I2C Initialization Process

When starting to send or receive data, you must perform initialization according to the following steps.

PE bit is set to 0.

1. Set SWRST to 1, communication reset;
2. PE bit set to 1, internal state reset;
3. Set slave address format and address;
4. Set baud rate
5. Set control register function and interrupt as needed;
6. SWRST bit is set to 0 to cancel internal state reset;
7. The initialization ends. Can send and receive data.

27.6Register Description

Table 27-4 I2C Base Address

Name	Base Address	Description
I2C	0x4003 B400	I2C base address

Table 27-5 I2C Register List

Register	Description	Offset Address	Bit Width	Reset Value
I2C_CR1	I2C control register 1	0x00	32	0x0000 0040
I2C_CR2	I2C control register 2	0x04	32	0x0000 0000
I2C_CR3	I2C Control Register 3	0x08	32	0x0000 0006
I2C_CR4	I2C control register 4	0x0C	32	0x0030 1B07
I2C_SLR0	I2C slave address register0	0x10	32	0x0000 1000
I2C_SLR1	I2C slave address register1	0x14	32	0x0000 0000
I2C_SLTR	I2C SCL Level Timeout Control Register	0x18	32	0xFFFF FFFF
I2C_SR	I2C state register	0x1C	32	0x0500 0000
I2C_CLR	I2C state reset	0x20	32	0x0000 0000
I2C_DTR	I2C data sending register	0x24	8	0xFF
I2C_DRR	I2C data receiving register	0x28	8	0x00
I2C_CCR	I2C clock control register	0x2C	32	0x0000 1F1F
I2C_FLTR	I2C Filter Control Register	0x30	32	0x0000 0010
I2C_FSTR	I2C FIFO Status Register	0x34	32	0x0000 0000
I2C_SLVADRR	I2C Slave Address Receive Register	0x38	32	0x0000 0000

27.6.1 I2C Control Register 1 (I2C_CR1)

Reset value: 0x0000 0040

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
SWRST	Res				ACK	STOP	START	RESTART	GCEN	Res	SMBMASTEREN	SMBDEFAULTEN	SMBALRTEN	SMBUS	PE

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31-16	Res	Reserved Bit	Read at “0”, write at “0”	RW
b15	SWRST	Software reset	0: Decommission 1: Software Reset Combination of native and PE bits, selecting internal state reset or communication reset SWRST / PE / Reset content 1 / 0 / Communication reset: All registers within I2C (except I2C_FSTR[3:0]) and internal states are reset. 1 / 1 / Internal state reset: I2C_SR (except BUSY bit), I2C_DSR register and internal state machine are reset	RW
b14-11	Res	Reserved Bit	Read at “0”, write at “0”	RW
b10	ACK	Send an answer	0: Reply bit sends' 0 ' (send ACK) 1: Reply bit sends "1" (send NACK)	RW
b9	STOP	Stop condition generation bit	0: Do not generate stop conditions 1: Generation stop condition This bit can be set to 1 and cleared to 0. Hardware zeroing conditions: Stopped condition detected Failure to arbitrate Start condition detected Communication reset	RW
b8	START	Start condition generating bit	0: No starting condition 1: Generation start condition This bit can be set to 1 and cleared to 0. Hardware zeroing conditions: Start condition detected Failure of arbitration Communication reset	RW
b7	RESTART	Repeating start condition generating bit	0: Do not generate repeat start condition 1: Generate repeat start condition This bit can be set to 1 and cleared to 0. Hardware zeroing conditions: 1) Starting condition detected 2) Failure of arbitration 3) Communications resetting	RW
b6	GCEN	Broadcast call is	0: Invalid broadcast address detection 1: Broadcast address detection is valid	RW
b5	Res	Reserved Bit	Read at “0”, write at “0”	RW
b4	SMBMASTEREN	Allow matching of SMBUS master address	0: Do not match SMBUS master address 1: Allow match of SMBUS master address	RW
b3	SMBDEFAULTEN	Allow matching of SMBUS default address bits	0: Do not match SMBUS default address 1: Allow match of SMBUS default address	RW
b2	SMBALRTEN	Allow matching of SMBUS alarm response address bits	0: Disable SMBUS alarm response address 1: Allow SMBUS alarm response address	RW
b1	SMBUS	SMBUS/I2C C bus mode selection bit	0: I2C bus mode 1: SMBUS bus mode	RW

b0	PE	I2C function enabled	0: I2C function disabled 1:I2C C function permissible Combined with SWRST bits, select internal state reset or communication reset	RW

27.6.2 I2C Control Register 2 (I2C_CR2)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res								SMBALRTIE	SMBMASTERIE	SMBDEFAULTIE	GENCALLIE	Res			
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res	TMOUTIE	Res	NACKIE	RFREQIE	Res	ARLOIE	Res	TEMPTYIE	RFULLIE	Res	STOPIE	TENDIE	SLADDR1IE	SLADDR0IE	STARTIE

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b24	Res	Reserved Bit	Read at “0”, write at “0”	RW
b23	SMBALRTIE	SMBUS Alert Response Address Matching Consistent Interrupt Allow	0: SMBUS Alarm Response Address Matching Consistent Interrupt Disable 1: SMBUS Alarm Response Address Matching Consistent Interrupt Allow	RW
b22	SMBMASTERIE	SMBUS Master Address Matching Consistent Interrupt Allowed	0: SMBUS Master Address Matching Consistent Interrupt Disable 1: SMBUS Master Address Matching Consistent Interrupt Allowed	RW
b21	SMBDEFAULTIE	SMBUS Default Address Matching Consistent Interrupt Allowed	0: SMBUS default address match consistent interrupt disable 1: SMBUS Default Address Matching Consistent Interrupt Allowed	RW
b20	GENCALLIE	Broadcast Call Address Matching Consistent Interrupt Allowed	0: Broadcast Call Address Matching Consistent Interrupt Disable 1: Broadcast Call Address Matching Consistent Interrupt Allowed	RW
b19~b15	Res	Reserved Bit	Read at “0”, write at “0”	RW
b14	TMOUTIE	When timeout interrupt permission	0: When timeout interrupt disabled 1: When timeout interrupt permission	RW
b13	Res	Reserved Bit	Read at “0”, write at “0”	RW
b12	NACKIE	NACKinterrupt Allowed	0: NACK interrupt disabled received 1: NACK interrupt allowed received	RW
b11	RFREQIE	FIFO receive request interrupt enable	0: Rx FIFO interrupt request disabled 1: Rx FIFO interrupt request enabled	RW
b10	Res	Reserved Bit	Read at “0”, write at “0”	RW
b9	ARLOIE	Arbitration failure interrupt permission	0: Arbitration failure interrupt prohibition 1: Arbitration failure interrupt permission	RW
b8	Res	Reserved Bit	Read at “0”, write at “0”	RW
b7	TEMPTYIE	Send data null interrupt allowed bit	0: Send data null interrupt disable 1: Null interrupt permission for sending data	RW
b6	RFULLIE	Received data full interrupt allowed bit	0: Full interrupt disablement of received data 1: Receive data full interrupt permission	RW
b5	Res	Reserved Bit	Read at “0”, write at “0”	RW
b4	STOPIE	Stop condition interrupt permission	0: Bus detected stop condition interrupt disable 1: Bus detects stop condition interrupt permit	RW
b3	TENDIE	Send a frame of data end interrupt allow bit	0: Send a frame of data end interrupt disable 1: Send a frame of data to end interrupt permit	RW
b2	SLADDR1IE	Match unanimous interrupt permission for slave address 1	0: Match unanimous interrupt disable from machine address 1 1: Match consistent interrupt permission from machine address 1	RW
b1	SLADDR0IE	Slave Address 0 Matches Uniform Interrupt Allowed	0: Match unanimous interrupt disable from machine address 0 1: Match consistent interrupt permission from machine address 0	RW
b0	STARTIE	Start condition/restart condition interrupt permission	0: Bus detected start condition interrupt disable 1: Bus detected start condition interrupt permit	RW

27.6.3 I2C Control Register 1 (I2C_CR3)

Reset value: 0x0000 0006

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res								FACKEN	Res				HTMOUT	LTMOUT	TMOUTEN

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b8	Res	Reserved Bit	Read at “0”, write at “0”	RW
b7	FACKEN	Time point selection of RFULLF flag position	0: When the 9th SCL clock rises, the RFULLF flag is "1". (At the falling edge of the 8th clock, the SCL line does not keep low level) 1: When the 8th SCL clock rises, the RFULL flag is "1". (On the falling edge of the 8th clock, the SCL line remains low. Unhold low by writing ACK bits.	RW
b6~b3	Res	Reserved Bit	Read at “0”, write at “0”	RW
b2	HTMOUT	Low high level When timeout detection permissible	0: Disable time-out detection when SCL line is high. 1: Time-out detection is enabled when the SCL line is high.	RW
b1	LTMOUT	Low level timeout detection allows	0: In SCL is low level Disable timeout. 1: Time-out detection is enabled when the SCL line is low.	RW
b0	TMOUTEN	When timeout Feature Allowed Bit	0: SCL low level When timeout detected clock function disabled 1: The timeout function of detecting SCL level is enabled	RW

27.6.4 I2C Control Register 4 (I2C_CR4)

Reset value: 0x0030 1B07

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res												SDADLY			
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res			BUSFREE_CLREN	Res	BUSWAIT	Res									

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b22	Res	Reserved Bit	Read at “0”, write at “0”	RW
b21~b20	Res	Reserved Bit	When reading, all bits are "1", and when writing, all bits are written "1".	RW
b19~b16	SDADLY[3:0]	SDA output delay counter	0000: No output delay 0001: 1 I2C reference clock cycle 0010: 2 I2C reference clock cycles 0011: 3 I2C reference clock cycles 0100: 4 I2C reference clock cycles 0101: 5 I2C reference clock cycles 0110: 6 I2C reference clock cycles 0111: 7 I2C reference clock cycles 1000: 8 I2C reference clock cycles 1001: 9 I2C reference clock cycles 1010: 10 I2C reference clock cycles 1011: 11 I2C reference clock cycles 1100: 12 I2C reference clock cycles 1101: 13 I2C reference clock cycles 1110: 14 I2C reference clock cycles 1111: 15 I2C reference clock cycles	RW
b15~b13	Res	Reserved Bit	Read at “0”, write at “0”	RW
b12	BUSFREE_CLREN	Bus Status Clear Bit	0: Disable the I2C bus STOP flag for the I2C_SLVADRR slave receive address register and the SMBALRTF, SMBMASTERF, SMBDEFAULTF，GENCALLF，MSL，SLADDR1F，SLADDR0F，STARTF bit is cleared 1: Enable the I2C bus STOP flag for the I2C_SLVADRR slave receive address register and the SMBALRTF, SMBMASTERF, SMBDEFAULTF,GENCALLF,MSL,TENDF,SLADDR1F, SLADDR0F, STARTF bit is cleared	RW
b11	Res	Reserved Bit	Read at “1”, write at “1”	RW
b10	BUSWAIT	Bus Wait Bit	0: When I2C_DRR is full and I2C_DSR is empty, it will not keep low level between the 9th clock and the 1st clock of the next transmission, and continue to receive the next data. 1: When I2C_DRR is full and I2C_DSR is empty, keep the low level between the 9th clock and the 1st clock of the next transmission, and release the low level by reading the I2C_DRR register.	RW
b9~b8	Res	Reserved Bit	All bits are "1" when reading, and all bits are written "1" when writing	RW
b7~b3	Res	Reserved Bit	Read at “0”, write at “0”	RW
b2~b0	Res	Reserved Bit	All bits are "1" when reading, and all bits are written "1" when writing	RW

27.6.5 I2C Slave Address Register 0 (I2C _ SLR0)

Reset value: 0x0000 1000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res					MASK0EN	MSLADDR0[9:0]									
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
ADDRMOD0	Res		SLADDR0EN	Res		SLADDR0[9:0]									

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b27	Res	Reserved Bit	Read at “0”, write at “0”	RW
b26	MASK0EN	Slave address 0 mask valid bit	0: The mask setting value of slave address register 0 invalid 1: The mask setting value of slave address register 0 valid	RW
b25~b24	MSLADDR0[9:8]	The high bit of a 10-bit slave address. Mask address	Set the high mask address of slave address 0. When the MASK0EN bit is "0", the mask set by MSLADDR0[9:8] invalid. When the MASK0EN bit is "1", the mask set by MSLADDR0[9:8] valid Its effective range is positively correlated with the value set by ADDRMOD0, namely: When the ADDRMOD0 bit is "0", the MSLADDR0[9:8] bits are invalid. When the ADDRMOD0 bit is "1", MSLADDR0[9:8] is the upper two bits of the 10-bit slave address mask address.	RW
b23~b16	MSLADDR0[7:0]	7-bit address/10-bit address low mask address	Set the low mask address of slave address 0. When the MASK0EN bit is "0", the mask value set by MSLADDR0[9:0] invalid. When the MASK0EN bit is "1", the mask value set by MSLADDR0[9:0] valid, and its valid range is positively correlated with the value set by ADDRMOD0, that is: When the ADDRMOD0 bit is "0", MSLADDR0[7:1] is the 7-bit slave address mask and the MSLADDR0[0] bit invalid. When the ADDRMOD0 bit is "1", MSLADDR0[7:0] is the lower 8-bit address mask of the 10-bit slave address.	RW
b15	ADDRMOD0	7-bit/10-bit address format selection bit	0: Select 7-bit address format 1: Select 10-bit address format	RW
b14~b13	Res	Reserved Bit	Read at “0”, write at “0”	RW
b12	SLADDR0EN	Slave address 0 significant bit	0: Invalid set value from machine address register0 1: Valid from machine address register0	RW
b11~b10	Res	Reserved Bit	Read at “0”, write at “0”	RW
b9~b8	SLADDR0[9:8]	The high bit of a 10-bit slave address.	Set the slave address. This bit setting is invalid when ADDRMOD0 bit is "0". When ADDRMOD0 bit is "1", this bit is two digits higher than the 10-bit slave address.	RW
b7~b0	SLADDR0[7:0]	Lower bits of 7-bit address/10-bit address	Set the slave address. When ADDRMOD0 bit is "0", SLADDR0 [7: 1] is the 7-bit slave address. SLADDR0 [0] bit is invalid. When ADDRMOD0 bit is "1", SLADDR0 [7: 0] is the lowest 8-bit address of the 10-bit slave address.	RW

27.6.6 I2C Slave Address Register1 (I2C _ SLR 1)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res					MASK1EN	MSLADDR1[9:0]									

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
ADDRMOD1	Res		SLADDR1EN	Res		SLADDR1[9:0]									

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b27	Res	Reserved Bit	Read at “0”, write at “0”	RW
b26	MASK1EN	Mask Enable Bit	0: Disable the mask function of I2C slave address 1 1: Enable the mask function of I2C slave address 1	RW
b25~b24	MSLADDR1[9:8]	High bit of the 10-bit slave address mask	Set the high mask address of slave address 1. When the MASK1EN bit is "0", the mask set by MSLADDR1[9:8] invalid. When the MASK1EN bit is "1", the mask set by MSLADDR1[9:8] valid Its effective range is positively correlated with the value set by ADDRMOD1, namely: When the ADDRMOD1 bit is "0", the MSLADDR1[9:8] bits are invalid. When the ADDRMOD1 bit is "1", MSLADDR1[9:8] is the upper two bits of the 10-bit slave address mask address.	RW
b23~b16	MSLADDR1[7:0]	7-bit address/low bit of 10-bit address mask	Set the low mask address of slave address 1. When the MASK1EN bit is "0", the mask value set by MSLADDR1[9:0] invalid. When the MAS1EN bit is "1", the mask value set by MSLADDR1[9:0] valid, and its valid range is positively correlated with the value set by ADDRMOD1, namely: When the ADDRMOD1 bit is "0", MSLADDR1[7:1] is the 7-bit slave address mask and the MSLADDR1[0] bit invalid. When the ADDRMOD1 bit is "1", MSLADDR1[7:0] is the lower 8-bit address mask of the 10-bit slave address.	RW
b15	ADDRMOD1	7-bit/10-bit address format selection bit	0: Select 7-bit address format 1: Select 10-bit address format	RW
b14~b13	Res	Reserved Bit	Read at “0”, write at “0”	RW
b12	SLADDR1EN	Slave address 1 significant bit	0: Invalid set value from machine address register1 1: Valid from machine address register1	RW
b11~b10	Res	Reserved Bit	Read at “0”, write at “0”	RW
b9~b8	SLADDR1[9:8]	The high bit of a 10-bit slave address.	Set the slave address. This bit setting is invalid when the ADDRMOD1 bit is "0". When ADDRMOD1 is "1", this bit is two digits higher than the 10-bit slave address.	RW
b7~b0	SLADDR1[7:0]	Lower bits of 7-bit address/10-bit address	Set the slave address. When ADDRMOD1 bit is "0", SLADDR1 [7: 1] is the 7-bit slave address. SLADDR1 [0] bit is invalid. When ADDRMOD1 is "1", SLADDR1 [7: 0] is the lowest 8-bit address of the 10-bit slave address.	RW

27.6.7 I2C SCL Level When Timeout Control Register (I2C_SLTR)

Reset value: 0xFFFF FFFF

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
TOUTHIGH[15:0]															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
TOUTLOW[15:0]															

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b16	TOUTHIGH	SCL High Level When timeout	TOUTHIGH Set SCL High Level When timeout. SCL high level timeout time = TOUTHIGH * I2C reference clock period	RW
b15~b0	TOUTLOW	SCL low Level When timeout	TOUTLOW Set SCL low Level When timeout. SCL low level timeout time = TOUTLOW * I2C reference clock period	RW

27.6.8 I2C Status Register (I2C_SR)

Reset value: 0x0500 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
RF REQ	Res	TFST[1:0]		RF FULL	RF EMPTY	TF FULL	TF EMPTY-	SMBALRTF	SMB MASTERF	SMB DEFAULTF	GENC ALLF	Res	TRA	BUSY	MSL
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res	TMOUTF	Res	NACKF	Res	ACKRF	ARLOF	Res	TEMPTYF	RFULLF	Res	STOPF	TENDF	SLADDR1F	SLADDR0F	STARTF

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31	RFREQ	Rx FIFO request flag bit	0: There is no valid data in the current Rx FIFO 1: The current Rx FIFO has valid data	R
b30	Res	Reserved Bit	Read at “0”, write at “0”	R
b29~b28	TFST	Tx FIFO status flag bit	00: The current Tx FIFO is null 01: The current Tx FIFO has 1 byte of valid data 10: The current Tx FIFO has 2 bytes of valid data 11: Reserved bit	
b27	RFFULL	Rx FIFO full flag bit	0: The current Rx FIFO is not full 1: The current Rx FIFO is full	R
b26	RFEMPTY	Rx FIFO null flag bit	0: The current Rx FIFO is not null 1: The current Rx FIFO is null	R
b25	TFFULL	Tx FIFO full flag bit	0: The current Tx FIFO is not full 1: The current Tx FIFO is full	R
b24	TFEMPTY	Tx FIFO null flag bit	0: The current Tx FIFO is not null 1: The current Tx FIFO is null	R
b23	SMBALRTF	SMBUS alarm response address match consistent flag bit	0: SMBUS alarm response address not matched 1: Detected master address Set the "1" condition: The received address matches 0b0001 100 Clear “0” condition: SMBALRTFCLR writes "1" Stopped condition detected Communication reset Internal state reset	R
b22	SMBMASTERF	SMBUS Master Address Matching Consistent Flag Bit	0: Not matched to SMBUS master address 1: Match to SMBUS master address The consistent conditions for address matching are as follows: Set the "1" condition: The received address matches 0b0001 000 Clear “1” condition: SMBMASTERFCLR writes "1" Stopped condition detected Communication reset Internal state reset	R
b21	SMBDEFAULTF	SMBUS Default Address Matching Consistent Flag Bit	0: SMBUS default address not matched 1: Match to SMBUS default address Set the "1" condition: The received address matches 0b1100 001 Clear “0” condition: SMBDEFAULTFCLR Write "1" Stopped condition detected Communication reset Internal state reset	R
b20	GENCALLF	Broadcast call address matching flag	0: Broadcast call address not matched 1: Match to broadcast call address Set the "1" condition: When the received slave address matches the broadcast call address (All "0") Clear “1” condition: GENCALLFCLR Write "1" Stopped condition detected Communication reset Internal state reset	R
b19	Res	Reserved Bit	Read at “0”, write at “0”	R
b18	TRA	Transmit receive select bit	This bit indicates whether to choose to send or receive data. 0: Receive data	RW

<table>
<tr><td></td><td></td><td></td><td>1: Send data
This bit can be set to 1 and cleared to 0.
Hardware setting "1" condition
Start condition detected
In master mode, the RW bit sent is 0
In slave mode, the address matches and the received RW bit is 1
"0" Condition of Hardware Cleaning
Stopped condition detected
In master mode, the RW bit sent is 1
In slave mode, the address matches and the received RW bit is 0
Communication reset
Internal state reset</td><td></td></tr>
<tr><td>b17</td><td>BUSY</td><td>Bus busy flag bit</td><td>0: Idle state, no communication on bus
1: In possession, the bus is communicating
Set the "1" condition:
Start condition detected on bus
Clear “1” condition:
Bus stop condition detected
Communication reset</td><td>R</td></tr>
<tr><td>b16</td><td>MSL</td><td>Master-slave selection bit</td><td>This bit indicates whether the master is still a slave.
0: Slave mode
1: master mode
The operating mode of I2C is represented by a combination of TRA bits.
<table>
<tr><th>MSL</th><th>TRA</th><th>I2C operating mode</th></tr>
<tr><td>0</td><td>0</td><td>Slave receiving mode</td></tr>
<tr><td>0</td><td>1</td><td>Slave mode</td></tr>
<tr><td>1</td><td>0</td><td>Master receiving mode</td></tr>
<tr><td>1</td><td>1</td><td>Master send mode</td></tr>
</table>
This bit can be set to 1 and cleared to 0.
Hardware setting "1" condition
Start condition detected when START bit is 1
"0" Condition of Hardware Cleaning
1) Stop condition detected
2) Failure of arbitration
3) Communications resetting
Internal state reset</td><td>RW</td></tr>
<tr><td>b15</td><td>Res</td><td>Reserved Bit</td><td>Read at “0”, write at “0”</td><td>R</td></tr>
<tr><td>b14</td><td>TMOUTF</td><td>When timeout flag bit</td><td>0: No SCL low level when timeout detected
1: SCL level when timeout
Set the "1" condition:
SCL does not flip during the period set by I2C_SLTR
Clear “1” condition:
TMOUTFCLR writes "1"
Communication reset
Internal state reset</td><td>R</td></tr>
<tr><td>b13</td><td>Res</td><td>Reserved Bit</td><td>Read at “0”, write at “0”</td><td>R</td></tr>
<tr><td>b12</td><td>NACKF</td><td>Nack flag bit</td><td>0: No NACK Received
1: NACK received
Set the "1" condition:
Receive NACK in send mode
Clear “0” condition:
NACKFCLR writes "1"
Communication reset
Internal state reset</td><td>R</td></tr>
<tr><td>b11</td><td>Res</td><td>Reserved Bit</td><td>Read at “0”, write at “0”</td><td>R</td></tr>
<tr><td>b10</td><td>ACKRF</td><td>Receiving bit</td><td>0: Received ACK bit is "0" (ACK received)
1: Received NACK bit is "1" (ACK received)
Set the "1" condition:
Receive NACK in send mode
Clear “0” condition:
Receive ACK in transmit mode
Communication reset
Internal state reset</td><td>R</td></tr>
<tr><td>b9</td><td>ARLOF</td><td>Arbitration failure flag bit</td><td>0: No arbitration failure
1: Arbitration failure
Set the "1" condition:
Failure to arbitrate</td><td>R</td></tr>
</table>

			Clear “0” condition: ARLOFCLR Write "1" Communication reset Internal state reset	
b8	Res	Reserved Bit	Read at “0”, write at “0”	R
b7	TEMPTYF	Send data null flag bit	0: I2C _ DTRregister full 1: I2C _ DTRregister empty Set the "1" condition: I2C _ DTR Data Transfer to I2C _ DSR TRA position 1 Clear “0” condition: Write I2C_DTR TRA bit clear 0 Communication reset Internal state reset	R
b6	RFULLF	Received data full flag bit	0: I2C _ DRRregister empty 1: I2C _ DRRregister full Set the "1" condition: Received data transferred from I2C DSR to I2C DRR Clear “0” condition: Read I2C DRR RFULLFCLR Write "1" Communication reset Internal state reset	R
b5	Res	Reserved Bit	Read at “0”, write at “0”	R
b4	STOPF	Stop condition flag bit	0: No stop condition detected on bus 1: Bus detected stop condition Set the "1" condition: Stopped condition detected Clear “0” condition: STOPFCLR Write "1" Communication reset Internal state reset	R
b3	TENDF	Send data end flag bit	0: I2C _ DSRregister sending 1: End of I2C _ DSRregister Set the "1" condition: Under the condition of TEMPTYF = 1, the 9th rising edge of SCL is "1" Clear “0” condition: Stopped condition detected Write I2C_DTR TENDFCLR writes "1" Communication reset Internal state reset	R
b2	SLADDR1F	Register 1 Matching Consistency Mark	0: No consistent address detected for slave address register 1 1: Consistent address detected for slave address register 1 Set the "1" condition: When the I2C_SLR1.ADDRMOD1 bit is "0", the received slave address matches I2C_SLR1.SLADDR1[7:1]. When I2C _ SLR 1.When the ADDRMOD1 bit is "1", the first byte address of the 10-bit slave address is received and 0b11110 + I2C _ SLR 1.SLADDR1 [9: 8] matches and the second byte address is I2C _ SLR 1.SLADDR1 [7: 0] matches consistently. Clear “0” condition: Stopped condition detected SLADDR1FCLR Write "1" Communication reset Internal state reset	R
b1	SLADDR0F	Register 0 Matching Consistency Mark	0: No consistent address detected for slave address register 0 1: Detected slave address register 0 consistent address Set the "1" condition: When I2C_SLR0.When ADDRMOD0 is "0", the received slave address and I2C_SLR0. When SLADDR0 [7: 1] matches. When I2C _ SLR 0.When ADDRMOD0 bit is "1", receive the first byte address of 10-bit slave address and 0b11110 + I2C _ SLR0.SLADDR0 [9: 8] matches and the second byte address is I2C _ SLR0.SLADDR0 [7: 0] matches consistently. Clear “0” condition: Stopped condition detected SLADDR0FCLR Write "1" Communication reset Internal state reset	R
b0	STARTF	Start condition/restart condition flag bit	0: Bus not detected starting condition 1: Start condition detected by bus	R

	Set "1" condition 1) Starting condition detected "0" Condition in Qing Dynasty 1) Stop condition detected 2) STARTFCLR writes "1" 3) Communications resetting Internal state reset

27.6.9 I2C State Reset Register (I2C _ CLR)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res								SMBALRTFCLR	SMBMASTERFCLR	SMBDEFAULTFCLR	GENCALLFCLR	Res			

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res	TMOUTFCLR	Res	NACKFCLR	Res	RFREQCLR	ARLOFCLR	Res		RFULLFCLR	Res	STOPFCLR	TENDFCLR	SLADDR1FCLR	SLADDR0FCLR	STARTFCLR

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b24	Res	Reserved Bit	Write "0" on write	W
b23	SMBALRTFCLR	SMBUS Alarm Response Address Matching Consistency Mark Zero bit	Write "1" to clear the SMBALRTF flag bit	W
b22	SMBMASTERFCLR	SMBUS Master Address Matching Consistency Mark Zero bit	Write "1" to clear the SMBMASTERF flag bit	W
b21	SMBDEFAULTFCLR	SMBUS Default Address Matching Consistency Mark Zero bit	Write "1" to clear SMBDEFAULTF flag bit	W
b20	GENCALLFCLR	Broadcast call address matching flag	Write "1" to clear GENCALLF flag bit	W
b19~b15	Res	Reserved Bit	Write "0" on write	W
b14	TMOUTFCLR	When timeout flag bit	Write '1' to clear the TMOUTF flag bit	W
b13	Res	Reserved Bit	Write "0" on write	W
b12	NACKFCLR	Nack flag bit	Write "1" to clear the NACKF flag bit	W
b11	Res	Reserved Bit	Write "0" on write	W
b10	RFREQCLR	Rx FIFO data request flag	Writing "1" clears RFREQF	W
b9	ARLOFCLR	Arbitration failure flag bit	Write '1' to clear the ARLOF flag bit	W
b8~b7	Res	Reserved Bit	Write "0" on write	W
b6	RFULLFCLR	Received data full flag bit	Write "1" to clear RFULLF flag bit	W
b5	Res	Reserved Bit	Write "0" on write	W
b4	STOPFCLR	Stop condition flag bit	Write '1' to clear the STOPF flag bit	W
b3	TENDFCLR	Send data end flag bit	Write '1' to clear the TENDF flag bit	W
b2	SLADDR1FCLR	Zero bit of matching consistency flag from machine address register1	Write "1" to clear the SLADDR1F flag bit	W
b1	SLADDR0FCLR	Zero bit of matching consistency flag from machine address register1	Write "1" to clear the SLADDR0F flag bit	W
b0	STARTFCLR	Start condition/restart condition flag zeroing	Write "1" to remove the STARTF flag	W

27.6.10 I2C Data Sending Register (I2C _ DTR)

Reset Value: 0xFF

b7	b6	b5	b4	b3	b2	b1	b0
DT[7:0]							

If the I2C _ DSRregister is empty, the sending data written to the I2C _ DSRregister is transmitted to the I2C _ DSRregister, and the sending mode starts sending data to the SDA.

The I2C _ DSRregister and I2C _ DTRregister are double buffers. In the course of I2C _ DSRregister data sending, if the data of I2C _ DTRregister is written in advance, the data can be sent continuously.

The I2C_DTR register is readable and writable. Write I2C_DTR register only once when the send data is empty interrupt request occurs.

If the current I2C is working in FIFO mode, the Tx FIFO can be directly accessed by accessing the I2C_DTR register address. If the current TEMPTYF valid, the write operation on the I2C_DTR register address can write the current bus data into the Tx FIFO. When the Tx FIFO valid and the current I2C allows transmission, the Tx FIFO data can be transferred to the I2C_DSR register and start sending data to SDA.

The selection of the send data source is done through the FEN bit in the I2C FIFO status register (I2C_FSTR).

When FEN is 1, the data path from the external data bus to the FIFO and the data path from the FIFO to the I2C_DSR are selected.

When FEN is 0, the data path from the external data bus to the I2C_DTR register and from the I2C_DTR register to the I2C_DSR register are selected.

Non-FIFO mode is selected by default.

27.6.11 I2C Data Receiving Register (I2C_DRR)

Reset Value: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
DR[7:0]							

If a frame of data is received, the received data can be transferred from the shift register (I _ 2C _ DSR) to the I _ 2C _ DRRregister, and then into the next data receiving state.

I2C _ DSRregister and I2C _ DRRregister are double buffers. In the process of receiving I2C _ DSRregister data, if the data of I2C _ DRRregister is read, the data can be received continuously.

Do not write to I2C _ DRRregister. Read I2C _ DRRregister only once when the receive data full interrupt request occurs.

When the I2C_SR.RFULLF flag is 1, if the data in the I2C_DRR register is not read and the next data is received immediately, the SCL clock will automatically remain at a low level after the 7th or 8th falling edge (refer to Figure 27-1Figure 27-13) of the data reception until the data in the I2C_DRR register is read.

When I2C works in FIFO mode, the Rx FIFO can be directly accessed by accessing the I2C_DRR register address.

When I2C works in FIFO mode, if 1 frame of data is received, the received data can be transferred from the shift register (I2C_DSR) to the Rx FIFO. When the Rx FIFO receives valid data, it will output the RFREQ request status signal. At this time, the data in the Rx FIFO can be directly accessed by reading the I2C_DRR register address.

When I2C is working in FIFO mode, when the I2C_SR.RFULLF flag is 1 (that is, the Rx FIFO has stored 2 bytes of data, and the FIFO is full), if the data in the Rx FIFO is not read and the next data is received immediately, the SCL clock will be in the data

After the 7th or 8th falling edge of reception, it automatically remains at a low level until the data in the Rx FIFO is read.

The I2C receive path is selected by the FEN bit in the I2C FIFO status register.

When FEN is 1, the I2C_DSR to Rx FIFO data path and the Rx FIFO to data bus path are selected.

When FEN is 0, the data path from I2C_DSR to I2C_DRR and the path from I2C_DRR to the data bus are selected.

The non-FIFO function is selected by default.

27.6.12 I2C Data Shift Register (I2C _ DSR)

b7	b6	b5	b4	b3	b2	b1	b0
DSR							

I2C _ DSRregister is used to send and receive shift registers of data. I2C _ DSRregister is neither readable nor writable.

At the time of data sending, send data from I2C _ DTRregister to I2C _ DSRregister and send data from SDA pin. At the time of data reception, data is transferred from the I2C _ DSRregister to the I2C _ DRRregister once a frame of data is received.

27.6.13 I2C Clock Control Register (I2C_CCR)

Reset value: 0x0000 1F1F

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res								FMPLUSEN	Res				CKDIV[2:0]		
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
SHIGHW[7:0]								SLOWW[7:0]							

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b24	Res	Reserved Bit	Read at “0”, write at “0”	RW
b23	FMPLUSEN	Fast plus mode enable bit	0: Fast plus mode disable 1: Fast plus mode enable	RW
b22~b19	Res	Reserved Bit	Read at “0”, write at “0”	RW
b18-b16	CKDIV[2:0]	Setting of I2C reference clock frequency	000: I2C reference clock frequency = PCLK0/1 001: I2C reference clock frequency = PCLK0/2 010: I2C reference clock frequency = PCLK0/4 011: I2C reference clock frequency = PCLK0/8 100: I2C reference clock frequency = PCLK0/16 101: I2C reference clock frequency = PCLK0/32 110: I2C reference clock frequency = PCLK0/64 111: I2C reference clock frequency = PCLK0/128	RW
b15~b8	SHIGHW[7:0]	Set SCL high level width bit	Setting the High Level Width of SCL Clock	RW
b7~b0	SLOWW[7:0]	Set SCL low level width bit	Setting the Low Level Width of SCL Clock	RW

Note:

1. In master mode, SHIGHW is used to set the high level width of the SCL clock. The setting is invalid in slave mode.
2. In master mode, SLOWW is used to set the high level width of the SCL clock. In slave mode, the set value is greater than the data preparation time. Data setup time ($t_{SU:DAT}$) is 250ns for standard mode, 100ns for fast mode, and 50ns for fast mode+.
3. Baud rate:
   1) *I2C_FLTR.DNFEN=0，I2C_CCR.CKDIV=0b000*:
      Baud rate =1/{[(SHIGHW+3)+ (SLOWW+3)]/ΦI2C+SCL rise time + SCL fall time}
   2) *I2C_FLTR.DNFEN=1，I2C_CCR. CKDIV=0b000*:
      Baud rate = 1/{[(SHIGHW+3+filter capability)+ (SLOWW+3+filter capability)]/ΦI2C+SCL rise time+SCL fall time}
   3) *I2C_FLTR.DNFEN=0，I2C_CCR. CKDIV!=0b000*:
      Baud rate =1/{[(SHIGHW+2)+ (SLOWW+2)]/ΦI2C+SCL rise time + SCL fall time}
   4) *I2C_FLTR.DNFEN=1，I2C_CCR.CKDIV!= 0b000*:
      Baud rate = 1/{[(SHIGHW+2+filter capability)+ (SLOWW+2+filter capability)]/ΦI2C+SCL rise time+SCL fall time}
4. *The rise time [tr] and fall time [tf] of the SCL line depend on the total capacitance [Cb] and pull-up resistor [Rp] of the bus. For details, please refer to the I2C bus specification.*
5. ΦI2C: I2C divided clock.

27.6.14 I2C Filter Control Register (I2C_FLTR)

Reset value: 0x0000 0010

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res										ANFEN	DNFEN	Res		DNF[1:0]	

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b6	Res	Reserved Bit	Read at “0”, write at “0”	RW
b5	ANFEN	Analog filtering allowed bit	0: Analog filtering function disabled 1: Analog Filtering Allowed	RW
b4	DNFEN	Digital filtering allowed bit	0: Digital filtering disabled 1: Digital filtering is allowed	RW
b3~b2	Res	Reserved Bit	Read at “0”, write at “0”	RW
b1-b0	DNF[1:0]	Selection of Filtering Capabilities for Digital Filters	00: Filter capability 1 I2C reference clock cycle 01: 2 I2C reference clock cycles with filtering capability 10: 3 I2C reference clock cycles with filtering capability 11: 4 I2C reference clock cycles with filtering capability	RW

27.6.15 I2C FIFO Controller Register (I2C_FSTR)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res								RFST		TFST		NACK TFFLUSH		RF FLUSH		TF FLUSHH		FEN

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b7	Res	Reserved Bit	Read at “0”, write at “0”	RW
b7~b6	RFST	Rx FIFO status bits	00: FIFO null 01: FIFO half full 10: FIFO full 11: reserved Communication reset	R
b5~b4	TFST	Tx FIFO status bits	00: FIFO null 01: FIFO half null 10: FIFO full 11: Reserved bit Communication reset	R
b3	NACK_TFFLUSH	NACK resets the Tx FIFO	0: Do not execute TFIFO FLUSH after receiving NACK in slave mode 1: Execute TFIFO FLUSH after receiving NACK in slave mode System Reset	RW
b2	RFLUSH	RX FIFO pointer reset	0: write invalid 1: Reset RX FIFO pointer System Reset	RW
b1	TFLUSH	TX FIFO pointer reset	0: write invalid 1: Reset TX FIFO pointer System Reset	RW
b0	FEN	FIFO function enable bit	0: FIFO function disabled 1: FIFO Function Enable System Reset	RW

27.6.16 I2C Slave Address Receiver Register (I2C_SLVADRR)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res						SLVADR									

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b10	Res	Reserved Bit	Read at “0”, write at “0”	RW
b9~b0	SLVADR	Slave Address Receive Register	6:0: 7-bit slave receiving address 9:0: 10-bit slave receiving address	R

28 Programmable Logic Array (PLA)

28.1 Overview

The configurable logic array provides 256 programmable digital logic operations for external pins, ADC, and timers without CPU intervention. This module implements 16 independent PLA units. Each PLA unit supports the output of GPIO port.

Main Features of PLA:

- 16 independent PLA units, each with two input multiplexers, supporting 28 input signals including external pins, timer channels, ADC, and PLA asynchronous outputs.
- A look-up table (LUT) is implemented in each PLA cell, providing 256 programmable digital logic functions.
- 16 PLA units can be combined to support complex logic operations.

28.2 Functional Description

28.2.1 PLA Block Diagram

The PLA module mainly consists of the following parts:

- Configure PLA through APB interface
- 16 PLA units

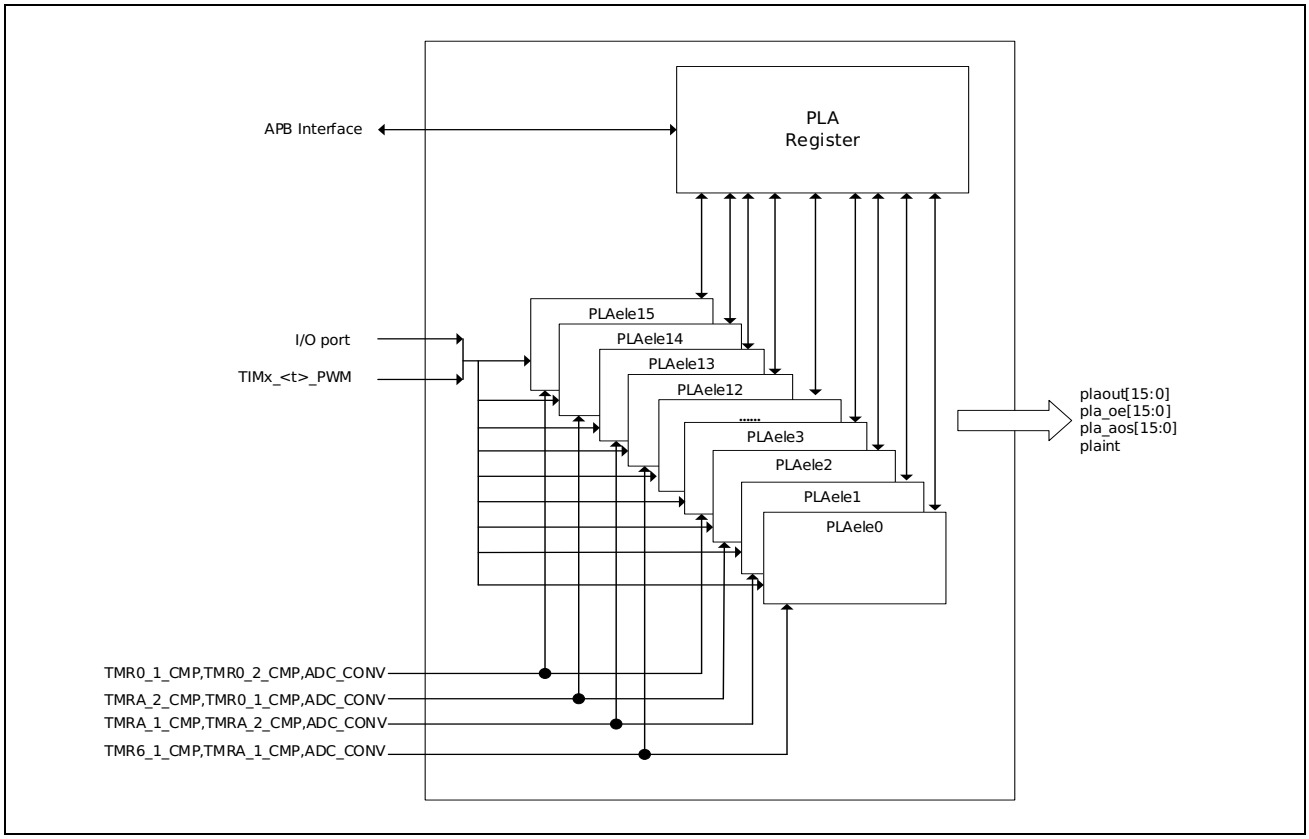


Figure 28-1 PLA Module Block Diagram

28.2.2 Action description

This section provides a description of the PLA module functionality. The module contains sixteen identical PLA units. Figure 13-2.PLunit block diagram shows the block diagram of the PLA unit. Each PLA unit contains two multiplexers and one LUT.

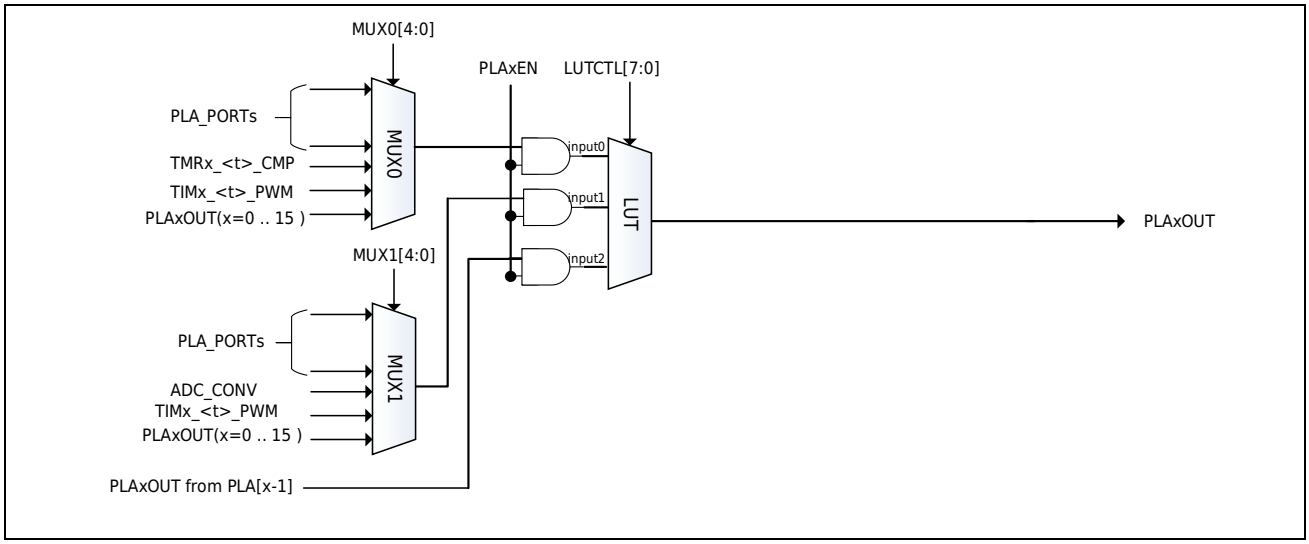


Figure 28-2 PLA Unit Block Diagram

28.2.3 PLA Input Multiplexer

Each PLA unit contains two multiplexers: MUX0 and MUX1. The inputs to each multiplexer can be: external pins (PLA_PORTS), timer compare event trigger signal (TMRx_<t>_CMP), timer channel (TMRx_<t>_PWM), ADC conversion signal (ADC_CONV) and PLA unit asynchronous output (PLAx_OUT). When the output of another PLA unit is used as the input of MUX, a complex combinational logic operation can be implemented.

Table 28-1 and Table 28-2 show the input selections for PLAxMUX0 and PLAxMUX1.

Table 28-1 PLAxMUX0 Input Selection

MUX0[4:0]	PLA0MUX0	PLA1MUX0	PLA2MUX0	PLA3MUX0	PLA4MUX0	PLA5MUX0	PLA6MUX0	PLA7MUX0	PLA8MUX0	PLA9MUX0	PLA10MUX0	PLA11MUX0	PLA12MUX0	PLA13MUX0	PLA14MUX0	PLA15MUX0
00000	PLA0_OUT	PLA0_OUT	PLA0_OUT	PLA0_OUT	PLA0_OUT	PLA0_OUT	PLA0_OUT	PLA0_OUT	PLA0_OUT	PLA0_OUT	PLA0_OUT	PLA0_OUT	PLA0_OUT	PLA0_OUT	PLA0_OUT	PLA0_OUT
00001	PLA1_OUT	PLA1_OUT	PLA1_OUT	PLA1_OUT	PLA1_OUT	PLA1_OUT	PLA1_OUT	PLA1_OUT	PLA1_OUT	PLA1_OUT	PLA1_OUT	PLA1_OUT	PLA1_OUT	PLA1_OUT	PLA1_OUT	PLA1_OUT
00010	PLA2_OUT	PLA2_OUT	PLA2_OUT	PLA2_OUT	PLA2_OUT	PLA2_OUT	PLA2_OUT	PLA2_OUT	PLA2_OUT	PLA2_OUT	PLA2_OUT	PLA2_OUT	PLA2_OUT	PLA2_OUT	PLA2_OUT	PLA2_OUT
00011	PLA3_OUT	PLA3_OUT	PLA3_OUT	PLA3_OUT	PLA3_OUT	PLA3_OUT	PLA3_OUT	PLA3_OUT	PLA3_OUT	PLA3_OUT	PLA3_OUT	PLA3_OUT	PLA3_OUT	PLA3_OUT	PLA3_OUT	PLA3_OUT
00100	TMRA_1_PWM3	TMRA_1_PWM4	TMRA_1_PWM3	TMRA_1_PWM4	TMRA_1_PWM3	TMRA_1_PWM4	TMRA_1_PWM3	TMRA_1_PWM4	TMRA_1_PWM3	TMRA_1_PWM4	TMRA_1_PWM3	TMRA_1_PWM4	TMRA_1_PWM3	TMRA_1_PWM4	TMRA_1_PWM3	TMRA_1_PWM4
00101	TMR6_1_PWMA	TMR6_1_PWMA	TMR6_2_PWMA	TMR6_3_PWMA	TMR6_1_PWMA	TMR6_1_PWMA	TMR6_2_PWMA	TMR6_3_PWMA	TMR6_1_PWMA	TMR6_1_PWMA	TMR6_2_PWMA	TMR6_3_PWMA	TMR6_1_PWMA	TMR6_1_PWMA	TMR6_2_PWMA	TMR6_3_PWMA
00110	TMR6_2_PWMA	TMR6_4_PWMA	TMR6_4_PWMA	TMRA_1_PWM1	TMR6_2_PWMA	TMR6_4_PWMA	TMR6_4_PWMA	TMRA_1_PWM1	TMR6_2_PWMA	TMR6_4_PWMA	TMR6_4_PWMA	TMRA_1_PWM1	TMR6_2_PWMA	TMR6_4_PWMA	TMR6_4_PWMA	TMRA_1_PWM1
00111	TMR6_3_PWMA	TMRA_1_PWM1	TMRA_1_PWM2	TMRA_1_PWM2	TMR6_3_PWMA	TMRA_1_PWM1	TMRA_1_PWM2	TMRA_1_PWM2	TMR6_3_PWMA	TMRA_1_PWM1	TMRA_1_PWM2	TMRA_1_PWM2	TMR6_3_PWMA	TMRA_1_PWM1	TMRA_1_PWM2	TMRA_1_PWM2
01000	PLAIN0	PLAIN4	PLAIN0	PLAIN20	PLAIN0	PLAIN4	PLAIN0	PLAIN20	PLAIN0	PLAIN4	PLAIN0	PLAIN20	PLAIN0	PLAIN4	PLAIN0	PLAIN20
01001	PLAIN2	PLAIN5	PLAIN20	PLAIN2	PLAIN2	PLAIN5	PLAIN20	PLAIN2	PLAIN2	PLAIN5	PLAIN20	PLAIN2	PLAIN2	PLAIN5	PLAIN20	PLAIN2
01010	PLAIN4	PLAIN8	PLAIN8	PLAIN6	PLAIN4	PLAIN8	PLAIN8	PLAIN6	PLAIN4	PLAIN8	PLAIN8	PLAIN6	PLAIN4	PLAIN8	PLAIN8	PLAIN6
01011	PLAIN6	PLAIN10	PLAIN9	PLAIN7	PLAIN6	PLAIN10	PLAIN9	PLAIN7	PLAIN6	PLAIN10	PLAIN9	PLAIN7	PLAIN6	PLAIN10	PLAIN9	PLAIN7
01100	PLAIN8	PLAIN12	PLAIN14	PLAIN10	PLAIN8	PLAIN12	PLAIN14	PLAIN10	PLAIN8	PLAIN12	PLAIN14	PLAIN10	PLAIN8	PLAIN12	PLAIN14	PLAIN10
01101	PLAIN10	PLAIN13	PLAIN15	PLAIN11	PLAIN10	PLAIN13	PLAIN15	PLAIN11	PLAIN10	PLAIN13	PLAIN15	PLAIN11	PLAIN10	PLAIN13	PLAIN15	PLAIN11
01110	PLAIN12	PLAIN16	PLAIN16	PLAIN18	PLAIN12	PLAIN16	PLAIN16	PLAIN18	PLAIN12	PLAIN16	PLAIN16	PLAIN18	PLAIN12	PLAIN16	PLAIN16	PLAIN18
01111	PLAIN14	PLAIN18	PLAIN17	PLAIN19	PLAIN14	PLAIN18	PLAIN17	PLAIN19	PLAIN14	PLAIN18	PLAIN17	PLAIN19	PLAIN14	PLAIN18	PLAIN17	PLAIN19
10000	PLA4_OUT	PLA4_OUT	PLA4_OUT	PLA4_OUT	PLA4_OUT	PLA4_OUT	PLA4_OUT	PLA4_OUT	PLA4_OUT	PLA4_OUT	PLA4_OUT	PLA4_OUT	PLA4_OUT	PLA4_OUT	PLA4_OUT	PLA4_OUT
10001	PLA5_OUT	PLA5_OUT	PLA5_OUT	PLA5_OUT	PLA5_OUT	PLA5_OUT	PLA5_OUT	PLA5_OUT	PLA5_OUT	PLA5_OUT	PLA5_OUT	PLA5_OUT	PLA5_OUT	PLA5_OUT	PLA5_OUT	PLA5_OUT
10010	PLA6_OUT	PLA6_OUT	PLA6_OUT	PLA6_OUT	PLA6_OUT	PLA6_OUT	PLA6_OUT	PLA6_OUT	PLA6_OUT	PLA6_OUT	PLA6_OUT	PLA6_OUT	PLA6_OUT	PLA6_OUT	PLA6_OUT	PLA6_OUT
10011	PLA7_OUT	PLA7_OUT	PLA7_OUT	PLA7_OUT	PLA7_OUT	PLA7_OUT	PLA7_OUT	PLA7_OUT	PLA7_OUT	PLA7_OUT	PLA7_OUT	PLA7_OUT	PLA7_OUT	PLA7_OUT	PLA7_OUT	PLA7_OUT
10100	PLA8_OUT	PLA8_OUT	PLA8_OUT	PLA8_OUT	PLA8_OUT	PLA8_OUT	PLA8_OUT	PLA8_OUT	PLA8_OUT	PLA8_OUT	PLA8_OUT	PLA8_OUT	PLA8_OUT	PLA8_OUT	PLA8_OUT	PLA8_OUT
10101	PLA9_OUT	PLA9_OUT	PLA9_OUT	PLA9_OUT	PLA9_OUT	PLA9_OUT	PLA9_OUT	PLA9_OUT	PLA9_OUT	PLA9_OUT	PLA9_OUT	PLA9_OUT	PLA9_OUT	PLA9_OUT	PLA9_OUT	PLA9_OUT
10110	PLA10_OUT	PLA10_OUT	PLA10_OUT	PLA10_OUT	PLA10_OUT	PLA10_OUT	PLA10_OUT	PLA10_OUT	PLA10_OUT	PLA10_OUT	PLA10_OUT	PLA10_OUT	PLA10_OUT	PLA10_OUT	PLA10_OUT	PLA10_OUT
10111	PLA11_OUT	PLA11_OUT	PLA11_OUT	PLA11_OUT	PLA11_OUT	PLA11_OUT	PLA11_OUT	PLA11_OUT	PLA11_OUT	PLA11_OUT	PLA11_OUT	PLA11_OUT	PLA11_OUT	PLA11_OUT	PLA11_OUT	PLA11_OUT
11000	PLA12_OUT	PLA12_OUT	PLA12_OUT	PLA12_OUT	PLA12_OUT	PLA12_OUT	PLA12_OUT	PLA12_OUT	PLA12_OUT	PLA12_OUT	PLA12_OUT	PLA12_OUT	PLA12_OUT	PLA12_OUT	PLA12_OUT	PLA12_OUT
11001	PLA13_OUT	PLA13_OUT	PLA13_OUT	PLA13_OUT	PLA13_OUT	PLA13_OUT	PLA13_OUT	PLA13_OUT	PLA13_OUT	PLA13_OUT	PLA13_OUT	PLA13_OUT	PLA13_OUT	PLA13_OUT	PLA13_OUT	PLA13_OUT
11010	PLA14_OUT	PLA14_OUT	PLA14_OUT	PLA14_OUT	PLA14_OUT	PLA14_OUT	PLA14_OUT	PLA14_OUT	PLA14_OUT	PLA14_OUT	PLA14_OUT	PLA14_OUT	PLA14_OUT	PLA14_OUT	PLA14_OUT	PLA14_OUT
11011	PLA15_OUT	PLA15_OUT	PLA15_OUT	PLA15_OUT	PLA15_OUT	PLA15_OUT	PLA15_OUT	PLA15_OUT	PLA15_OUT	PLA15_OUT	PLA15_OUT	PLA15_OUT	PLA15_OUT	PLA15_OUT	PLA15_OUT	PLA15_OUT

Table 28-2 PLAxMUX1 Input Selection

MUX1[4:0]	PLA0MUX1	PLA1MUX1	PLA2MUX1	PLA3MUX1	PLA4MUX1	PLA5MUX1	PLA6MUX1	PLA7MUX1	PLA8MUX1	PLA9MUX1	PLA10MUX1	PLA11MUX1	PLA12MUX1	PLA13MUX1	PLA14MUX1	PLA15MUX1
00000	PLA0_OUT	PLA0_OUT	PLA0_OUT	PLA0_OUT	PLA0_OUT	PLA0_OUT	PLA0_OUT	PLA0_OUT	PLA0_OUT	PLA0_OUT	PLA0_OUT	PLA0_OUT	PLA0_OUT	PLA0_OUT	PLA0_OUT	PLA0_OUT
00001	PLA1_OUT	PLA1_OUT	PLA1_OUT	PLA1_OUT	PLA1_OUT	PLA1_OUT	PLA1_OUT	PLA1_OUT	PLA1_OUT	PLA1_OUT	PLA1_OUT	PLA1_OUT	PLA1_OUT	PLA1_OUT	PLA1_OUT	PLA1_OUT
00010	PLA2_OUT	PLA2_OUT	PLA2_OUT	PLA2_OUT	PLA2_OUT	PLA2_OUT	PLA2_OUT	PLA2_OUT	PLA2_OUT	PLA2_OUT	PLA2_OUT	PLA2_OUT	PLA2_OUT	PLA2_OUT	PLA2_OUT	PLA2_OUT
00011	PLA3_OUT	PLA3_OUT	PLA3_OUT	PLA3_OUT	PLA3_OUT	PLA3_OUT	PLA3_OUT	PLA3_OUT	PLA3_OUT	PLA3_OUT	PLA3_OUT	PLA3_OUT	PLA3_OUT	PLA3_OUT	PLA3_OUT	PLA3_OUT
00100	ADC_CONV	ADC_CONV	ADC_CONV	ADC_CONV	ADC_CONV	ADC_CONV	ADC_CONV	ADC_CONV	ADC_CONV	ADC_CONV	ADC_CONV	ADC_CONV	ADC_CONV	ADC_CONV	ADC_CONV	ADC_CONV
00101	TMR6_4_PWMA	TMR6_2_PWMA	TMR6_1_PWMA	TMR6_1_PWMA	TMR6_4_PWMA	TMR6_2_PWMA	TMR6_1_PWMA	TMR6_1_PWMA	TMR6_4_PWMA	TMR6_2_PWMA	TMR6_1_PWMA	TMR6_1_PWMA	TMR6_4_PWMA	TMR6_2_PWMA	TMR6_1_PWMA	TMR6_1_PWMA
00110	TMRA_1_PWM1	TMR6_3_PWMA	TMR6_3_PWMA	TMR6_2_PWMA	TMRA_1_PWM1	TMR6_3_PWMA	TMR6_3_PWMA	TMR6_2_PWMA	TMRA_1_PWM1	TMR6_3_PWMA	TMR6_3_PWMA	TMR6_2_PWMA	TMRA_1_PWM1	TMR6_3_PWMA	TMR6_3_PWMA	TMR6_2_PWMA
00111	TMRA_1_PWM2	TMRA_1_PWM2	TMRA_1_PWM1	TMR6_4_PWMA	TMRA_1_PWM2	TMRA_1_PWM2	TMRA_1_PWM1	TMR6_4_PWMA	TMRA_1_PWM2	TMRA_1_PWM2	TMRA_1_PWM1	TMR6_4_PWMA	TMRA_1_PWM2	TMRA_1_PWM2	TMRA_1_PWM1	TMR6_4_PWMA
01000	PLAIN1	PLAIN6	PLAIN2	PLAIN0	PLAIN1	PLAIN6	PLAIN2	PLAIN0	PLAIN1	PLAIN6	PLAIN2	PLAIN0	PLAIN1	PLAIN6	PLAIN2	PLAIN0
01001	PLAIN3	PLAIN7	PLAIN3	PLAIN1	PLAIN3	PLAIN7	PLAIN3	PLAIN1	PLAIN3	PLAIN7	PLAIN3	PLAIN1	PLAIN3	PLAIN7	PLAIN3	PLAIN1
01010	PLAIN5	PLAIN9	PLAIN10	PLAIN4	PLAIN5	PLAIN9	PLAIN10	PLAIN4	PLAIN5	PLAIN9	PLAIN10	PLAIN4	PLAIN5	PLAIN9	PLAIN10	PLAIN4
01011	PLAIN6	PLAIN11	PLAIN11	PLAIN5	PLAIN6	PLAIN11	PLAIN11	PLAIN5	PLAIN6	PLAIN11	PLAIN11	PLAIN5	PLAIN6	PLAIN11	PLAIN11	PLAIN5
01100	PLAIN9	PLAIN14	PLAIN12	PLAIN8	PLAIN9	PLAIN14	PLAIN12	PLAIN8	PLAIN9	PLAIN14	PLAIN12	PLAIN8	PLAIN9	PLAIN14	PLAIN12	PLAIN8
01101	PLAIN11	PLAIN15	PLAIN13	PLAIN9	PLAIN11	PLAIN15	PLAIN13	PLAIN9	PLAIN11	PLAIN15	PLAIN13	PLAIN9	PLAIN11	PLAIN15	PLAIN13	PLAIN9
01110	PLAIN13	PLAIN17	PLAIN18	PLAIN16	PLAIN13	PLAIN17	PLAIN18	PLAIN16	PLAIN13	PLAIN17	PLAIN18	PLAIN16	PLAIN13	PLAIN17	PLAIN18	PLAIN16
01111	PLAIN15	PLAIN19	PLAIN19	PLAIN17	PLAIN15	PLAIN19	PLAIN19	PLAIN17	PLAIN15	PLAIN19	PLAIN19	PLAIN17	PLAIN15	PLAIN19	PLAIN19	PLAIN17
10000	PLA4_OUT	PLA4_OUT	PLA4_OUT	PLA4_OUT	PLA4_OUT	PLA4_OUT	PLA4_OUT	PLA4_OUT	PLA4_OUT	PLA4_OUT	PLA4_OUT	PLA4_OUT	PLA4_OUT	PLA4_OUT	PLA4_OUT	PLA4_OUT
10001	PLA5_OUT	PLA5_OUT	PLA5_OUT	PLA5_OUT	PLA5_OUT	PLA5_OUT	PLA5_OUT	PLA5_OUT	PLA5_OUT	PLA5_OUT	PLA5_OUT	PLA5_OUT	PLA5_OUT	PLA5_OUT	PLA5_OUT	PLA5_OUT
10010	PLA6_OUT	PLA6_OUT	PLA6_OUT	PLA6_OUT	PLA6_OUT	PLA6_OUT	PLA6_OUT	PLA6_OUT	PLA6_OUT	PLA6_OUT	PLA6_OUT	PLA6_OUT	PLA6_OUT	PLA6_OUT	PLA6_OUT	PLA6_OUT
10011	PLA7_OUT	PLA7_OUT	PLA7_OUT	PLA7_OUT	PLA7_OUT	PLA7_OUT	PLA7_OUT	PLA7_OUT	PLA7_OUT	PLA7_OUT	PLA7_OUT	PLA7_OUT	PLA7_OUT	PLA7_OUT	PLA7_OUT	PLA7_OUT
10100	PLA8_OUT	PLA8_OUT	PLA8_OUT	PLA8_OUT	PLA8_OUT	PLA8_OUT	PLA8_OUT	PLA8_OUT	PLA8_OUT	PLA8_OUT	PLA8_OUT	PLA8_OUT	PLA8_OUT	PLA8_OUT	PLA8_OUT	PLA8_OUT
10101	PLA9_OUT	PLA9_OUT	PLA9_OUT	PLA9_OUT	PLA9_OUT	PLA9_OUT	PLA9_OUT	PLA9_OUT	PLA9_OUT	PLA9_OUT	PLA9_OUT	PLA9_OUT	PLA9_OUT	PLA9_OUT	PLA9_OUT	PLA9_OUT
10110	PLA10_OUT	PLA10_OUT	PLA10_OUT	PLA10_OUT	PLA10_OUT	PLA10_OUT	PLA10_OUT	PLA10_OUT	PLA10_OUT	PLA10_OUT	PLA10_OUT	PLA10_OUT	PLA10_OUT	PLA10_OUT	PLA10_OUT	PLA10_OUT
10111	PLA11_OUT	PLA11_OUT	PLA11_OUT	PLA11_OUT	PLA11_OUT	PLA11_OUT	PLA11_OUT	PLA11_OUT	PLA11_OUT	PLA11_OUT	PLA11_OUT	PLA11_OUT	PLA11_OUT	PLA11_OUT	PLA11_OUT	PLA11_OUT
11000	PLA12_OUT	PLA12_OUT	PLA12_OUT	PLA12_OUT	PLA12_OUT	PLA12_OUT	PLA12_OUT	PLA12_OUT	PLA12_OUT	PLA12_OUT	PLA12_OUT	PLA12_OUT	PLA12_OUT	PLA12_OUT	PLA12_OUT	PLA12_OUT
11001	PLA13_OUT	PLA13_OUT	PLA13_OUT	PLA13_OUT	PLA13_OUT	PLA13_OUT	PLA13_OUT	PLA13_OUT	PLA13_OUT	PLA13_OUT	PLA13_OUT	PLA13_OUT	PLA13_OUT	PLA13_OUT	PLA13_OUT	PLA13_OUT
11010	PLA14_OUT	PLA14_OUT	PLA14_OUT	PLA14_OUT	PLA14_OUT	PLA14_OUT	PLA14_OUT	PLA14_OUT	PLA14_OUT	PLA14_OUT	PLA14_OUT	PLA14_OUT	PLA14_OUT	PLA14_OUT	PLA14_OUT	PLA14_OUT
11011	PLA15_OUT	PLA15_OUT	PLA15_OUT	PLA15_OUT	PLA15_OUT	PLA15_OUT	PLA15_OUT	PLA15_OUT	PLA15_OUT	PLA15_OUT	PLA15_OUT	PLA15_OUT	PLA15_OUT	PLA15_OUT	PLA15_OUT	PLA15_OUT

28.2.4 Look-up Table (LUT) Control

Each PLA unit has a LUT, which introduces 256 logical combination functions controlled by the LUTCTL[7:0] bits of the PLAx_LUTCTL register. The output of the lookup table is controlled by the three bits input2, input1, and input0, as shown in Table 28-3.

Table 28-3 LUT Control

input2	input1	input0	Output
0	0	0	LUTCTL[0]
0	0	1	LUTCTL[1]
0	1	0	LUTCTL[2]
0	1	1	LUTCTL[3]
1	0	0	LUTCTL[4]
1	0	1	LUTCTL[5]
1	1	0	LUTCTL[6]
1	1	1	LUTCTL[7]

For example, to implement the logic of {input2^input1^input0}, you need to set LUTCTL[7:0] = 8'b10010110.

Application example: To implement the logic of {input2^input1^input0} of the pla_ele0 unit, first configure the PLA and GPIO:

Set PSPCR to 0x0000 to disable special functions such as JTAG.

Set PFSR (0,15) to 0x9, PLAIN0 to PA15,

Set PFSR (1,3) to 0x9, and PLAIN1 to PB3,

Set PFSR (0,2) to 0xa, and PLA0OUT to PA2,

PLA_GCTL=0x00000001, enables PLA0 unit.

PLA0_LUTCTL=0x00000096, implements the logic of {input2^input1^input0},

PLA0_MUXS = 0x00000088, select MUX0 01000 and MUX1 01000,

PLA0_CTL=0x000000C0, select LUT result for direct output, and allow PLA_OUT0 output.

28.2.5 PLA Output

Each PLA unit has a GPIO port as its output which is the result of the LUT.

Table 28-4 PLA Output Port Correspondence Table

Pin Name	Input/Output	Function
PA2/PA8/PB3	O	PLA0OUT
PA7/PA9/PB0/PB4	O	PLA1OUT
PA10/PB5/PC10	O	PLA2OUT
PC13/PA11/PA15/PB8	O	PLA3OUT
PC0/PA3	O	PLA4OUT
PC1/PA4	O	PLA5OUT
PA5/PC2	O	PLA6OUT
PA6/PA12/PC3	O	PLA7OUT
PA13/PB12	O	PLA8OUT
PB13	O	PLA9OUT
PB14	O	PLA10OUT
PB15	O	PLA11OUT
PC4/PC6	O	PLA12OUT
PC5/PC7	O	PLA13OUT
PC8/PC11	O	PLA14OUT
PC9/PC12	O	PLA15OUT

28.3Register Description

Table 28-5 PLA Base Address

Name	Base Address	Description
PLA	0x4004 FC00	PLA Base Address

Table 28-6 Register List

Register	Description	Offset Address	Bit Width	Reset Value
PLA_GCTL	Global Control Registers	0x00	32	0x0000 0000
PLA_STAT	Status register	0x0C	32	0x0000 0000
PLAx_MUXS	Multiplexer Select Register	0x10 + 0x0C * x	32	0x0000 0000
PLAx_LUTCTL	LUT control register	0x14 + 0x0C * x	32	0x0000 0000
PLAx_CTL	Control register	0x18 + 0x0C * x	32	0x0000 0000

Note*1: In the address calculation formula, x = 0~15

28.3.1 Global Control Register (PLA_GCTL)

Reset value: 0x0000 0000

This register can only be accessed as a word (32 bits)

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
PLA15EN	PLA14EN	PLA13EN	PLA12EN	PLA11EN	PLA10EN	PLA9EN	PLA8EN	PLA7EN	PLA6EN	PLA5EN	PLA4EN	PLA3EN	PLA2EN	PLA1EN	PLA0EN

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b16	Res	Reserved Bit	Read at 0, write at 0	RW
b15	PLA15EN	PLA15 unit enable	0: Disable PLA15 unit 1: Allow PLA15 unit	RW
b14	PLA14EN	PLA14 unit enable	0: Disable PLA14 unit 1: Allow PLA14 unit	RW
b13	PLA13EN	PLA13 unit enable	0: Disable PLA13 unit 1: Allow PLA13 unit	RW
b12	PLA12EN	PLA12 unit enable	0: Disable PLA12 unit 1: Allow PLA12 unit	RW
b11	PLA11EN	PLA11 unit enable	0: Disable PLA11 unit 1: Allow PLA11 unit	RW
b10	PLA10EN	PLA10 unit enable	0: Disable PLA10 unit 1: Allow PLA10 unit	RW
b9	PLA9EN	PLA9 unit enable	0: Disable PLA9 unit 1: Allow PLA9 unit	RW
b8	PLA8EN	PLA8 unit enable	0: Disable PLA8 unit 1: Allow PLA8 unit	RW
b7	PLA7EN	PLA7 unit enable	0: Disable PLA7 unit 1: Allow PLA7 unit	RW
b6	PLA6EN	PLA6 unit enable	0: Disable PLA6 unit 1: Allow PLA6 unit	RW
b5	PLA5EN	PLA5 unit enable	0: Disable PLA5 unit 1: Allow PLA5 unit	RW
b4	PLA4EN	PLA4 unit enable	0: Disable PLA4 unit 1: Allow PLA4 unit	RW
b3	PLA3EN	PLA3 unit enable	0: Disable PLA3 unit 1: Allow PLA3 unit	RW
b2	PLA2EN	PLA2 unit enable	0: Disable PLA2 unit 1: Allow PLA2 unit	RW
b1	PLA1EN	PLA1 unit enable	0: Disable PLA1 unit 1: Allow PLA1 unit	RW
b0	PLA0EN	PLA0 unit enable	0: Disable PLA0 unit 1: Allow PLA0 unit	RW

28.3.2 Status Register (PLA_STAT)

Reset value: 0x0000 0000

This register can only be accessed as a word (32 bits)

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
PLA15OUT	PLA14OUT	PLA13OUT	PLA12OUT	PLA11OUT	PLA10OUT	PLA9OUT	PLA8OUT	PLA7OUT	PLA6OUT	PLA5OUT	PLA4OUT	PLA3OUT	PLA2OUT	PLA1OUT	PLA0OUT

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b16	Reserved	-	Read as 0	R
b15	PLA15OUT	PLA15 unit output status	0: The current logic level of the PLA15 unit is low 1: The current logic level of the PLA15 unit is high	R
b14	PLA14OUT	PLA14 unit output status	0: The current logic level of the PLA14 unit is low 1: The current logic level of the PLA14 unit is high	R
b13	PLA13OUT	PLA13 unit output status	0: The current logic level of the PLA13 unit is low 1: The current logic level of the PLA13 unit is high	R
b12	PLA12OUT	PLA12 unit output status	0: The current logic level of the PLA12 unit is low 1: The current logic level of the PLA12 unit is high	R
b11	PLA11OUT	PLA11 unit output status	0: The current logic level of the PLA11 unit is low 1: The current logic level of the PLA11 unit is high	R
b10	PLA10OUT	PLA10 unit output status	0: The current logic level of the PLA10 unit is low 1: The current logic level of the PLA10 unit is high	R
b9	PLA9OUT	PLA9 unit output status	0: The current logic level of the PLA9 unit is low 1: The current logic level of the PLA9 unit is high	R
b8	PLA8OUT	PLA8 unit output status	0: The current logic level of the PLA8 unit is low 1: The current logic level of the PLA8 unit is high	R
b7	PLA7OUT	PLA7 unit output status	0: The current logic level of the PLA7 unit is low 1: The current logic level of the PLA7 unit is high	R
b6	PLA6OUT	PLA6 unit output status	0: The current logic level of the PLA6 unit is low 1: The current logic level of the PLA6 unit is high	R
b5	PLA5OUT	PLA5 unit output status	0: The current logic level of the PLA5 unit is low 1: The current logic level of the PLA5 unit is high	R
b4	PLA4OUT	PLA4 unit output status	0: The current logic level of the PLA4 unit is low 1: The current logic level of the PLA4 unit is high	R
b3	PLA3OUT	PLA3 unit output status	0: The current logic level of the PLA3 unit is low 1: The current logic level of the PLA3 unit is high	R
b2	PLA2OUT	PLA2 unit output status	0: The current logic level of the PLA2 unit is low 1: The current logic level of the PLA2 unit is high	R
b1	PLA1OUT	PLA1 unit output status	0: The current logic level of the PLA1 unit is low 1: The current logic level of the PLA1 unit is high	R
b0	PLA0OUT	PLA0 unit output status	0: The current logic level of the PLA0 unit is low 1: The current logic level of the PLA0 unit is high	R

28.3.3 Multiplexer Selection Register (PLAx_MUXS, x=0~15)

Reset value: 0x0000 0000

This register can only be accessed as a word (32 bits)

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
MUX0[4]	MUX1[4]	Res						MUX0[3:0]				MUX1[3:0]			

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b16	Res	Reserved Bit	Read as 0	RW
b15	MUX0[4]	Multiplexer 0 input selects bit 4	This bit selects the input of MUX0 Refer to Table 28-1 PLAxMUX0 Input Selection	RW
b14	MUX1[4]	Multiplexer 1 input selects bit 4	This bit selects the input of MUX1 Refer to Table 28-2 PLAxMUX1 Input Selection	RW
b13~b8	Res	Reserved Bit	Read as 0, write as 0	RW
b7~b4	MUX0[3:0]	Multiplexer 0 input selects bits 3~0	These bits select the input of MUX0 Refer to Table 28-1 PLAxMUX0 Input Selection	RW
b3~b0	MUX1[3:0]	Multiplexer 1 input selection bits 3 to 0	These bits select the input of MUX1 Refer to Table 28-2 PLAxMUX1 Input Selection	RW

28.3.4 Lookup Table Control Register (PLAx_LUTCTL, x = 0 ~ 15)

Reset value: 0x0000 0000

This register can only be accessed as a word (32 bits)

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res								LUTCTL[7:0]							

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b8	Res	Reserved Bit	Read as 0, write as 0	RW
b7~b0	LUTCTL[7:0]	LUT control bit	These bits control which logic function of input2, input1, and input0 will affect the output. For example: IN1|IN0: LUTCTL=8`b11101110 IN2& (IN1^IN0):LUTCTL=8`b01100000	RW

28.3.5 Control Register (PLAx_CTL, x = 0 ~ 15)

Reset value: 0x0000 0000

This register can only be accessed as a word (32 bits)

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res								OSEL	OEN	Res					

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b8	Res	Reserved Bit	Read as 0, write as 0	RW
b7	OSEL	Output selection	0: The result of LUT is not output as PLAx 1: LUT result as output of PLAx	RW
b6	OEN	Output enable	0: PLAx output disable 1: PLAx output enable	RW
b5~b0	Res	Reserved Bit	Read as 0, write as 0	RW

28.4Precautions

Users are prohibited from accessing the space from 0x4004 FD00 to 0x4004 FFFF.

When configuring the PLA register, be sure to read and write 32 bits to ensure that the register setting is consistent with the user's requirements.

When the user changes the LUTCTL value, careful calculation is required and the input selection of PLAx_MUXS is coordinated to avoid a combined loop. When modifying the PLA register, the PLA_GCTL register needs to be cleared to prevent a combined loop.

The PLA module only supports the synchronous configuration of a single unit. The timing relationship between multiple units needs to be set by the user according to the actual situation.

29 Serial Peripheral Interface (SPI)

29.1 Introduction

This product is equipped with 1-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 needed.

29.2 SPI Main Characteristics

Table 29-1 Characteristics of SPI

Essentials	Description
Serial communication function	• Supporting 4-wire SPI mode and 3-wire clock synchronization mode • Supports full-duplex and send-only communication modes • Polarity and phase of adjustable communication clock SCK
Data format	• Selectable data shift order: MSB start/LSB start • Selectable data width:4/5/6/7/8/9/10/11/12/13/14/15/16/20/24/32 bit • A maximum of 32 bits of data can be transmitted or received ifn four single frame
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 PCLK1. • Maximum Baud Rate Allowed in Slave Mode is 6 Frequency Division of PCLK1
data buffer	• Data buffer area with 16 bytes • Supports double buffering
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 excluding chip-selected signal line • The relative timing relationship between the chip select signal and the communication clock can be adjusted • The invalid time of the chip select signal between two consecutive communications can be adjusted • Polarity adjustable
Transmission Control in master Mode	• Start transmission by writing data to the data register • Communication auto suspend function
Interrupt	• Received data area full • Sending data area is empty • SPI error (mode/overload/underload/parity) • SPI Vacant • Transfer completed (event sources only)
Low power control	• Configurable module stop
Other functions	• SPI initialization function

Note:

– *When the communication auto-suspend feature is used in master receive mode, overload errors will not occur because the communication clock stops.* For details, please refer to [Overload Error].

29.3SPI Function Description

29.3.1 SPI System Block Diagram

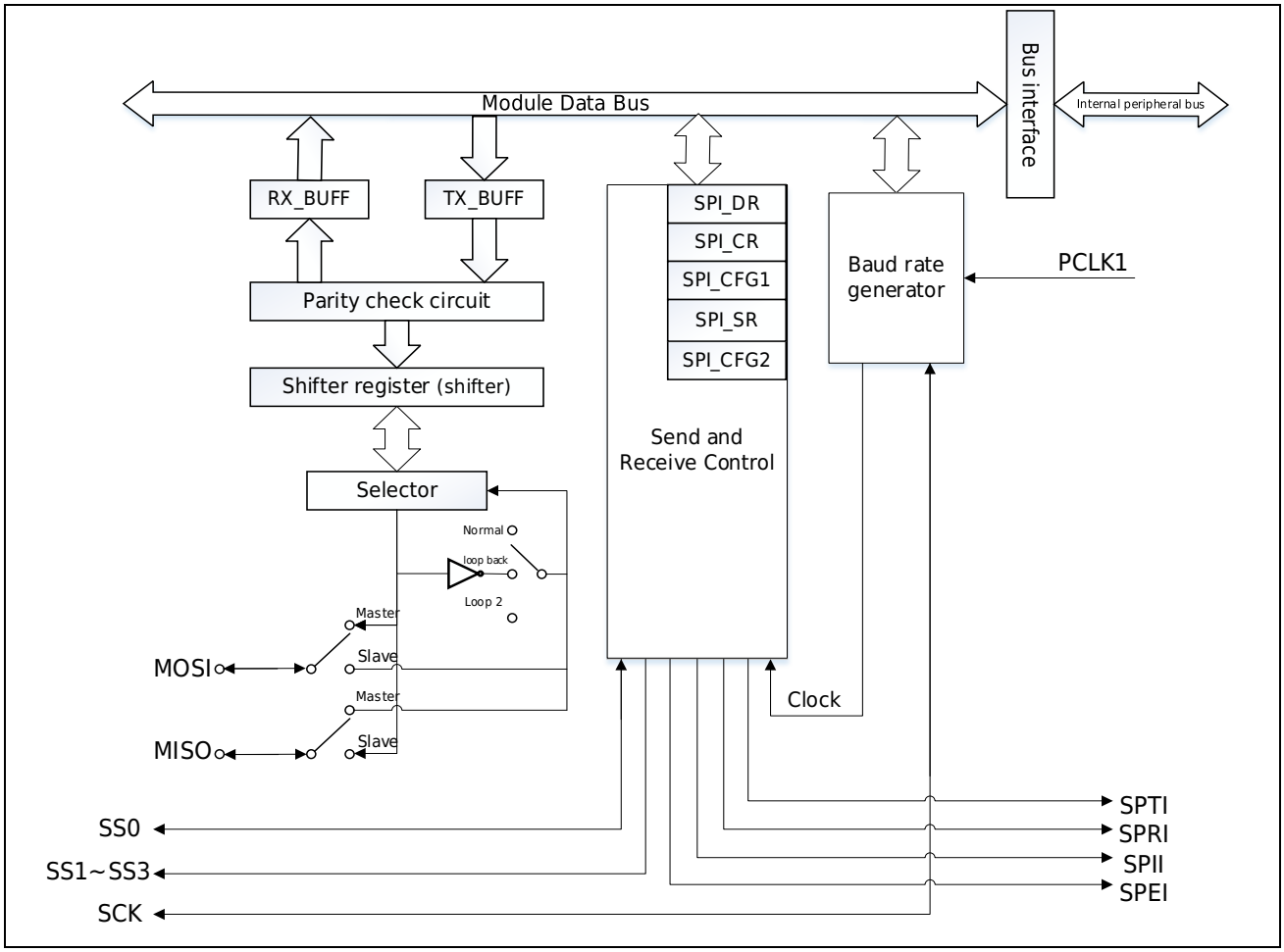


Figure 29-1 System Block Diagram

Table 29-2 Pin Description

Pin Name	Port Direction	Function
SCK	Input/Output	Communication clock pin
MOSI	Input/Output	Master data transfer pin
MISO	Input/Output	Slave data transfer pin
SS0	Input/Output	Select input/output pin from machine
SS1	Output	Select output pin from machine
SS2	Output	Select output pin from machine
SS3	Output	Select output pin from machine

29.3.2 SPI Action System Description

29.3.2.1 Pin Status in Master Mode

When the SPI works in master mode, the status of each pin is shown in the Table 29-3 below.

Table 29-3 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 universal I/O)
		MOSI	Output
		MISO	Input

Note:

Please set the pin input type to CMOS input and the output to high drive mode. Please refer to the GPIO register [Universal Control Register (PCRxy)] for setting.

29.3.2.2 Pin State In Slave Mode

When the SPI works in slave mode, the status of each pin is shown in the Table 29-4 below.

Table 29-4 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 universal I/O)
		MOSI	Input
		MISO	Output/Hi-Z
Clock synchronization operation (SPIMDS=1)	Slave mode (MSTR=0)	SCK	Input
		SS0~SS3 (not used)	Hi-Z (available as universal I/O)
		MOSI	Input
		MISO	Output

Note:

Please set the pin input type to CMOS input and the output to high drive mode. Please refer to the GPIO register [Universal Control Register (PCRxy)] for setting.

29.3.2.3 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 valid level.

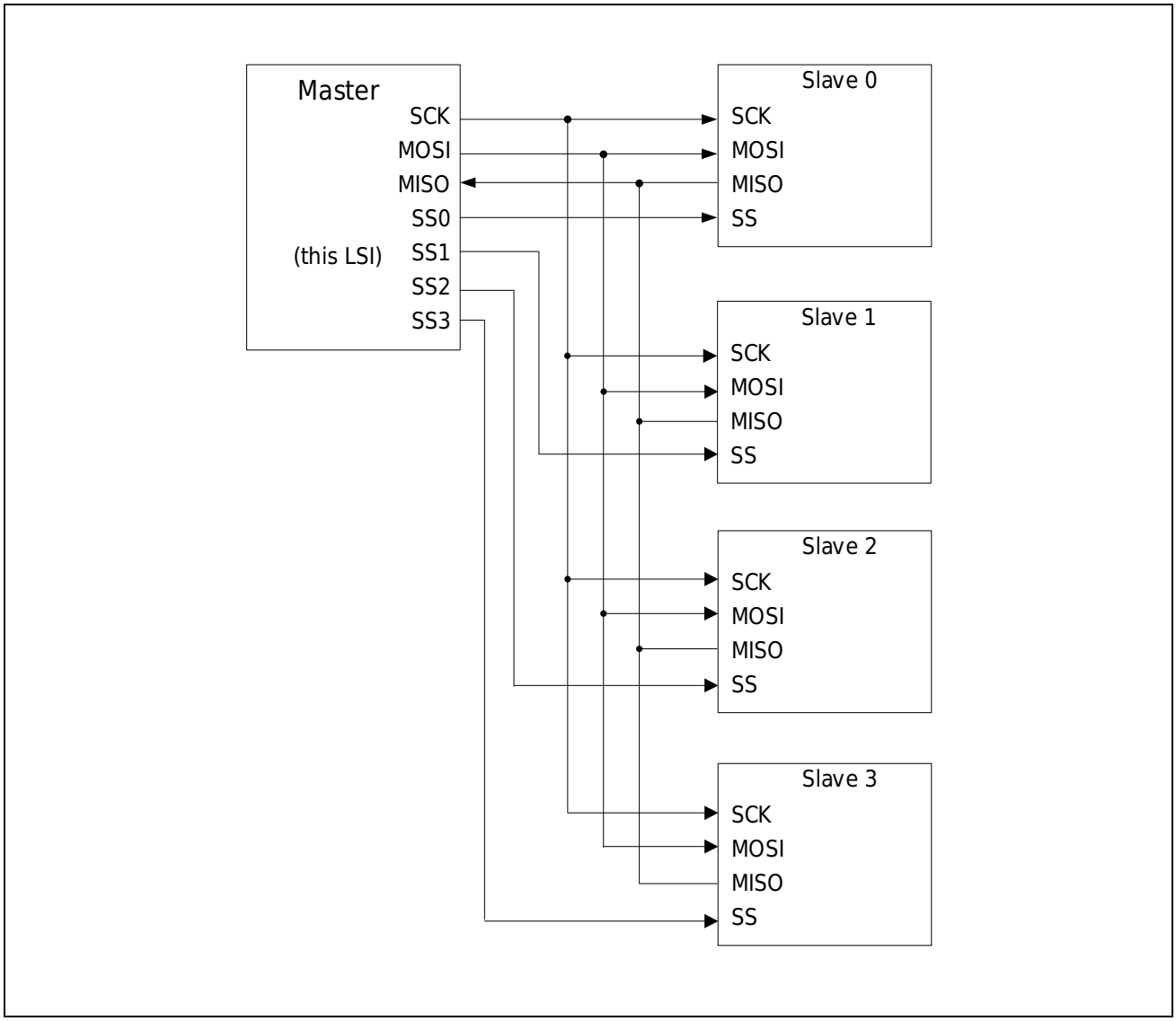


Figure 29-2 Master Mode Structure

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.

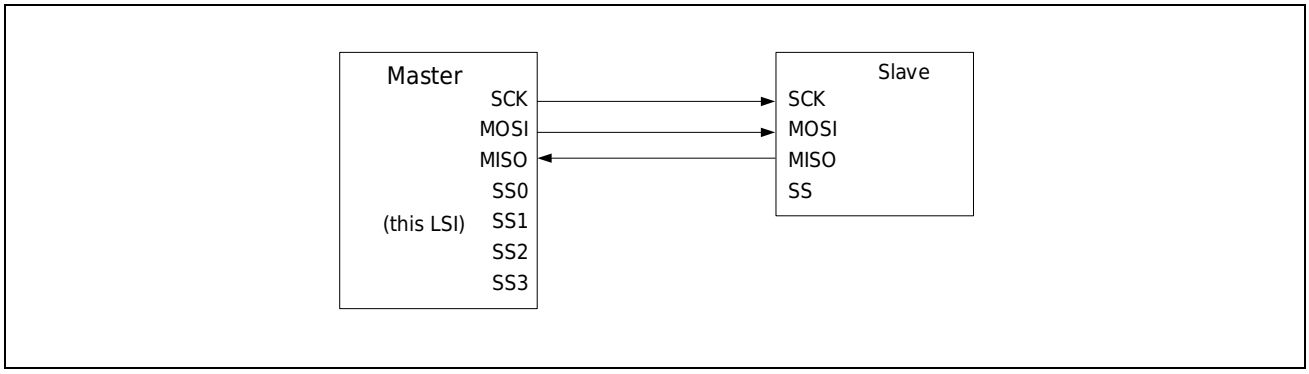


Figure 29-3 Three-wire clock synchronization operation

29.3.3 Data Communication Description

29.3.3.1 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 [1: 0] bit setting. The calculation method is shown in the following formula, where N is the setting value of the MBR[1:0] bit, ranging from 0 to 3, and M is the setting value of the CLKDIV[3:0] bit, ranging from 0 to 15.

Table 29-5 Some Setting Values Bit Speed

MBR[1:0] Bit Setting Value (N)	CLKDIV[3:0] Bit Setting Value (M)	Frequency division ratio	Baud rate			
			PCLK1=5 MHz	PCLK1=10 MHz	PCLK1=20 MHz	PCLK1=40 MHz
0	0	2	2.50Mbps	5.00Mbps	10.0Mbps	20.0Mbps
0	1	4	1.25Mbps	2.50Mbps	5.00Mbps	10.0Mbps
0	3	8	625Kbps	1.25Mbps	2.50Mbps	5.00Mbps
0	7	16	313Kbps	625Kbps	1.25Mbps	2.50Mbps
0	15	32	156Kbps	313Kbps	625Kbps	1.25Mbps
1	15	64	78Kbps	156Kbps	313Kbps	625Kbps
2	15	128	39Kbps	78Kbps	156Kbps	313Kbps
3	15	256	20Kbps	39Kbps	78Kbps	156Kbps

29.3.3.2 Data Format

The data format of SPI depends on the setting value of the parity check enable bit PAE in the SPI command register SPI_CFG2 and the SPI control register SPI_CR. 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 DSIZE[3:0] bits in the register SPI_CFG2) as the transfer object, regardless of the MSB/LSB shift order.

SPI_CFG2.DSIZE[3:0] determines the bit width of the data, which ranges from 4 to 32 bits. SPI_CR.PAE determines the last bit of the data. When PAE is 1, the last bit is used as the parity check bit. When it is 0, it is the lowest bit of the data itself. As shown in Figure 29-4.

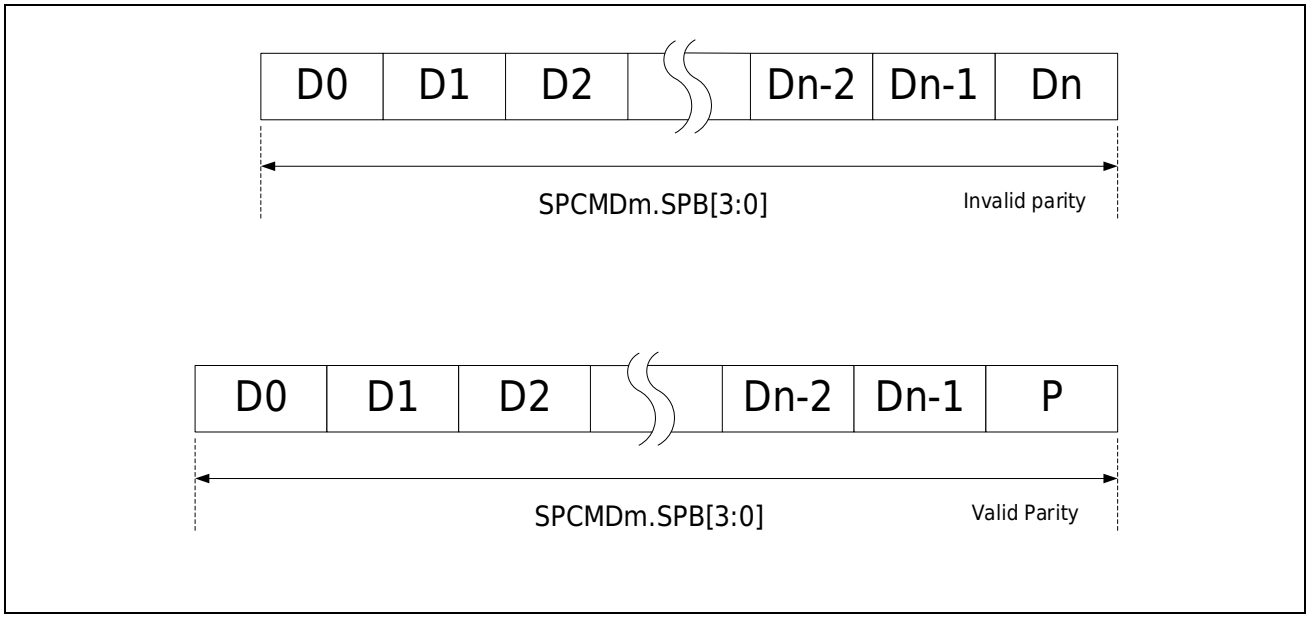


Figure 29-4 Data Format

When SPI data is sent, the transmitted data first enters the transmitting buffer (TX _ BUFF), and then copies the TX _ BUFF data to the shifter register (shifter). When SPI data is received, the data is moved in from shifter in turn, and then the shifter data is copied to the receiving 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 check control bit SPI_CR.PAE:

1) MSB first pass, parity invalid

Data d31 ~ d0 are copied from TX _ BUFF to shifter in order, and moved out of shifter in order of d31 ~ d0.

Upon receiving, data d31 ~ d0 is moved in from the lowest bit of shifter and copied to RX _ BUFF after all data is moved in.

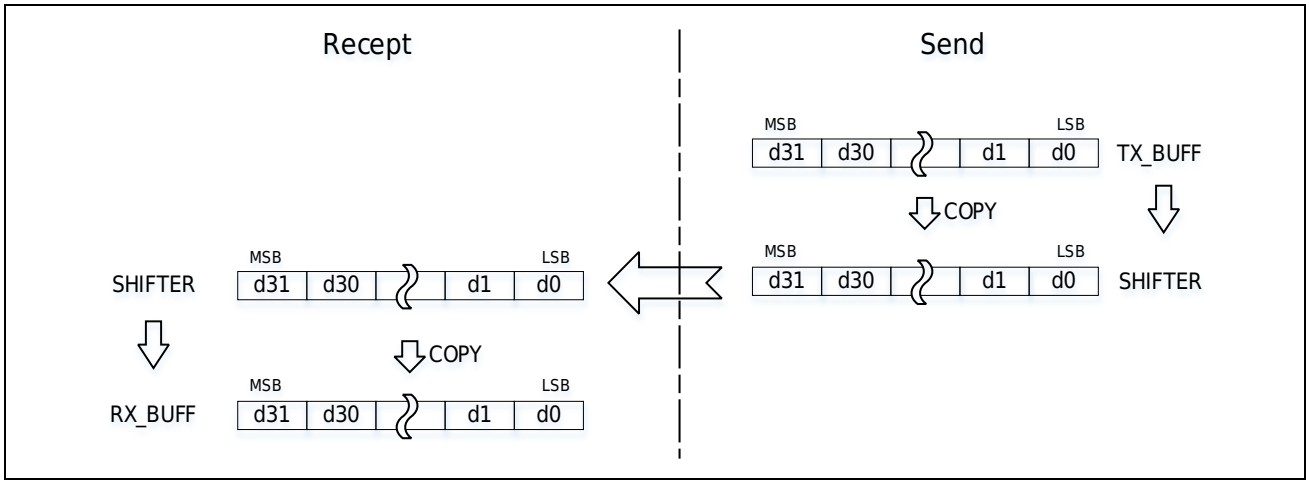


Figure 29-5 MSB sent First, Parity Check Invalid

2) LSB first pass, parity invalid

During transmission, d31 ~ d0 is copied from TXBUFF to shifter in the order of d0 ~ d31, and moved out of shifter in the order of d0 ~ d31.

Upon receiving, data d0 ~ d31 is moved in from the lowest bit of shifter, and then copied from shifter to RX _ BUFF in the order d31 ~ d0 after all data is moved in.

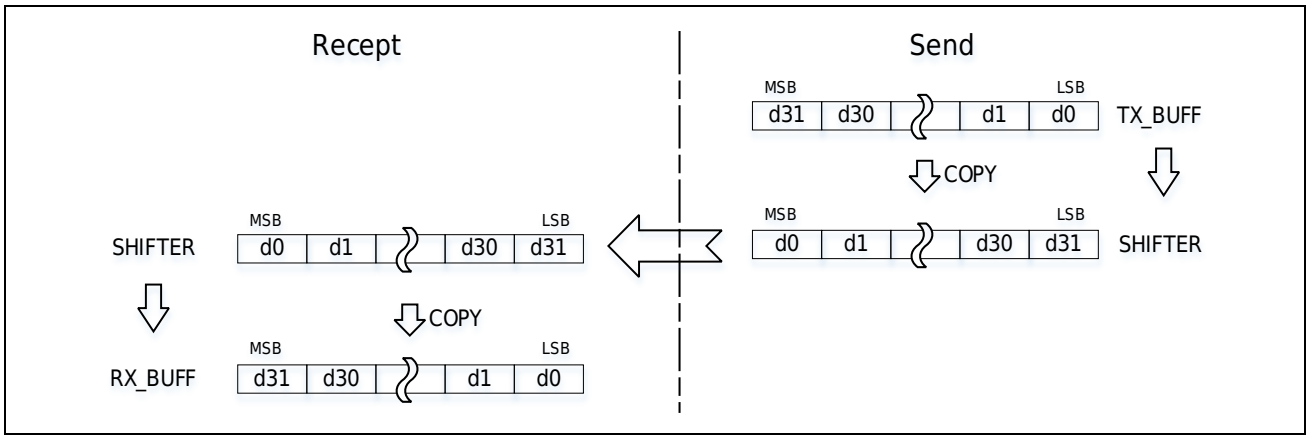


Figure 29-6 LSB sent First, Parity Check Invalid

3) MSB transmitted first, parity is valid

When transmitting, the value of parity bit P is calculated according to the value of d31 ~ d1, then P is replaced by d0, copied to shifter according to the order of d31 ~ d1, P, and moved out of shifter according to the order of d31 ~ P.

Upon receiving, the data d31 ~ P moves in from the lowest bit of shifter, and parity is performed when the data is copied to shifter. Finally, copy the data to RX _ BUFF.

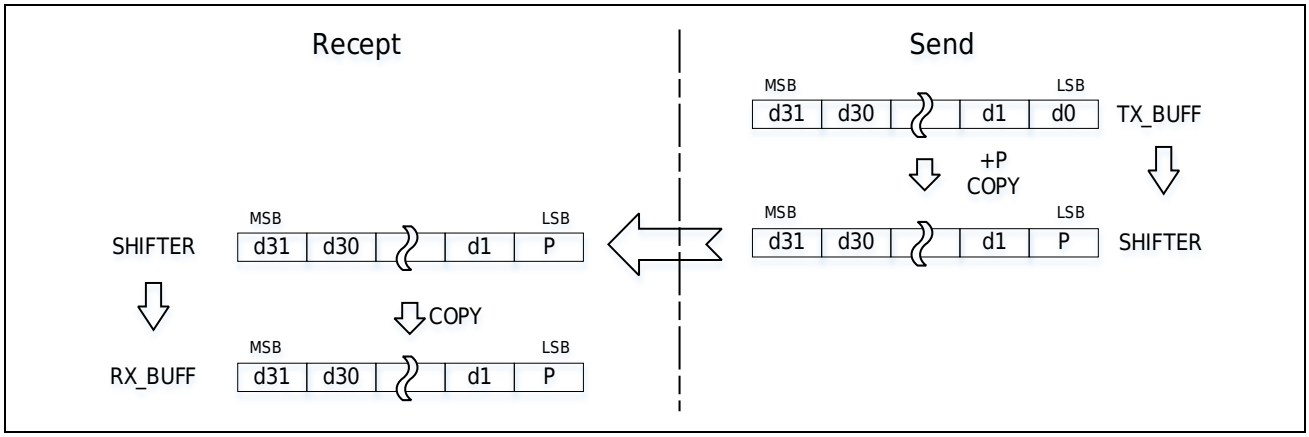


Figure 29-7 MSB Sent First, When Parity Check Valid

4) LSB first pass, parity is valid

When transmitting, the value of parity bit P is calculated according to the value of d30 ~ d0, and then P is used instead of d31, which is copied from TXBUFF to shifter in the order of d0 ~ P and moved out of shifter in the order of d0 ~ P.

Upon 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 order of Pd0.

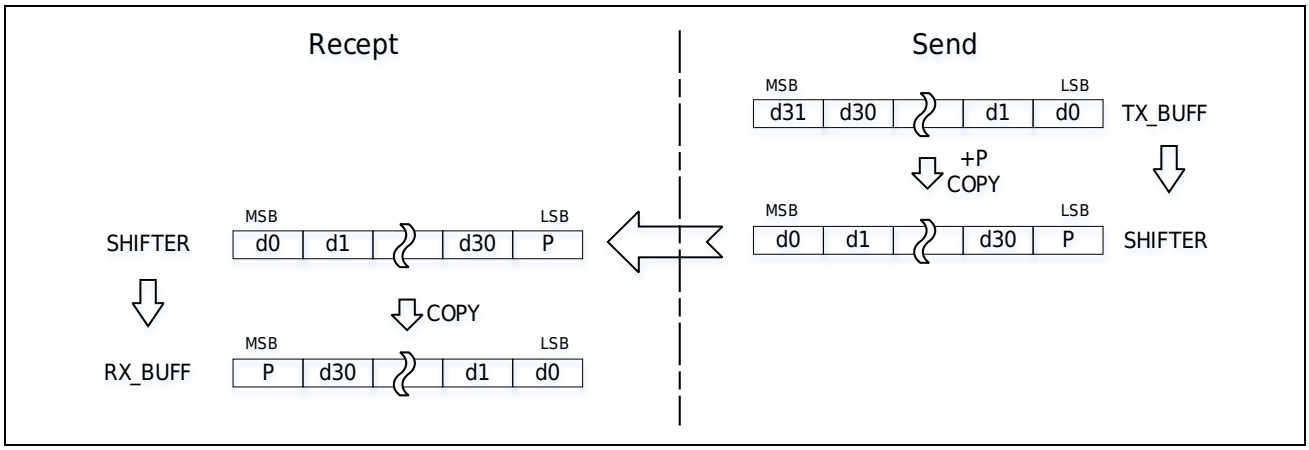


Figure 29-8 LSB Sent First, When Parity Check Valid

29.3.3.3 Transfer format

1) CPHA = 0

When SPI _ CFG2.When the CPHA bit is 0, SPI samples the data at the odd edge of SCK and updates the data at the even edge. Figure 29-9 is the timing diagram of SPI transmission when CPHA=0. When the input level of the SSi signal becomes active, MOSI/MISO begins to update the transmitted data. After the SSi signal becomes valid for the first time on the edge of the first SCK signal, the data is sampled once per SCK cycle. MOSI/MISO signals are updated at 1/2 SCK cycles 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.

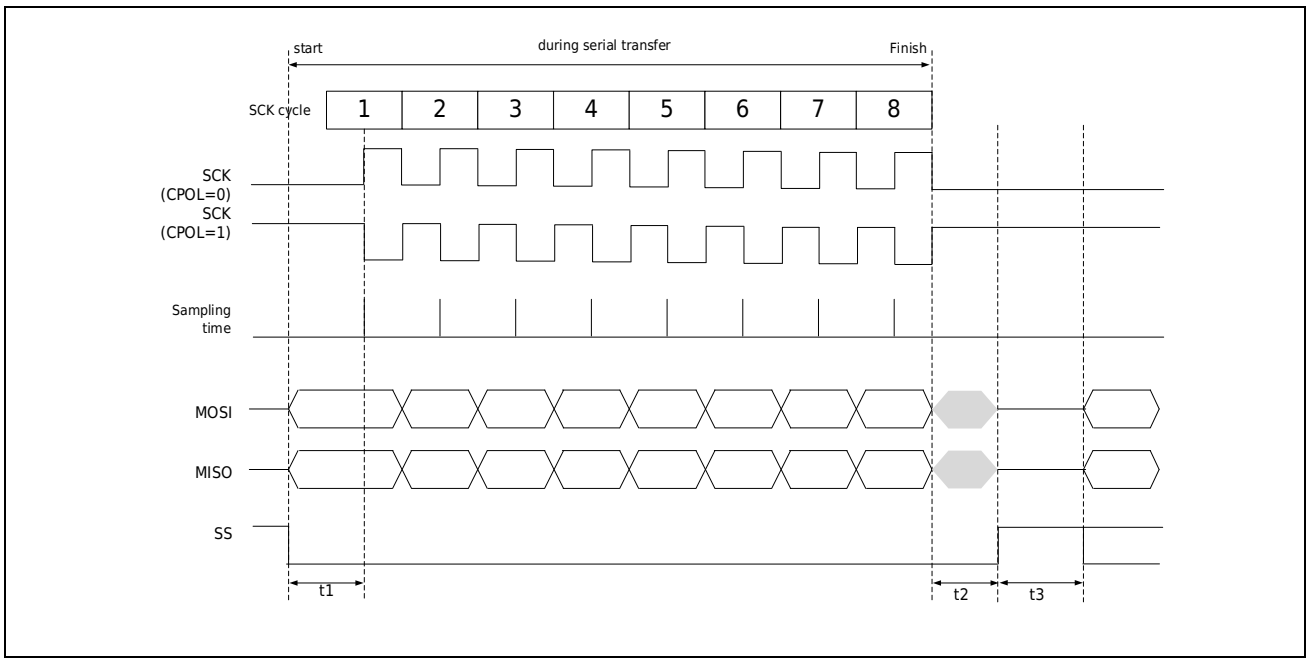


Figure 29-9 Data Transfer Format Diagram (CPHA = 0)

In the above figure, t1 represents the interval from the SS signal being valid to the SCK oscillation (SCK delay time, which can be controlled by setting SPI_CFG1.MSSI[2:0] and SPI_CFG2.MSSIE), t2 represents the interval from the SCK oscillation stopping to the SS signal becoming invalid (SS invalid delay time, which can be controlled by setting SPI_CFG1.MSSDL[2:0] and SPI_CFG2.MSSDLE), and t3 represents the minimum waiting time from the end of serial transmission to the start of the next transmission (the next access delay, which can be controlled by setting SPI_CFG1.MIDI[2:0] and SPI_CFG2.MIDIE). t1, t2 and t3 are controlled by the Master device on the SPI system. For details, please refer to the chapter [Master Action in SPI Operation Mode].

2) CPHA = 1

When the SPI_CFG2.CPHA bit is 1, the SPI updates data on the odd edges of SCK and samples data on the even edges. Figure 29-10 It is the timing diagram of SPI transmission when CPHA=1. MOSI/MISO starts to update the data transmission at the first SCK signal edge after the SSi signal becomes a valid level. After this, data is updated every SCK cycle. Data is sampled every 1/2 SCK period 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.

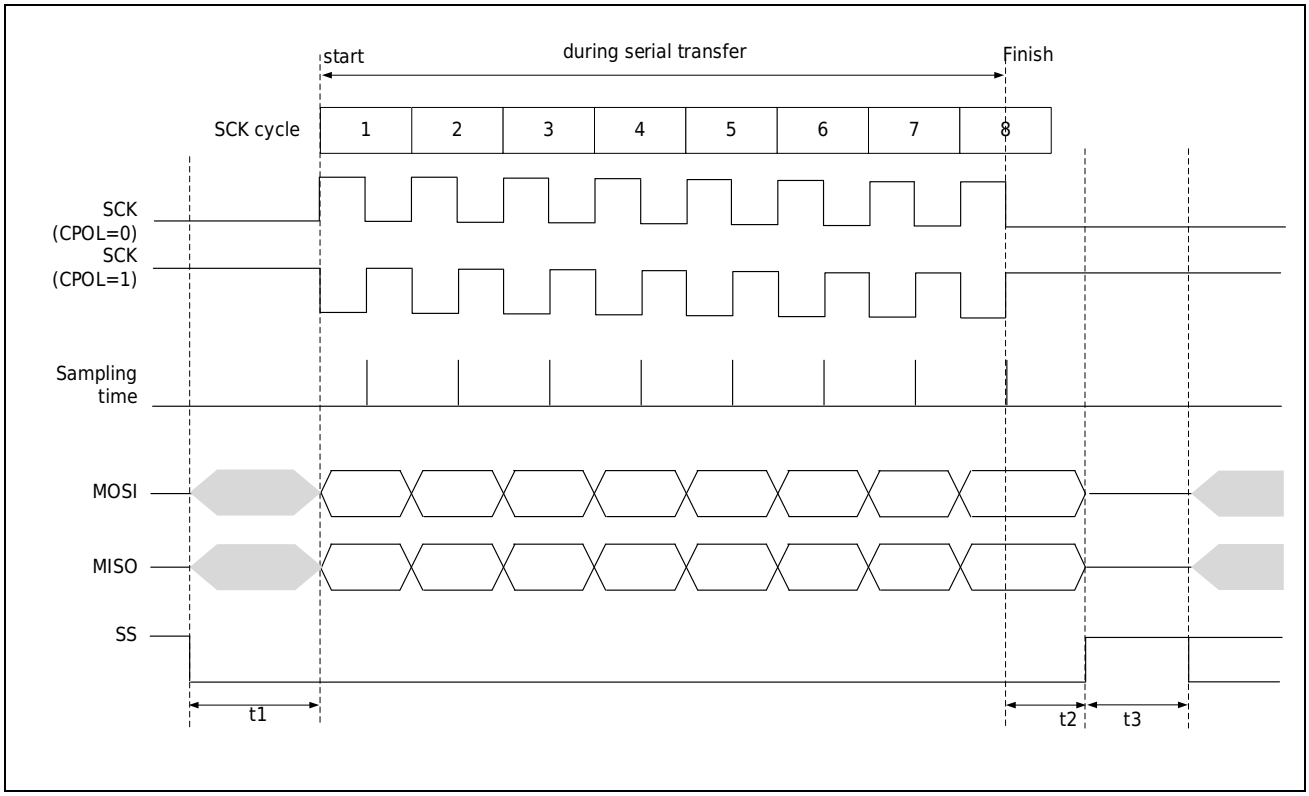


Figure 29-10 Data Transfer Format (CPHA = 1)

The same is true for t1, t2, t3 and CPHA bits.

29.3.3.4 Communication Mode

This SPI has two communication modes: full-duplex synchronous serial communication and send-only serial communication, which can be selected through the TXMDS bit of the SPI control register (SPI_CR).

1) Full-duplex synchronous serial communication mode

When the SPI_CR.TXMDS bit is 0, the SPI operates in full-duplex synchronous serial communication mode. As shown in Figure 29-11, the SPI_CFG1.FTHLV[1:0] bits are "0b00", the SPI_CFG2.CPHA bits are 1 and the SPI_CFG2.CPOL bits are 0, and the SPI performs 8-bit serial transfers.

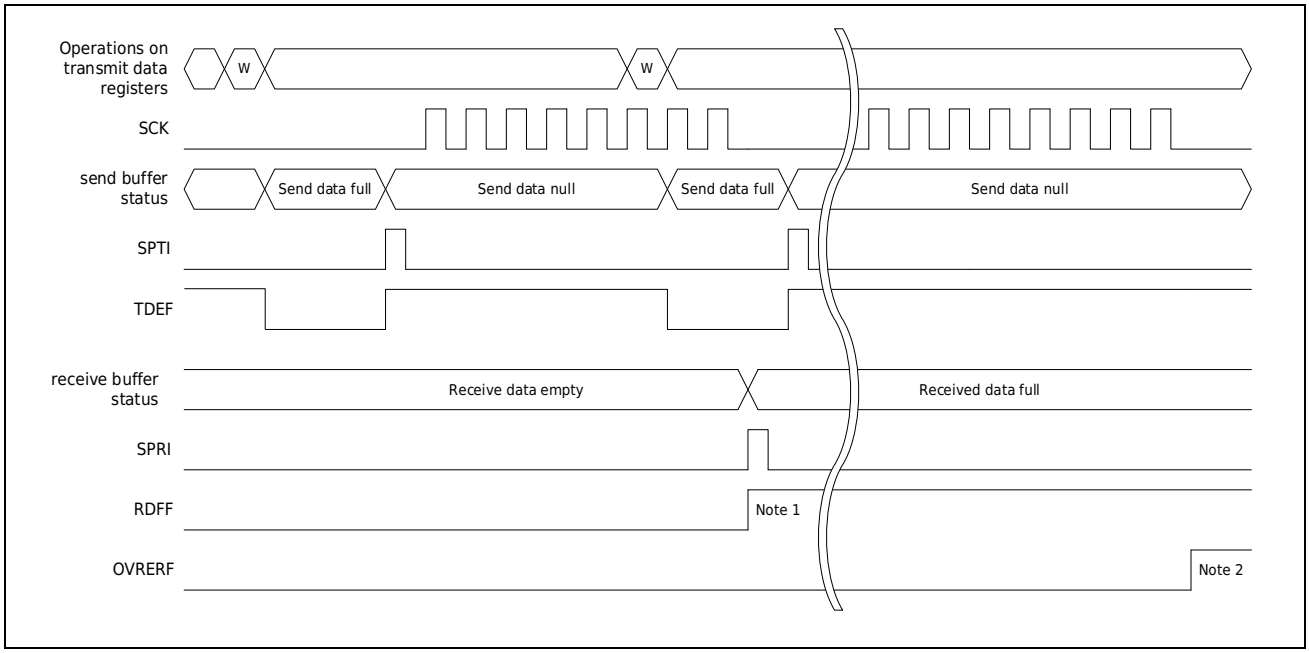


Figure 29-11 Full Duplex Synchronous Serial Communication

Note:

1. *At the end of this serial transmission, if the receive data buffer register is empty, the SPI will copy the received data from the shift register to the receive data buffer register, the flag bit full of the receive data buffer register is set to 1 (RDFF), and a interrupt request (SPRI) full of the receive data is generated.*
2. *When the serial transmission ends, if the receive data buffer register 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 invalid, and the data in the receive shift register will be discarded.*

2) Send only communication method

When the SPI_CR.TXMDS bit is 1, the SPI operates in send-only communication mode. As shown in Figure 29-12, the SPI_CFG1.FTHLV[1:0] bits are "0b00", the SPI_CFG2.CPHA bits are 1 and the SPI_CFG2.CPOL bits are 0, and the SPI performs 8-bit serial transfers.

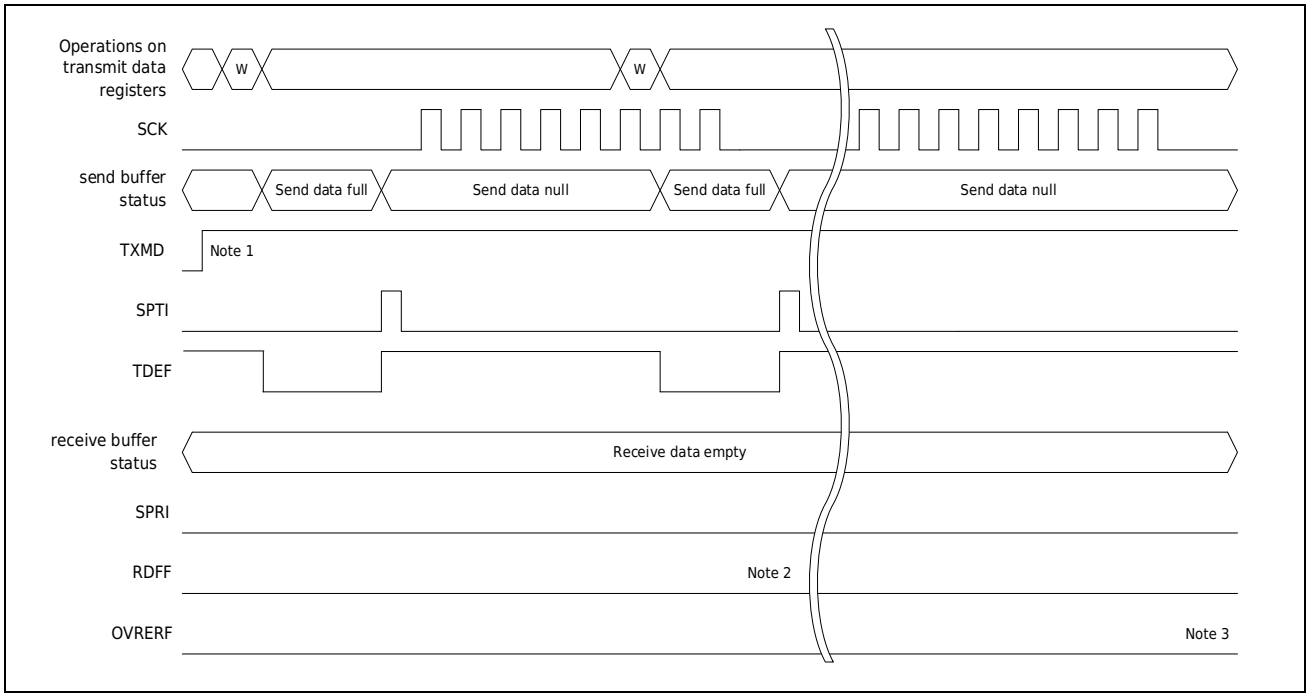


Figure 29-12 Sending Communication Only

Note:

1. *Make sure that there is no unread data in the receive buffer register (that is, RDFF is 0) and no data overload error (that is, OVRERF is 0) before setting into send-only communication.*
2. *At the end of this serial transmission, even if the receive data buffer register is empty, no data will be accepted, and RDFF will always remain 0.*
3. *The OVRERF flag bit is always in the 0 state because the receive data buffer register is always empty by sending only the communication mode.*

29.3.3.5 Communication Interval Time

1) Communication Interval under Normal Settings

When the SPI_CFG1.CTMDS bit is 0, the SPI uses normal communication settings. The interval time between two consecutive communications can be set by SPI_CFG1.MIDI[2:0] and SPI_CFG2.MIDIE. The minimum interval time is 1*SCK+2*PCLK1. As shown in Figure 29-13, the SPI_CFG2.CPHA bit is 1 and the SPI_CFG2.CPOL bit is 0, and the SPI performs serial transfer.

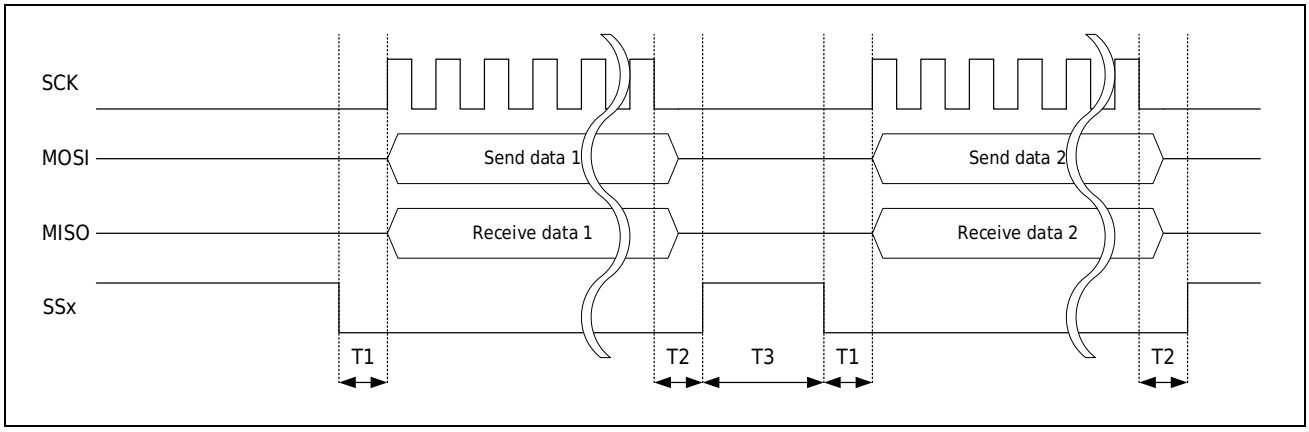


Figure 29-13 Universal Communication

In the above figure, t1 represents the time interval from the valid SS signal to the first edge of SCK (SCK delay time, which can be controlled by setting SPI_CFG1.MSSI[2:0] and SPI_CFG2.MSSIE), t2 represents the time interval from the last edge of SCK to the invalid SS signal (SS invalid delay time, which can be controlled by setting SPI_CFG1.MSSDL[2:0] and SPI_CFG2.MSSDLE), and t3 represents the minimum waiting time from the end of this transmission to the beginning of the next transmission (the next access data delay, which can be controlled by setting SPI_CFG1.MIDI[2:0] and SPI_CFG2.MIDIE).

Note:

1) *When the communication rate is 2 divided by PCLK1 (SPI_CFG2.MBR[1:0]=0b00 and SPI_CFG2.CLKDIV[3:0]=0b0000), please configure t3 to 2*SCK+2*PCLK1 or above, that is, SPI_CFG2.MIDIE=1 and SPI_CFG1.MIDI[2:0]=1~7.*
2) Communication interval time under continuous communication setting

When the SPI_CFG1.CTMDS bit is 1, the SPI uses the continuous communication setting. Completely continuous and uninterrupted data communication can be achieved. As shown in the Figure 29-14, the SPI_CFG2.CPHA bit is 1 and the SPI_CFG2.CPOL bit is 0, and the SPI performs serial transmission.

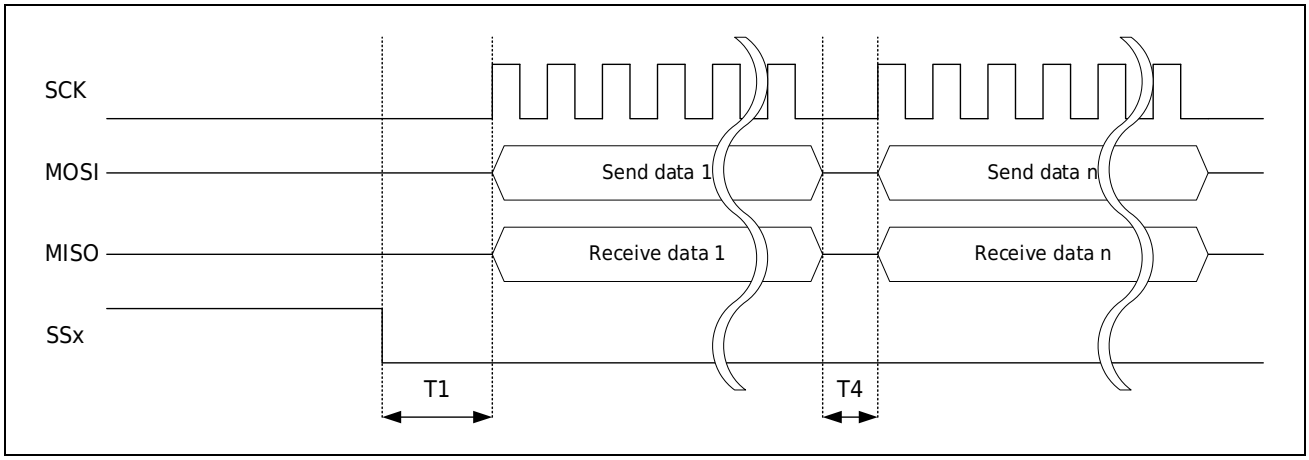


Figure 29-14 Continuous Communication

In the above figure, t1 represents the time interval from the validity of the SS signal to the first edge of SCK (SCK delay time, which can be controlled by setting SPI_CFG1.MSSI[2:0] and SPI_CFG2.MSSIE), and t4 represents the time interval between each transmission, which can be controlled by setting SPI_CFG1.MSSI[2:0] and SPI_CFG2.MSSIE. The relationship between t1 and t4 is defined as t4 = t1-1*SCK.

Note:

1) *In continuous communication mode, if the master has no new data written into the Tx Buffer register, it will enter a pause state, SCK stops outputting, and the SS signal remains valid, waiting for new send data to be written into the Tx Buffer register before continuing to send the next frame of data.*
2) *In continuous communication mode, if the slave has no new data written into the Tx Buffer register, an underrun error will occur, and SPI_SR.MODFERF and SPI_SR.UDRERF will be set to 1. For details, please refer to the description of underrun error.*
3) *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.*
4) *When using the transmission format of CPHA=0, the slave needs to write new data to the Tx Buffer register before the previous frame of data is sent. Otherwise, an underload error will occur in the slave regardless of whether the master starts the communication of the next frame of data.*
5) *In continuous communication mode, the time interval between the last SCK of the previous frame data and the first SCK of the next frame data can also be adjusted through MSSI.*

29.3.4 Operating Instructions

29.3.4.1 Operational Mode Summary

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 MSTR and SPIMDS bits in the SPI control register SPI_CR 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 Table 29-6.

Table 29-6 SPI Pattern and Register Setting Relationship

Mode	Master (SPI operation)	Slave (SPI running)	Master (clock synchronization 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 running)	Master (clock synchronization 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	~PCLK1/2	~PCLK1/6	~PCLK1/2	~PCLK1/6
Timer	Internal baud rate generator	SCK input	Internal baud rate generator	SCK input
Clock polarity	2 species	2 species	2 species	2 species
Clock phase	2 species	2 species	2 species	2 types (Note 3)
Start transfer bit	MSB/LSB	MSB/LSB	MSB/LSB	MSB/LSB
Transmission data length	4~32 bits	4~32 bits	4~32 bits	4~32 bits
SCK delay control	Yes	No	Yes	No
SS invalid delay control	Yes	No	Yes	No
Next Access Delay Control	Yes	No	Yes	No
Transport start method	Write send buffer by sending buffer empty interrupt request	SS Input Valid or SCK Clock Edge	Write send buffer by sending buffer empty interrupt request	SCK oscillation
Mode Failure Error	No	Yes	No	Yes
Transmit buffer air detection	Yes	Yes	Yes	Yes
Receive buffer full detection	Yes (note 1)	Yes (note 1)	Yes (note 1)	Yes (note 1)
Overload error detection	Yes (note 1)	Yes (note 1)	Yes (note 1)	Yes (note 1)
Parity error detection	Yes (note 1, note 2)	Yes (note 1, note 2)	Yes (note 1, note 2)	Yes (note 1, note 2)
Underload error detection	No	Yes	No	Yes

Note:

1) When the SPI_CR.TXMDS bit is 1, the Rx Buffer full detection, overrun error detection and parity check error detection are not performed.

2) When the SPI_CR.PAE bit is 0, no parity check error detection is performed.

3) When the SPI_CFGR2.CPHA bit is 0, it is only valid in continuous transmission mode. For specific setting steps, please refer to [*Slave Maneuver in Clock Synchronous Operation Mode*]

29.3.4.2 Master Action in SPI Operation Mode

1) Action Description for as master

When the SPI data transmission buffer register (TX_BUFF) is empty (the TDEF flag in the status register SPI_SR is 1), write the data of the frame length set by the FTHLV[1:0] bit of the format control register SPI_CFG1 to the SPI After the data register (SPI_DR), SPI will update the SPI_DR data to TX_BUFF. If the shift register (shifter) is empty, SPI copies the TX _ BUFF data to the shift register to start serial transfer.

When transmitted data is copied to shifter, SPI will change shifter state to full state; When serial delivery is over, change to null. The status of shifter cannot be read.

When the SCK edge is needed to send the last sampling sequence of SPI, the end time of this serial transmission and SPI _ CFG2.CPHA position was independent. 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 data registry SPI _ DR.

The final sampling sequence depends on the bit length of the transmitted data, and the SPI data length of the master mode depends on SPI_CFG2.DSIZE[3:0].The polarity of the SS output pin depends on the set value of SPI _ CFG1 register. For details about the SPI transmission format, please refer to [Transfer format].

2) Initialization of SPI Master Mode

① Set the communication configuration register 1 (SPI_CFG1), including the setting of the baud rate, the setting of the number of frames used, the setting of various delay times, etc.;

② Set the communication configuration register 2 (SPI_CFG2), including SS level setting, data shift sequence setting, various delay allow bit setting, data format and clock polarity phase setting, etc.;

③ If you need to use interrupt, set the interrupt register of the system;

④ If you need to use DMA, set the relevant register for DMA;

⑤ Set the input/output pin;

⑥ Set the SPI control register SPI_CR, including the mode and operation mode settings, self-diagnosis function settings, parity check settings, etc.

⑦ Confirm the setting of SPI_CR register;

⑧ Clear various flag bits;

⑨ Set the interrupt permission bit;

⑩ Set the SPE bit of the control register SPI_CR to 1 and the action starts.

29.3.4.3 The Slave Action in SPI Operation Mode

1) SPI as Slave Action Description

When SPI _ CFG2.When the 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 invalid to valid 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 valid 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 null 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 delivery, SPI keeps shifter full.

If SPI detects the SCK edge of the last sampling sequence, then this serial transmission ends with SPI _ CFG2.CPHA position was independent. When RX _ BUFF is null, SPI copies shifter's acceptance data 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 failure error occurs if SPI detects that the SS0 input signal is invalid.

The final sampling sequence depends on the bit length of the transmitted data, and the SPI data length depends on SPI_CFG2.DSIZE[3:0].The polarity of SS0 input signal depends on SPI _ CFG1.SS0PV bit setting. For details about the SPI transmission format, please refer to [Transfer format].

Note:

– *When the SPI_CFG2.CPHA bit is 0, the change in SS0 input signal level from invalid to valid is regarded 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.*

2) SPI Slave Mode Initialization

① Set the communication configuration register 1 (SPI_CFG1), mainly including the setting of the number of frames used;

② Set up communication configuration register2 (SPI _ CFG2), including transmission rate, data format and clock polarity phase setting;

③ If you need to use interrupt, set the interrupt register of the system;

④ If you need to use DMA, set the relevant register for DMA;

⑤ Set the input/output pin;

⑥ Set the SPI control register SPI_CR, including the mode and operation mode settings, self-diagnosis function settings, parity check settings, etc.

⑦ Confirm the setting of SPI_CR register;

⑧ Clear various flag bits;

⑨ Set the interrupt permission bit;

⑩ Set the SPE bit of the control register SPI_CR to 1 and the action starts.

29.3.4.4 Master Action in Clock Synchronous Operation Mode

When the SPIMDS bit in the SPI control register SPI_CR is 1, the SPI is in clock synchronization operation mode. When operating in this mode, SPI only uses the three 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 mode of operation, the internal operation of the module is the same as the SPI mode of operation.

1) Action Description for SPI as Master

When the SPI data transmit buffer register (TX_BUFF) is empty (the TDEF flag in the status register SPI_SR is 0), after writing the data of the frame length set by the FTHLV[1:0] bits of the format control register SPI_CFG1 to the SPI data register (SPI_DR), the SPI will update the SPI_DR data to TX_BUFF. If the shift register (shifter) is empty, SPI copies the TX _ BUFF data to the shift register to start serial transfer.

When transmitted data is copied to shifter, SPI will change shifter state to full state; When serial delivery is over, change to null. The status of shifter cannot be read.

When the SCK edge is needed to send the last sampling sequence of SPI, the end time of this serial transmission and SPI _ CFG2.CPHA position was independent. 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 data registry SPI _ DR.

The final sampling timing depends on the bit length of the sent data. The SPI data length in master mode depends on the setting value of the SPI_CFG2.DSIZE[3:0] bit. For details about the SPI transmission format, please refer to [传送格式].

2) Initialization Setting of master in Clock Synchronous Operation Mode

① Set the communication configuration register 1 (SPI_CFG1), including the setting of the baud rate, the setting of the number of frames used, the setting of various delay times, etc.;

② Set the communication configuration register 2 (SPI_CFG2), including the data shift sequence setting, the setting of the allowable bit of various delays, the setting of the data format and the clock polarity phase, etc.;

③ If you need to use interrupt, set the interrupt register of the system;

④ If you need to use DMA, set the relevant register for DMA;

⑤ Set the input/output pin;

⑥ Set the SPI control register SPI_CR, including the mode and operation mode settings, self-diagnosis function settings, parity check settings, etc.

⑦ Confirm the setting of SPI_CR register;

⑧ Clear various flag bits;

⑨ Set the interrupt permission bit;

⑩ Set the SPE bit of the control register SPI_CR to 1 and the action starts.

29.3.4.5 Slave Maneuver in Clock Synchronous Operation Mode

1) SPI as Slave Action Description

When SPI _ CFG2.When the CPHA bit is 0, SPI needs to detect that the SS0 input signal becomes active as a trigger signal for starting serial communication. Since the SS0 pin is not used in the clock synchronous operation mode, the following settings must be followed for normal communication.

① Perform communication settings and set the SPE bit of SPI_CR to 1;

② Set SPI_CFG2.CPHA to 1;

③ Set SPI_CFG1.CTMDS to 1;

④ Write the data to be sent into SPI_DR;

⑤ Set SPI_CFG2.CPHA to 0;

⑥ Waiting for the master clock to start communicating.

When the CPHA bit is "1", if SPI detects the original SCK edge when the SS0 input signal is at an active level, it needs to start to drive valid 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 null 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 delivery, SPI keeps shifter full.

If SPI detects the SCK edge of the last sampling sequence, the serial transmission ends. When RX _ BUFF is null, SPI copies shifter's acceptance data 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 the setting value of the SPI_CFG2.DSIZE[3:0] bits.

2) Initialization Settings of the Slave when the Clock Synchronizes the Operation Mode

① Set the communication configuration register 1 (SPI_CFG1), mainly including the setting of the number of frames used;

② Set up communication configuration register2 (SPI _ CFG2), including transmission rate, data format and clock polarity phase setting;

③ If you need to use interrupt, set the interrupt register of the system;

④ If you need to use DMA, set the relevant register for DMA;

⑤ Set the input/output pin;

⑥ Set the SPI control register SPI_CR, including the mode and operation mode settings, self-diagnosis function settings, parity check settings, etc.

⑦ Confirm the setting of SPI_CR register;

⑧ Clear various flag bits;

⑨ Set the interrupt permission bit;

⑩ Set the SPE bit of the control register SPI_CR to 1 and the action starts.

29.3.4.6 Processing Flow Of Several Spi Actions

1) SPI data transfer process as master

① Wait for the empty interrupt of the data sending buffer register or confirm by polling that the data sending buffer register is empty;

② Writes the data to send to the data registry SPI_DR;

③ Repeat steps ① and ② until the last data is sent;

④ Reset the allowable bit TXIE of send data register empty interrupt, and set the allowable bit IDIE of SPI idle state interrupt to 1;

⑤ The hardware generates an SPI idle state interrupt;

⑥ Set SPE to 0, stop SPI, and reset IDIE.

2) Data reception process

① Wait for the full interrupt of the data receiving buffer register or confirm by polling that the data receiving buffer register is full;

② Read data from the receive buffer register by accessing SPI _ DR;

③ Repeat steps ① and ② until the last received data is read;

④ Clears the RXIE allow bit RXIE for the full interrupt of the data receiving buffer register.

3) Communication error handling process

① Wait for the communication error interrupt or confirm by polling that the communication error flag bit (MODFERF/OVERRF/UDRERF/PERF)is set to 1;

② Verify SS0 status and troubleshoot mode errors;

③ Reset SPE and stop SPI;

④ Identify the types of communication errors by error flag bits and handle them;

⑤ Clear the error flag bit;

⑥ Start SPI and restart the communication.

29.3.5 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.

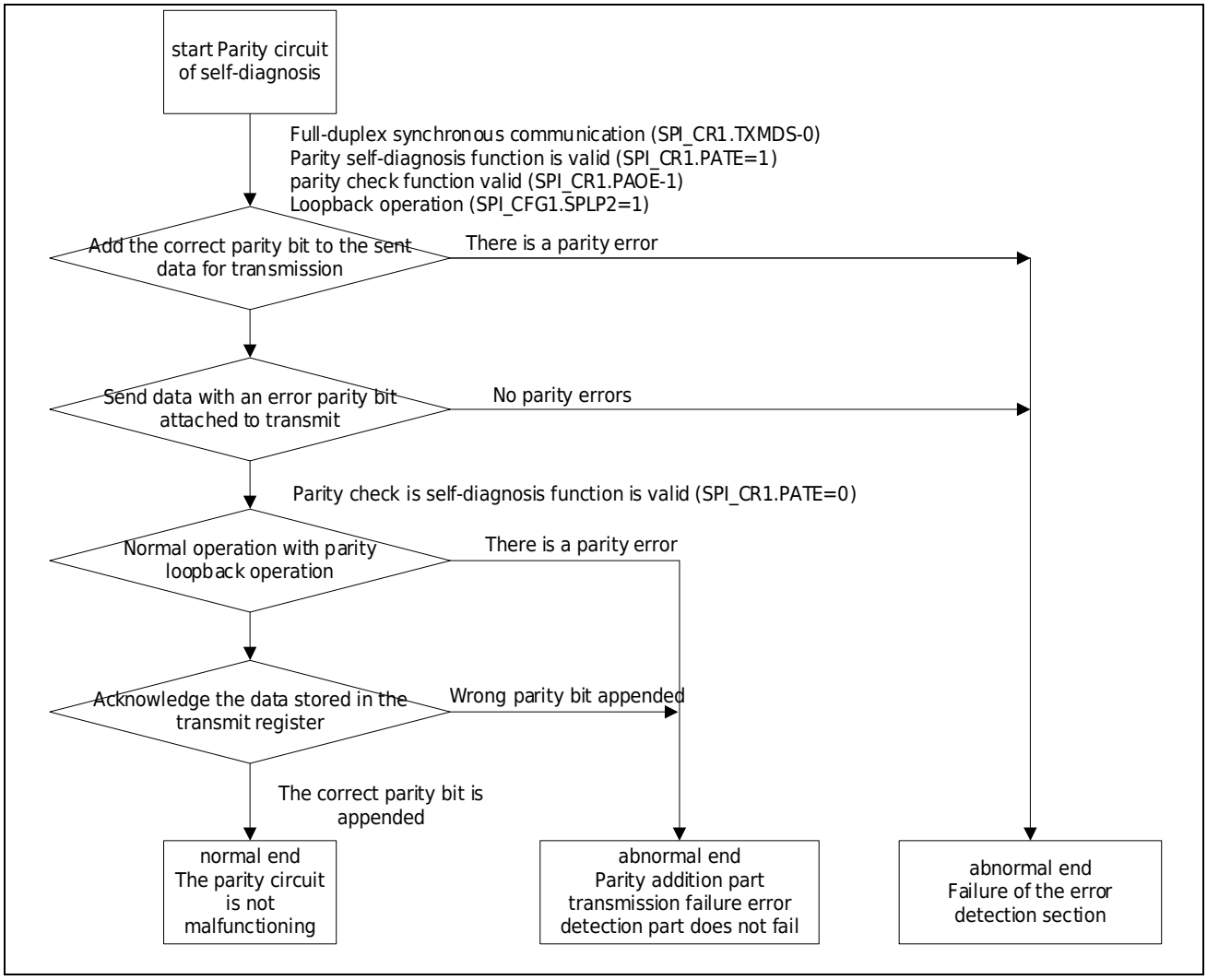


Figure 29-15 Parity Checking Process

29.3.6 Error Detection

In the normal SPI serial transmission, the system sends 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 failure error. The correspondence between abnormal transmission and SPI error detection is shown in the following Table 29-7.

Table 29-7 Correspondence Table for Error Detection

Serial number	Occurrence condition	SPI operation	Detect errors
①	Write SPI _ DRregister in full transmit buffer	Keep sending buffer content	No
		Write data loss	
②	Read SPI _ DRregister when Receive Buffer is empty	Output last serial received data	No
③	In slave mode, serial delivery begins without data being transferred to the shift 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 Loss of sending and receiving data Stop SPI function	schema error
⑤	End serial transfer in full receiving 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 effective	Parity error flag is valid	Parity error

①In the cases described, SPI does not detect errors. To prevent data omissions during data write to SPI _ DRregister, data must be written to SPI _ DRregister by an empty interrupt of the send buffer. Similarly, SPI does not detect errors. To prevent unrelated data from being read, SPI _ DR data read must be done by receiving a register full interrupt request.

29.3.6.1 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 valid level, and the SPI has an underrun error, SPI_SR.MODFERF and SPI_SR. The UDRERF flag will be set to 1.

When an underrun error is detected, the SPI will stop driving the signal output and set SPI_CR.SPE to 0.

Monitoring of underload errors can be done by accessing SPI _ SRregister 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 first.

29.3.6.2 Pattern Error

When SPI is in master mode, do not set SPI_CR.MODFE to 1. In slave mode, when the SSi valid level width does not reach the time required to send data, a mode fault occurs, SPI_SR.MODFERF is set to 1, and SPI_CR.SPE is set to 0. When transmission is required, clear SPI_SR.MODFREF and then set SPI_CR.SPE to 1.

29.3.6.3 Overload Error

SPI _ SR.OVRERF flag is set to 1 if serial delivery is ended in full receive buffer. Since SPI does not copy the data of the shift register to the receive buffer in the case of an OVRERF flag of 1, the receive buffer stores the received data before an error occurs. The OVRERF flag must be set to 1 before the SPI_SR register can be read.

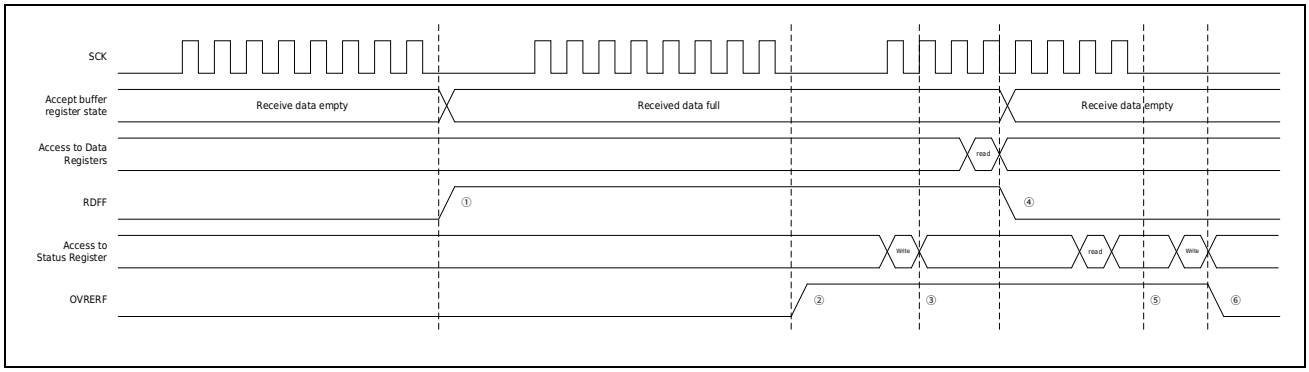


Figure 29-16 Overload Error Handling

The following describes the operation content of the flags in the sequence shown in ①~⑥ in the figure:

① The serial transmission ends when the receiving buffer is empty, and the SPI operates normally, copying the data of the shift register to the receiving buffer register, and setting the RDFF flag 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. SPI copies the data of the and will not set the shift register to the receive buffer. Even when the PAE bit is 1, a parity error is not detected.

③ Clear the OVRERF flag bit.

④ Read access to the data register SPI_DR 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 shift register to the receive buffer register, 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 shift register to the receive buffer, the SPI judges that the shift register is empty and allows data to be transferred from the transmit buffer register transfer to the shift register.

⑥ Normal receive operation is only possible after changing the OVRERF flag 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 an overload error must be detected as early as possible by methods such as reading the SPI_SR register immediately after reading the SPI_DR register.

In master mode, if the communication automatic suspend function is enabled (set the SPI_CR.CSUSPE bit to 1), the SPI will suspend the communication clock in the last sampling cycle before the overload error occurs. At this time, since the shift register has not completed the reception of the last bit, the SPI remains in the normal communication state and the overload error will not occur. During the suspension of the communication clock, the receiving buffer register can be read. After reading, the status of the receiving buffer register becomes empty, and the SPI restarts the communication clock to complete the last bit of data reception. For details, please refer to the in the Figure 29-17 and Figure 29-18.

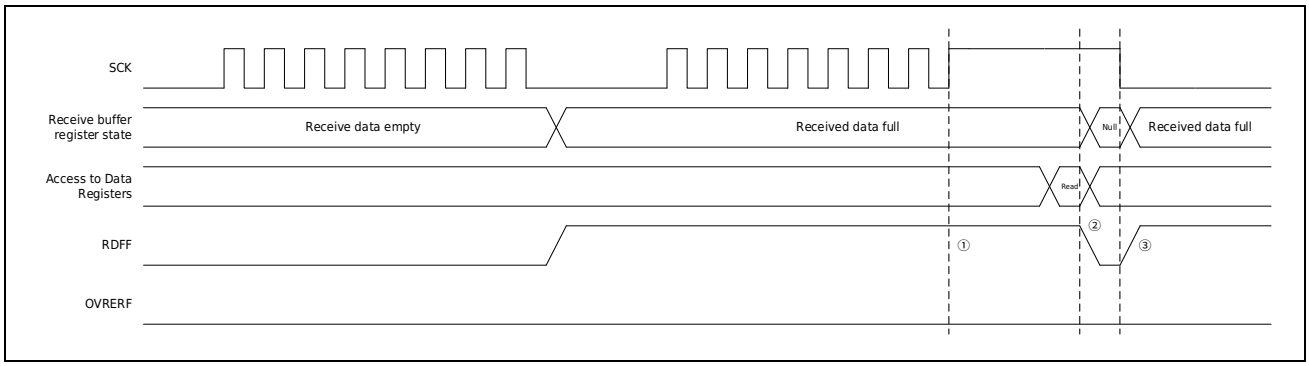


Figure 29-17 Schematic diagram of the action when the clock automatic stop function is enabled (CPHA=1)

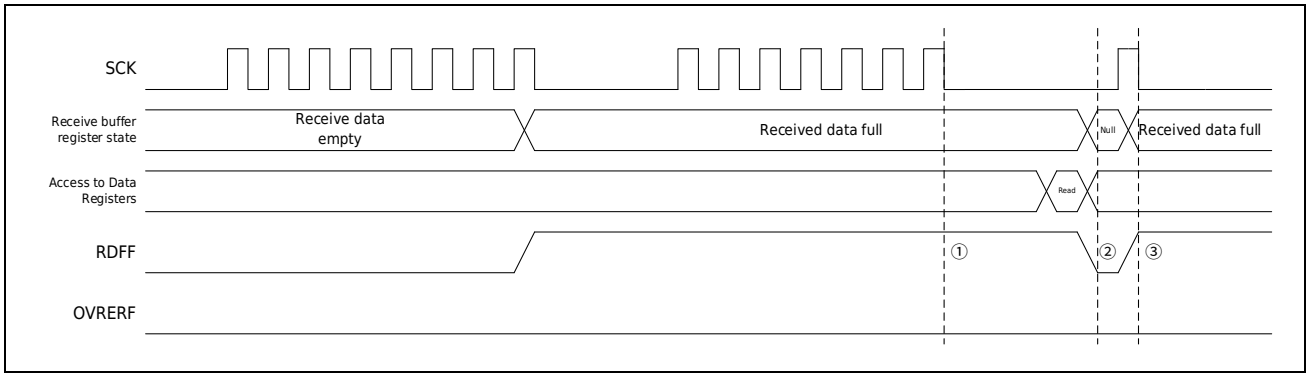


Figure 29-18 Schematic diagram of the action when the clock automatic stop function is enabled (CPHA=0)

The following describes the operation of the flags in the sequence shown in ①~③ of the figure:

① When the receive buffer register is full, the SPI suspends the communication clock before the last bit of data is received. At this will not occur Overload error.

② After reading the data in the receiving buffer register by accessing SPI_DR, the receiving buffer

register becomes empty, the RDFF flag is cleared, and the SPI restarts the communication clock to complete the last bit of data communication.

③ The last bit of data communication is completed, the receive buffer register becomes full again, the RDFF flag is set to 1, and the received data can be read by accessing SPI_DR.

29.3.6.4 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 shift register to the receive buffer. To clear the PERF flag, you need to read SPI _ SRregister with PERF flag 1 before writing 0 to the PERF flag.

An example of the operation of the OVRERF flag and the PERF flag is shown in Figure 29-19 below. 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.

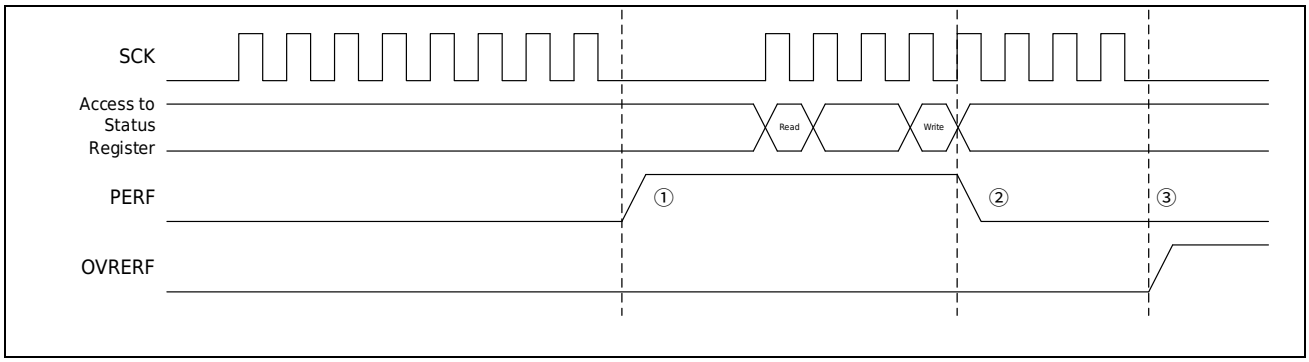


Figure 29-19 Parity Error

The following describes the operation of the flags in the sequence shown in Figure ①~③:

① SPI did not detect an overload error and serial delivery ended normally. SPI copies the data of the shift 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 shift register to the receiving buffer, SPI does not perform parity check on the data, and no parity check error occurs.

The occurrence of parity errors can be monitored by accessing SPI _ SR register directly or by terminating SPI _ SR register via SPI error. The occurrence of parity errors must be monitored as soon as possible by means of access state registerSPI _ SR.

29.3.7 SPI Initialization

Write operation or mode fault error detection is used to zero SPE bits, which can invalidate SPI functions and initialize some SPI functions. If a system reset occurs, all SPI functions are initialized.

29.3.7.1 Clear SPE Bits for Initialization

When the SPI_CR.SPE bit is 0, the SPI performs the following initialization operations:

- Abort in-progress serial delivery.
- If you are in the slave state, stop driving the output signal (the state changes to Hi-Z).
- SPI internal state is initialized.
- Clear send buffer register, SPI _ SR.TDEF flag set to 1.

SPI control bits are not initialized by zeroing SPE bits. As a result, SPI can be started in the same transport mode as before initialization, as long as SPE is relocated to 1.

Clearing SPE bits 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 the SPE bit clears the send buffer register and sets the SPI _ SR.TDEF flag to 1. Therefore, if the SPI_CR.TXIE bit is set to 1 after initialization, an SPI Tx Buffer register null interrupt will be generated. To avoid this interrupt, you must reset the SPE bit to zero while setting the TXIE bit to zero.

29.3.7.2 System Reset Initialization

All SPI control bits, status bits and data registers are initialized by system reset.

29.4Interruptand Events

29.4.1 Interrupt

The SPI Rx Buffer full, Tx Buffer null, mode fault (overload, underload, parity check error) and SPI idle can be used as interrupt sources or internal trigger sources, and 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 the SPI interrupt/internal trigger source is shown in Table 29-8. Once the condition is met, the corresponding interrupt/internal trigger request is generated. For receive buffer full and transmit buffer empty interrupt sources, data transfer changes buffer status to clear.

Table 29-8 SPI interrupt Source Description

Interrupt/Internal Trigger Source	Sketch	Conditions
Receiver buffer full	SPRI	When the Rx Buffer becomes full while the SPI_CR.RXIE bit is 1
Tx Buffer Null	SPTI	When the Tx Buffer becomes null while the SPI_CR.TXIE bit is 1
SPI errors (overload, underload, parity errors)	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	SPII	When the SPI_CR.IDIE bit is 1 Master mode: data transfer is completed or SPI_CR.SPE is written from 1 to 0. Slave mode: SPI_CR.SPE is written from 1 to 0.
Transfer End	SPEND	Master mode: Data Transfer End. Slave mode: Four-wire, when the SSL pin is at an invalid level; three-wire, when the SCK pin clock reaches the number of clocks required for data transmission

29.4.2 Events

SPI produces the following types of event trigger sources available:

- Data sending buffer register empty
- Data reception buffer register full
- SPI communication errors (including overload, underload, parity, etc.)
- SPI is idle
- End of SPI communication

Users can write the vector corresponding to the event trigger source into different trigger object registers to implement various event trigger functions.

For the vector corresponding to the event trigger source, refer to [Interrupt controller (INTC)].

29.5Register Description

Table 29-9 SPI Base Address

Name	Base Address	Description
SPI	0x4001 C000	SPI Base Address

Table 29-10 SPI Register List

Register	Description	Offset Address	Bit Width	Reset Value
SPI_DR	SPI data register	0x00	32	0x0000 0000
SPI_CR	SPI control register	0x04	32	0x0000 0000
SPI_CFG1	SPI Communication Configuration Register 1	0x0C	32	0x0000 0010
SPI_SR	SPI status register	0x14	32	0x0000 0020
SPI_CFG2	SPI Communication Configuration Register 2	0x18	32	0x0000 0F1D

29.5.1 SPI Data Register (SPI _ DR)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
SPD[31:16]															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
SPD[15:0]															

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b0	SPD[31:0]	Serial data	SPI Data Storage	RW

29.5.2 SPI Control Register (SPI_CR)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
PAE	PAOE	PATE	MODFE	IDIE	RXIE	TXIE	EIE	CSUSPE	SPE	SPLPBK2	SPLPBK	MSTR	Res	TXMDS	SPIMDS

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b16	Res	Reserved Bit	Read at “0”, write at “0”	RW
b15	PAE	Parity permissible	0: Send data without parity bits, receive data without parity check 1: Send data with parity check bit, and receive data with parity check (SPI_CR.TXMDS=0); Send data with parity check bit, and receive data without parity check (SPI_CR.TXMDS=1)	RW
b14	PAOE	Parity mode selection	0: Select parity for sending and receiving 1: Select odd check for sending and receiving	RW
b13	PATE	Parity self-diagnosis	0: The parity self-diagnostic function is invalid 1: Parity self-diagnosis function is valid	RW
b12	MODFE	Mode fault error detection permit	0: Disable mode fault error detection 1: Allow mode fault error detection	RW
b11	IDIE	SPI Idle Interrupt Allowed	0: Disable Idle Interrupt Request Generation 1: Allow Idle Interrupt Request Generation	RW
b10	RXIE	SPI Receive Interrupt Permit	0: Disable SPI from receiving interrupt requests 1: Allow SPI to receive interrupt request generation	RW
b9	TXIE	SPI Send Interrupt Permit	0: Disable SPI from sending interrupt requests 1: Allow SPI to send interrupt request generation	RW
b8	EIE	SPI error interrupt permission	0: Disable SPI error interrupt request generation 1: Allow SPI error interrupt request generation	RW
b7	CSUSPE	Communication auto suspend function Permissible	0: Communication auto suspend function is invalid 1: Communication auto-suspend function is valid	RW
b6	SPE	SPI Feature Allowed	0: Invalid SPI feature 1: The SPI function is valid	RW
b5	SPLPBK2	SPI Loop 2-bit	0: Normal mode 1: Loopback mode (send data = receive data)	RW
b4	SPLPBK	SPI loopback position	0: Normal mode 1: Loopback mode (inverse of transmitted data = received data)	RW
b3	MSTR	SPI Master-Slave Mode Selection	0: Slave mode 1: master mode	RW
b2	Res	Reserved Bit	Read at “0”, write at “0”	RW
b1	TXMDS	Communication mode selection	0: Full-duplex synchronous serial communication 1: Send Serial Communication Only	RW
b0	SPIMDS	SPI mode selection	0: SPI operation (4-line) 1: Clock synchronous operation (3-wire)	RW

29.5.3 SPI Communication Configuration Register1 (SPI _ CFG1)

Reset value: 0x0000 0010

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res	MIDI[2:0]			Res	MSSDL[2:0]			Res	MSSI[2:0]			Res			
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
CLKDIV[3:0]				SS3PV	SS2PV	SS1PV	SS0PV	Res	SPRDTD	Res			CTMDS	FTHLV[1:0]	

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31	Res	Reserved Bit	Read at “0”, write at “0”	RW
b30~b28	MIDI[2:0]	Master next access data interval idle time setting bit	000: 1 SCK + 2 PCLK1s 001: 2 SCK + 2 PCLK1s 010: 3 SCK + 2 PCLK1s 011: 4 SCK + 2 PCLK1s 100: 5 SCK + 2 PCLK1s 101: 6 SCK + 2 PCLK1s 110: 7 SCK + 2 PCLK1s 111: 8 SCK + 2 PCLK1s	RW
b27	Res	Reserved Bit	Read at “0”, write at “0”	RW
b26~b24	MSSDL[2:0]	Master SS invalid delay setting bit	00 0: 1 SCK 001: 2 SCKs 010: 3 SCKs 011: 4 SCKs 100: 5 SCKs 101: 6 SCKs 110: 7 SCKs 111: 8 SCKs	RW
b23	Res	Reserved Bit	Read at “0”, write at “0”	RW
b22~b20	MSSI[2:0]	Master SS idle time setting bit	000: 1 SCKs 001: 2 SCKs 010: 3 SCKs 011: 4 SCKs 100: 5 SCKs 101: 6 SCKs 110: 7 SCKs 111: 8 SCKs	RW
b19~b16	Res	Reserved Bit	Read at “0”, write at “0”	RW
b15~b12	CLKDIV	Communication reference clock selection bit	0000：PCLK1/2 0001：PCLK1/4 00 10：PCLK1/6 0011：PCLK1/8 0100：PCLK1/10 0101：PCLK1/12 0110：PCLK1/14 0111：PCLK1/16 1000：PCLK1/18 1001：PCLK1/20 1010：PCLK1/22 1011：PCLK1/24 1100：PCLK1/26 1101：PCLK1/28 1110：PCLK1/30 1111：PCLK1/32	RW
b11	SS3PV	SS3 signal polarity set	0: Low level of SS3 signal valid 1: High level of SS3 signal valid	RW

b10	SS2PV	SS2 signal polarity set	0: Low level of SS2 signal valid 1: High level of SS2 signal valid	RW
b9	SS1PV	SS1 signal polarity set	0: Low level of SS1 signal valid 1: High level of SS1 signal valid	RW
b8	SS0PV	SS0 signal polarity set	0: Low level of SS0 signal valid 1: High level of SS0 signal valid	RW
b7	Res	Reserved Bit	Read at “0”, write at “0”	RW
b6	SPRDTD	Data register read target selection	0: SPI_DR read receive buffer 1: SPI_DR read send buffer (must be read when TDEF=1)	RW
b5~b3	Res	Reserved Bit	Read at “0b010”, write at “0b010”	RW
b2	CTMDS	Continuous communication mode selection	0: Normal communication mode 1: Continuous communication mode	RW
b1~b0	FTHLV[1:0]	FPS	00: 1 frame 01: 2 frames 10: 3 frames 11: 4 frames	RW

29.5.4 SPI Status Register (SPI _ SR)

Reset value: 0x0000 0020

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res								RDFF	FMIDLF	TDEF	UDRERF	PERF	MODFERF	IDLNF	OVRERF

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b8	Res	Reserved Bit	Read at “0”, write at “0”	RW
b7	RDFF	Receive buffer full flag	0: The Rx Buffer is not full. 1: The Rx Buffer is full. Hardware set, clear	R
b6	FMIDLF	Continuous mode master IDLE status flag	0: Continuous mode master is in normal operation state 1: Continuous mode master is in idle state	R
b5	TDEF	Transmit buffer null flag	0: The transmit buffer has data 1: No data in the occurrence buffer Hardware set, clear	R
b4	UDRERF	Underloaded error mark	0: No underload error occurred 1: Underload error occurs (MODFERF = 1) This bit is initialized when MODFERF = 0 Hardware setting alignment After reading 1 0 and reset.	RW
b3	PERF	Parity error flag	0: No parity error occurred 1: Parity error Hardware setting alignment After reading 1 0 and reset	RW
b2	MODFERF	Mode fault flag	0: Unsent mode failure error 1: Mode failure error Hardware setting alignment After reading 1 0 and reset	RW
b1	IDLNF	SPI Idle Mark	0: SPI is idle 1: SPI is transport Hardware set, clear	R
b0	OVRERF	Overload error mark	0: No overload error occurred 1: Overload error Hardware setting alignment After reading 1 0 and reset	RW

29.5.5 SPI Communication Configuration Register2 (SPI _ CFG2)

Reset value: 0x0000 0F1D

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
MSSIE	MSSDLE	MIDIE	LSBF	DSIZE[3:0]				SSA[2:0]			Res	MBR[1:0]		CPOL	CPHA

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b16	Res	Reserved Bit	Read at “0”, write at “0”	RW
b15	MSSIE	SCK delay allows	0: SCK delay is 1 SCK 1: SCK delay is the set value of SSI	RW
b14	MSSDLE	SS Invalid delay allows	0: SS invalid delay is 1 SCK 1: SS Invalid delay is the set value of SSDL	RW
b13	MIDIE	SPI next delay allows	0: The next access delay is 1 SCK+2 PCLK1 1: next delay is the set value of MIDI	RW
b12	LSBF	SPI LSB first bit	0: MSB first 1: LSB first	RW
b11~b8	DSIZE[3:0]	Positioning SPI Data Length	0000: 4 bits 0001: 5 bits 0010: 6 bits 0011: 7 bits 0100: 8 bits 0101: 9 bits 0110: 10 bits 0111: 11 bits 1000: 12 bits 1001: 13 bits 1010: 14 bits 1011: 15 bits 1100: 16 bits 1101: 20 bits 1110: 24 bits 1111: 32 bits	RW
b7~b5	SSA[2:0]	SS signal valid setting bit	000：SS0 001：SS1 010：SS2 011：SS3 Other: prohibitions	RW
b4	Res	Reserved Bit	Read at “1”, write at “1”	RW
b3~b2	MBR[1:0]	Bit Rate Frequency Division Location	00: Select the reference communication clock 01: Select the 2-division of the reference communication clock 10: Select the 4-division of the reference communication clock 11: Select the 8-division of the reference communication clock	RW
b1	CPOL	SCK polarity set bit	0: Low SCK at idle 1: SCK is high when idle	RW
b0	CPHA	SCK phase setting bit	0: Data sampling at odd edges, changing at even edges 1: Changes in data at odd edges and data sampling at even edges	RW

30 Controller Area Network (MCAN)

30.1 Overview

The Controller Area Network (CAN) subsystem consists of two CAN modules and a shared message RAM memory. For the base addresses of these three components, please refer to chapter [Memory mapping].

Both CAN modules (MCAN1 and MCAN2) comply with ISO 11898-1: 2015 (CAN Protocol Specification Version 2.0 Part A, B) and CAN FD Protocol Specification Version 1.0 (CAN with Flexible Data-Rate Specification Version 1.0).

The 2KB message RAM memory can realize the filter (Rx Filter), Rx FIFO (Rx FIFO), Rx Buffer (Rx Buffer), Tx Event FIFO (Tx Event FIFO), and Tx Buffer (Tx Buffer) functions. The message RAM is shared between the MCAN1 and MCAN2 modules.

30.2 Main Features

- Applicable to ISO 11898-1:2015 standard
- CAN FD supports sending and receiving up to 64 data bytes
- Support CAN error log
- Support AUTOSAR and SAE J1939
- Support receiving filtering function
- Two configurable receiving FIFOs
- A separate signal is sent when a high priority message is received
- Up to 64 dedicated Rx Buffer s
- Up to 32 dedicated Tx Buffers
- Configurable Tx FIFO or queue
- Configurable Tx Event FIFO
- The master CPU can directly access the message RAM
- The two modules MCAN1 and MCAN2 share the same message RAM
- Support programmable loopback test mode
- Support maskable module interrupts
- Two clock domains: CAN core clock and AHB bus interface clock

30.3Functional Description

30.3.1 MCAN System Block Diagram

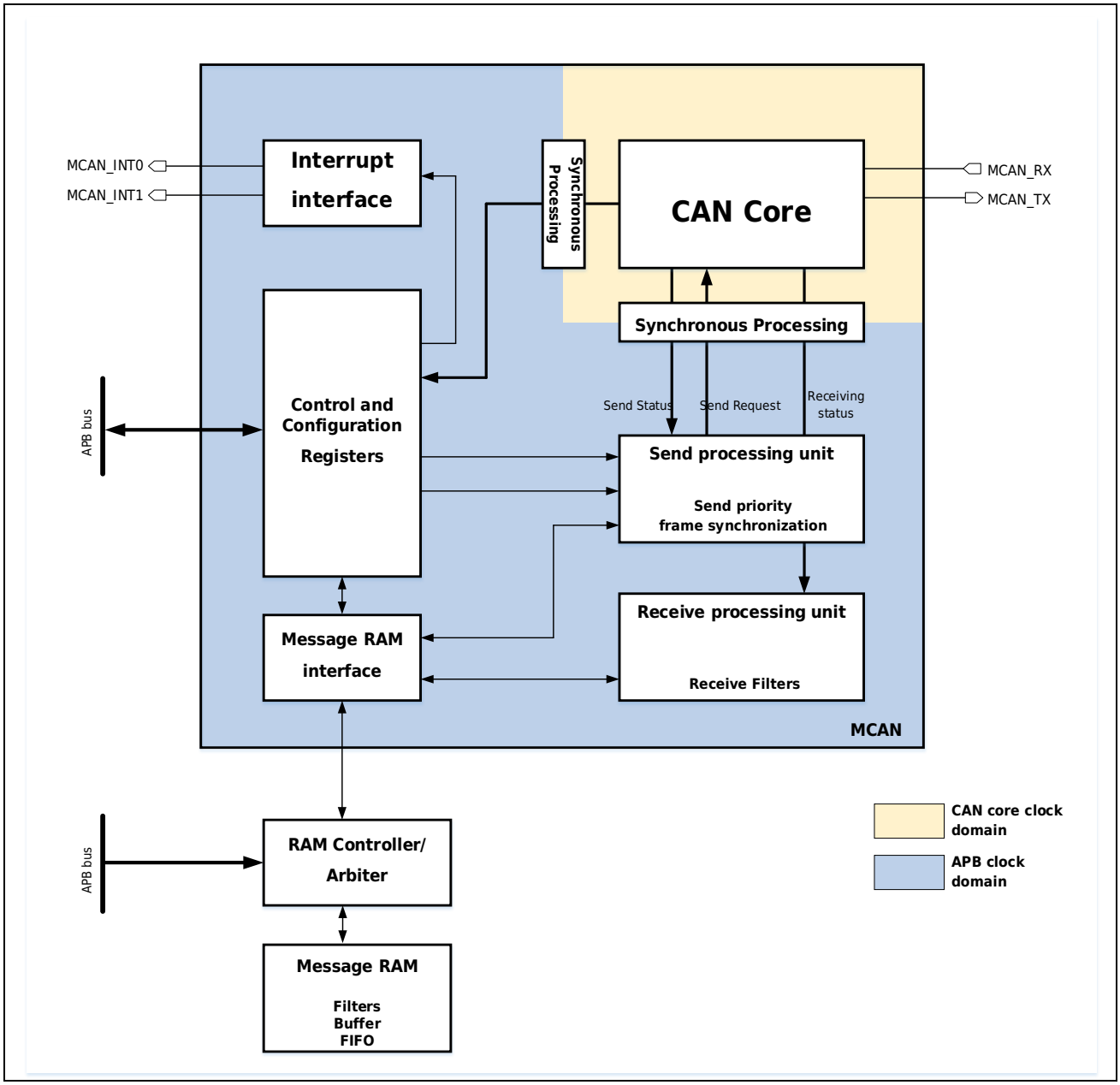


Figure 30-1 MCAN System Block Diagram

CAN Core

The CAN core includes a protocol controller, receive and send shift registers. It handles all protocol functions of ISO 11898-1:2015 and supports 11-bit standard ID and 29-bit extended ID.

Synchronize

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 clock of the AHB clock domain is the MCAN control logic clock, and the clock of the CAN core clock domain is the MCAN communication clock. Please refer to section [Working Clock Specification]. In order to ensure the stability of MCAN communication, the control logic clock frequency must be greater than or equal to the communication clock frequency.

Send Processing Unit

This processing unit controls the transfer of messages from the external message RAM to the CAN core. A maximum of 32 Tx Buffers can be configured for sending. The Tx Buffer can be used as a dedicated Tx Buffer, a Tx FIFO (or queue), or a combination of both. The Tx Event FIFO stores the send timestamp together with the corresponding message ID. Cancellation of sending is also supported.

Receive Processing Unit

This processing unit controls the transfer of received messages from the CAN core to the external message RAM. The receive processing unit supports two Rx FIFOs (the size of each FIFO is configurable) and up to 64 dedicated Rx Buffer s for storing all messages that pass the receive filter. Different from the Rx FIFO, the dedicated Rx Buffer is only used to store messages with a specific ID. The reception timestamp is storedwith its message. For the 11-bit standard ID, up to 128 filters can be defined; for the 29-bit extended ID, up to 64 filters can be defined.

AHB Interface

Connect the MCAN to the AHB bus.

Message RAM

The interface connects MCAN access to an external 2KB message RAM via a RAM controller and arbiter.

Double Interrupt Line

MCAN provides two interrupt lines, MCAN_INT0 and MCAN_INT1. Programming the MCAN_ILE.EINT0 and MCAN_ILE.EINT1 bits enables or disables the two interrupt lines, respectively.

30.3.2 Pin Description

Table 30-1 MCAN Pin Description

Pin Name	Direction	Functional Description
MCAN_RX	Input	MCAN receive pin
MCAN_TX	Output	MCAN send pin

30.3.3 Operating Mode

30.3.3.1 Software Initialization

Software initialization is started by setting MCAN_CCCR.INIT. Software reset, hardware reset or entering Bus_Off will set MCAN_CCCR.INIT. 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 register is shown in Table 30-2.

Table 30-2 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 send 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 Rx FIFO0 Configuration Register
MCAN_RXBC	MCAN Rx Buffer Configuration Register
MCAN_RXF1C	MCAN Rx FIFO1 Configuration Register
MCAN_RXESC	MCAN Rx Buffer and FIFO Element Size Configuration Registers
MCAN_TXBC	MCAN Tx Buffer Configuration Register
MCAN_TXESC	MCAN Tx Buffer Element Size Configuration Register
MCAN_TXEFC	MCAN Tx 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——Rx FIFO 0 status
- MCAN_RXF1S——Rx FIFO 1 status
- MCAN_TXFQS——Tx FIFO/Queue Status

- MCAN_TXBRP——Tx Buffer Request Pending
- MCAN_TXBTO——Tx Buffer send Occurs
- MCAN_TXBCF——Tx Buffer Cancel Complete
- MCAN_TXEFS——Tx Event FIFO Status
- MCAN_TXBAR——Tx Buffer add request
- MCAN_TXBCR——Tx Buffer Cancel Request

The master can set MCAN_CCCR.TEST and MCAN_CCCR.MON only when MCAN_CCCR.INIT and MCAN_CCCR.CCE are both set. The master 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.

30.3.3.2 Normal Operation

After the MCAN has completed initialization and MCAN_CCCR.INIT is cleared, the MCAN will synchronize itself with the CAN bus and be ready for communication.

When receiving messages, the messages (including message ID and DLC) received by the receive filter are stored in a dedicated Rx Buffer , or Rx FIFO0, or Rx FIFO1.

When sending a message, first initialize or update the dedicated Tx Buffer and/or Tx FIFO (queue). Automatic transmission after receiving a remote frame is not supported.

30.3.3.3 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 disable (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 CAN FD frames or Classic CAN frames are sent can be configured via the FDF bit in the corresponding Tx Buffer element. If MCAN_CCCR.FDOE=0, received frames are treated as Classic CAN frames and thus an Error Frame is sent when a CAN FD frame is received. If CAN FD mode disable, no CAN FD frame will be sent even if the FDF bit of the Tx Buffer element is set. MCAN_CCCR.FDOE and MCAN_CCCR.BRSE can be changed only when both MCAN_CCCR.INIT and MCAN_CCCR.CCE are set.

If MCAN_CCCR.FDOE=0, the settings of the FDF and BRS bits are ignored and the frame is sent in Classic CAN format. If MCAN_CCCR.FDOE = 1 and MCAN_CCCR.BRSE = 0, only the FDF bit of the Tx Buffer element is evaluated. If MCAN_CCCR.FDOE = 1 and MCAN_CCCR.BRSE = 1, bit rate switching is enabled for sending CAN FD frames. All Tx Buffer elements with the FDF and BRS bits set are sent in the CAN FD format with bit rate switched.

It is recommended to switch modes during CAN operation only when the following conditions are met:

- 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, the sent CAN FD bit rate switching should be disabled.
- During system startup, all nodes will send Classical CAN messages before being verified as being able to communicate in the CAN FD format. If verified, all nodes switch to CAN FD operation.
- Wake-up messages in CAN Partial Networking must be sent in the Classic 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. Afterwards, all nodes switch back to classic CAN communication.

In the CAN FD format, the encoding of DLC codes 0 to 8 (corresponding data byte length) is the same as that of classic CAN, and the encoding of codes 9 to 15 is different from that of classic CAN, as shown in Table 30-3.

Table 30-3 DLC Encoding in MCAN

DLC Code		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
data Number of bytes	Classic 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 a CAN FD frame, if the BRS bit is recessive, the bit duration will be switched within the frame after the BRS (Bit Rate Switch) bit. Before the BRS bit, in the CAN FD arbitration phase, the bit timing is defined in the register MCAN_NBTP (Nominal Bit Timing & Prescaler Register). In the following CAN FD data phase, the bit timing is defined in the register MCAN_DBTP (Data Bit Timing

& Prescaler Register). The bit duration switches back from the data phase bit duration at the CRC delimiter or when an error is detected, whichever occurs first.

The maximum configurable bit rate of the CAN FD data phase depends on the CAN communication clock frequency. For example, if the CAN communication clock frequency is 20MHz and the shortest configurable bit time is 4 time slices (tq), the bit rate of the data phase is 5Mbps.

In both CAN FD data frame formats with and without bit rate switching, the value of the ESI (Error Status Indicator) bit is determined by the error status of the sending node at the start of transmission. If the sending node is in error passive state, ESI is sent as a recessive bit, otherwise it is sent as a dominant bit.

30.3.3.4 Send Delay Compensation

In the data phase of CAN FD transmission, only one node is sending and all other nodes are receiving nodes. The length of the bus has no effect. When sendting via pin MCAN_TX, the MCAN receives the sent data from its local CAN transceiver via pin MCAN_RX. The received data is delayed by the send delay, and if this delay is greater than TSEG1 (the time period before the sampling point), a bit error is detected. In order to make the data phase bit time shorter than the transmission delay, delay compensation is introduced. Without send delay compensation, the bit rate of the CAN FD frame data phase will be limited by the send delay.

The MCAN protocol unit implements a delay compensation mechanism to compensate for the transmission delay, thereby enabling transmission at a higher bit rate in the CAN FD data phase without being affected by the delay of the specific CAN transceiver.

In order to check whether there is a bit error in the data phase of the sending node, the delayed send data is compared with the received data at the Secondary Sample Point SSP. If a bit error is detected, the sending node will react to the bit error at the next regular sampling point. During the arbitration phase, latency compensation is always disabled.

send delay compensation supports configurations where the data bit time is shorter than the send delay, as described in detail in ISO 11898-1:2015 and is enabled by setting MCAN_DBTP.TDC.

At the Secondary Sample Point SSP compares the received bit with the sent bit. The position of the SSP is defined as the sum of the delay measured from the MCAN send output pin MCAN_TX through the transceiver to the receive input pin MCAN_RX, and the send delay compensation offset configured by MCAN_TDCR.TDCO. The send delay compensation offset is used to adjust the position of the SSP in the received bits (e.g., half of the bit time in the data phase). The position of the Secondary Sample Point is rounded down to the next integer mtq (Minimum time quantum, which is one MCAN communication clock cycle).

MCAN_PSR.TDCV shows the actual send delay compensation value. When MCAN_CCCR.INIT is set, MCAN_PSR.TDCV is cleared; when MCAN_DBTP.TDC is set, MCAN_PSR.TDCV is updated every time an FD frame is sent.

For the send delay compensation implemented in MCAN, the following boundary conditions must be considered:

- The sum of the measured MCAN_TX to MCAN_RX delay and the configured send delay compensation offset MCAN_TDCR.TDCO must be less than 6 bit times of the data phase.
- The sum of the measured MCAN_TX to MCAN_RX delay and the configured send delay compensation offset MCAN_TDCR.TDCO must be less than or equal to 127mTQ. If this sum is greater than 127 MTQ, the maximum value of 127 MTQ is used for send delay compensation.
- The data phase ends at the sampling point of the CRC delimiter, after which the received bits are no longer checked at the SSP.

If send delay compensation is enabled by programming MCAN_DBTP.TDC = 1, the measurement starts at the falling edge of the FDF bit to the res bit of each CAN FD frame. When this edge is detected on the MCAN_RX pin of the sendting node, the measurement stops. The resolution of this measurement is one mtq.

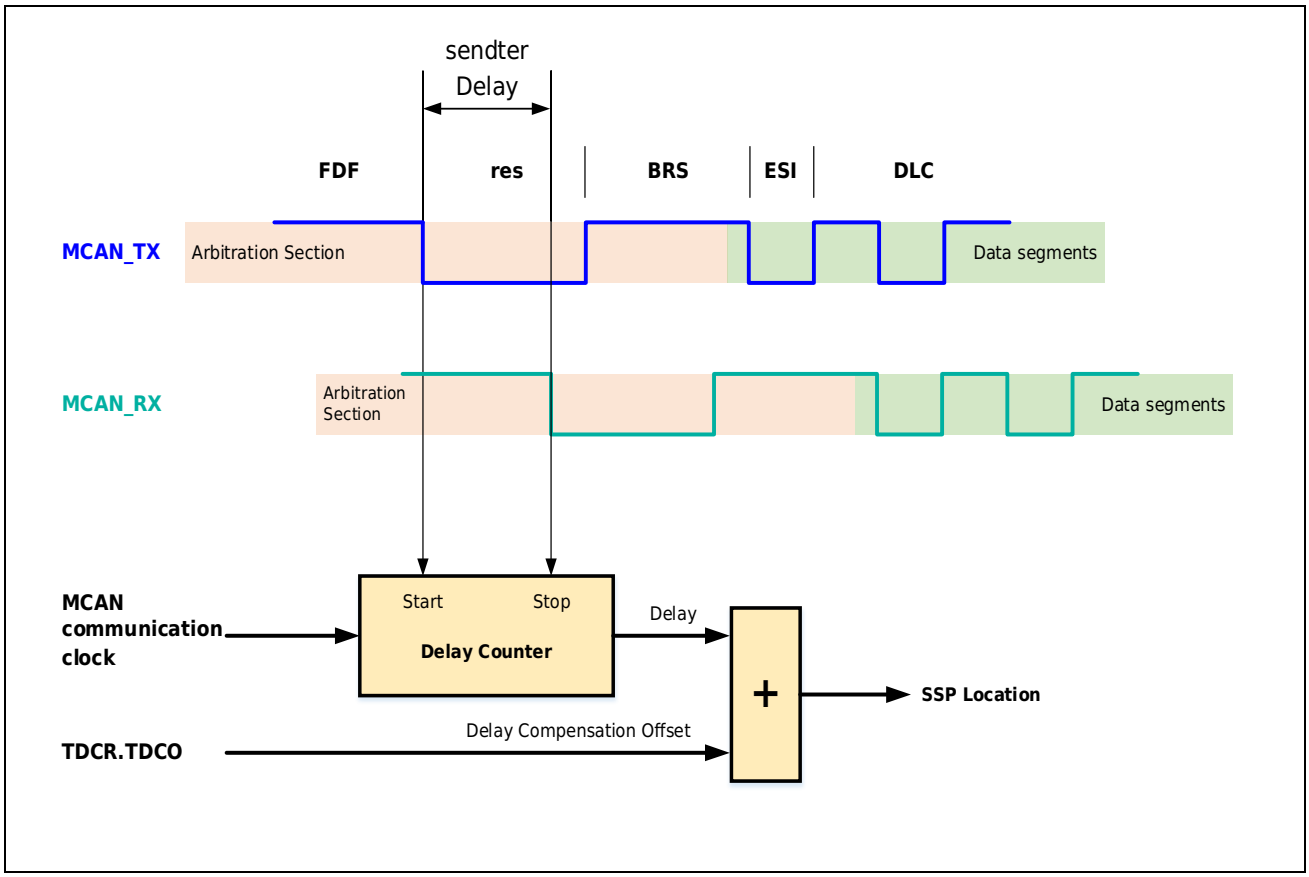


Figure 30-2 Transceiver Delay Measurement

To avoid dominant glitches in the received FDF bit, ending the delay compensation measurement before the falling edge of the received res bit, resulting in an early SSP position, the send delay compensation filter window can be enabled by programming MCAN_TDCR.TDCF to define the minimum value of the SSP position. A dominant edge on the MCAN_RX pin will cause the SSP position to be further advanced, but this will be ignored for the send delay measurement. The measurement stops when the SSP position is at least MCAN_TDCR.TDCF and the MCAN_RX pin is low.

Note:

The send delay used by MCAN is a real-time measurement value and does not support configuring a fixed send delay.

30.3.3.5 Restricted Working Mode

In restricted operating mode, the node can receive data frames and remote frames and acknowledge valid frames, but will not send data frames, remote frames, active error frames, or overload frames. In the event of an error or overload condition, no dominant bit is sent, but instead a bus idle condition is waited for to occur in order to resynchronize to the CAN communication. When the error record (MCAN_TDCR.CEL) is activated, the error counters (MCAN_TDCR.REC, MCAN_TDCR.TEC) are frozen. The master can set the MCAN to restricted operating mode by setting MCAN_CCCR.ASM. The master can set MCAN_CCCR.ASM only when both MCAN_CCCR.CCE and MCAN_CCCR.INIT are set to 1. MCAN_CCCR.ASM can be cleared by the master at any time.

When the sending processing unit cannot read data from the message RAM in time, it will automatically enter the restricted working mode. To exit restricted operating mode, the master CPU must clear MCAN_CCCR.ASM.

Restricted operating modes can be used to accommodate applications with different CAN bit rates. In this case, the application tests different bit rates and exits the restricted operation mode after receiving a valid frame.

Note:

Restricted operating mode must not be used in conjunction with loopback mode (internal or external).

30.3.3.6 Bus Monitoring Mode

Configure MCAN_CCCR.MON to 1 and set MCAN to bus monitoring mode. In bus monitoring mode (see ISO 11898-1:2015, 10.14 Bus monitoring), the MCAN can receive valid data frames and valid remote frames, but cannot initiate transmission. In this mode, MCAN only sends recessive bits on the CAN bus. If MCAN needs to send a dominant bit (ACK bit, overload flag, active error flag), the bit will be redirected internally so that MCAN can listen to the dominant bit, but the CAN bus can

remain in the recessive state. In bus listening mode, the register MCAN_TXBRP remains in reset state.

The bus listening mode can be used to analyze the traffic on the CAN bus without affecting the bus by sending dominant bits. Figure 30-3 shows the connection of MCAN_TX and MCAN_RX signals to MCAN in bus listening mode.

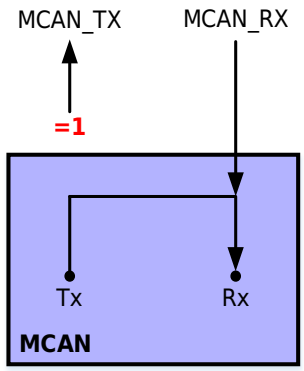


Figure 30-3 Pin control in bus monitoring mode

30.3.3.7 Disable Automatic Retransmission (DAR) Mode

According to the CAN specification (see ISO 11898-1:2015, 8.3.4 Recovery Management), the MCAN can automatically resend frames that have failed arbitration or were disturbed by errors during transmission. Automatic resend is enabled by default.

In DAR mode, all transmissions started on the CAN bus are automatically cancelled. The Tx Buffer send request pending bit MCAN_TXBRP.TRPx is reset after a successful transmission. It is also reset if the transmission has not started when canceled, the transmission has been aborted due to arbitration failure, or an error occurs during frame transmission.

- Successfully sent:
  - The corresponding Tx Buffer send occurred bit MCAN_TXBTO.TOx is set
  - The corresponding Tx Buffer cancel completed bit MCAN_TXBCF.CFx is not set
- Successfully sent even though it was cancelled:
  - The corresponding Tx Buffer send occurred bit MCAN_TXBTO.TOx is set
  - The corresponding Tx Buffer cancel completed bit MCAN_TXBCF.CFx is set
- Arbitration failed or frame transmission was disturbed:
  - The corresponding Tx Buffer send occurred bit MCAN_TXBTO.TOx is not set
  - The corresponding Tx Buffer cancel completed bit MCAN_TXBCF.CFx is set

If the frame is sent successfully, and storage of Tx Events is enabled, the Tx Event FIFO element is written with event type ET = 0b10 (send occurs even if canceled).

30.3.3.8 Power Down (Sleep Mode)

The MCAN can be set to power-down mode via the CC control register MCAN_CCCR.CSR. The MCAN_CCCR.CSR bit reads 1 as long as the clock stop request valid.

When all pending send requests are completed, the MCAN waits until the bus idle state is detected. After detecting the bus idle state, the MCAN sets MCAN_CCCR.INIT to 1 to prevent any further CAN transmission. The MCAN then confirms that it is ready to enter the Power-Down state by setting MCAN_CCCR.CSA to 1. Registers can continue to be accessed until the module clocks (control logic clock and communication clock) are turned off, but MCAN_CCCR.INIT remains at 1.

Note:

In case of severe interference on the CAN bus, the idle state may never be reached and therefore the MCAN will not set MCAN_CCCR.INIT. This condition can be detected by polling MCAN_PSR.ACT. If the MCAN does not enter the idle state, software can write CCCR.INIT = 1, which will immediately stop the CAN communication of the MCAN, regardless of whether transmission or reception is in progress.

To exit Power-down mode, the application must start the module clock before resetting the CC control register flag MCAN_CCCR.CSR. The MCAN will confirm that the clock is turned on by resetting MCAN_CCCR.CSA. Afterwards, the application can restart CAN communication by resetting MCAN_CCCR.INIT.

30.3.3.9 Test Pattern

To enable write access to the MCAN test register MCAN_TEST (see chapter [MCAN Test Register (MCAN_TEST)]), MCAN_CCCR.TEST must be set to 1 so that the test mode and test functions can be configured.

By programming MCAN_TEST.TX, the send pin MCAN_TX can realize four output functions. In addition to the default function of serial data output, the CAN sample point signal can also be driven to monitor the bit duration of the MCAN and can output a constant dominant or recessive level. The actual value of pin MCAN_RX can be read from MCAN_TEST.RX. Both 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 cycles from configuring MCAN_TEST.TX to pin MCAN_TX implementing the specified function. This also applies when reading pin MCAN_RX via MCAN_TEST.RX.

Note:

The test mode is intended for production testing and self-checking only. Software control of the MCAN_TX pin interferes with all CAN protocol functionality. It is not recommended to use the test mode for actual applications.

30.3.3.10 External Loopback Mode

Configure MCAN_TEST.LBCK to 1 and set MCAN to external loopback mode. In loopback mode, the MCAN processes its own sent messages as received messages and stores the messages (if they pass the receive filter) in the Rx Buffer or Rx FIFO. Figure 30-4 shows the connection of MCAN_TX and MCAN_RX pin signals to MCAN in external loopback mode.

This mode is used for hardware self-test. In order to avoid external influences, MCAN ignores ACK errors (hidden bits are sampled in the ACK field of data frames or remote frames) in loopback mode. 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 messages can be monitored from the pin MCAN_TX.

30.3.3.11 Internal Loopback Mode

Configure MCAN_TEST.LBCK and MCAN_CCCR.MON to 1 to set the MCAN to internal loopback mode. This mode can be used for "Hot Selftest", that is, the MCAN can perform tests without affecting the connected and running CAN system. In this mode, pin MCAN_RX is disconnected from MCAN and pin MCAN_TX remains recessive. Figure 30-4 shows the connection of MCAN_TX and MCAN_RX pin signals to the MCAN in internal loopback mode.

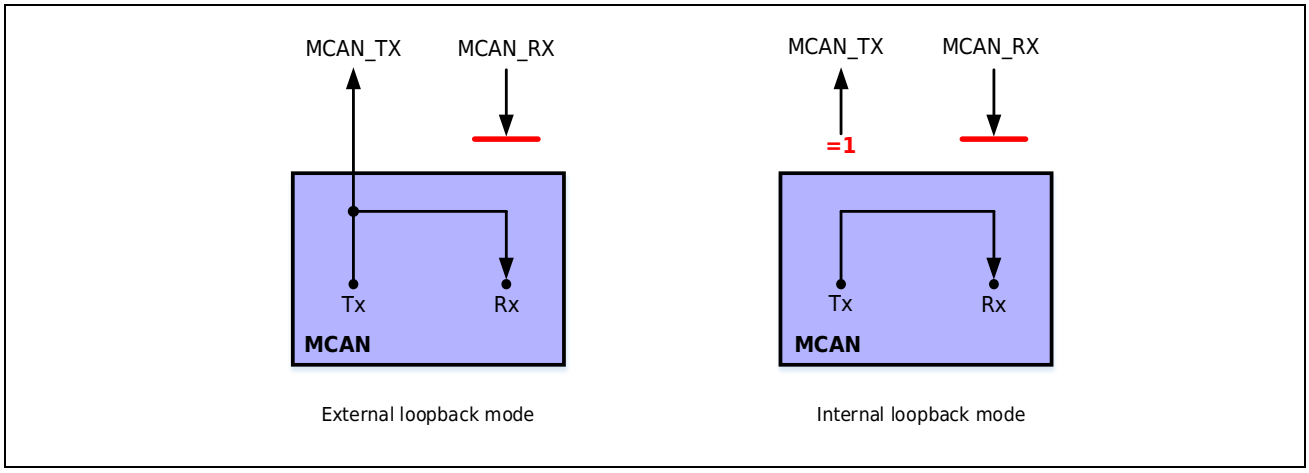


Figure 30-4 Pin Control of Loopback Mode

30.3.4 Timestamp Generation

The MCAN provides a 16-bit cycle counter to generate internal timestamps. The counting time unit can be configured to 1~16 times of CAN bit time through the prescaler MCAN_TSCC.TCP. The counter value can be read through MCAN_TSCV.TCV. A write access to register MCAN_TSCV resets the counter to 0. When the timestamp counter cycles, the interrupt flag MCAN_IR.TSW is set.

When a SOF (Start Of Frame) is received or sent, the count value is captured and stored in the timestamp portion of the Rx Buffer or Rx FIFO (RXTS[15:0]) or Tx Event FIFO (TXTS[15:0]) element.

30.3.5 Timeout Counter

The MCAN provides a 16-bit time-out counter to indicate time-out conditions for Rx FIFO0, Rx FIFO1, and Tx Event FIFO. It works as a down counter and, like the timestamp counter, uses a prescaler controlled by MCAN_TSCC.TCP. The time-out 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. When MCAN_CCCR.INIT=1, the timeout counter stops counting, for example when CAN enters Bus_Off state.

The working mode is selected by MCAN_TOCC.TOS. When operating in continuous mode, the counter starts when MCAN_CCCR.INIT is reset. Writing to MCAN_TOCV will preset the counter to the value configured by MCAN_TOCC.TOP and continue counting down.

When the timeout counter is controlled by one of the FIFOs, an null FIFO will preset the counter to the value configured by MCAN_TOCC.TOP. After storing the first FIFO element, 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 of the timeout counter comes from the CAN core sampling point signal. Therefore, due to the CAN core synchronization (resynchronization) mechanism, the time point at which the timeout counter is decremented may vary. If the bit rate switching feature is used in CAN FD, the timeout counter counts using different clocks in the Arbitration phase and the Data phase.

30.3.6 Receiving Processing

The receive processing unit controls receive filtering, transfer of received messages to the Rx Buffer or one of the two Rx FIFOs, and Rx FIFO put and get indexes.

30.3.6.1 Receiving Filters

MCAN can configure two sets of acceptance filters, one for standard ID and the other for extended ID. These two sets of filters can be assigned to the Rx Buffer , Rx FIFO0 or Rx FIFO1. For receiving filtering, each filter list is compared starting from the 0th element until the first matching element is found. Receiving 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 acceptance filtering or rejection filtering
  - Each filter element can be enabled/disabled individually
  - Compare filter elements in order, stopping execution after the first matching filter element
- Relevant 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 into Rx FIFO0 or FIFO1
  - Store received frames into the Rx Buffer
  - Reject received frames
  - 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 into Rx 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 (such as a CRC error), it discards the message with the following effects:

- Rx Buffer

  The new data flag for a matching Rx Buffer will not be set, but the Rx Buffer will be (partially) overwritten by the received data. See MCAN_PSR.LEC and MCAN_PSR.DLEC for error types.
- Rx FIFO

  The put index of the matching Rx FIFO is not updated, but the associated Rx FIFO element (part) is overwritten by the received data. See MCAN_PSR.LEC and MCAN_PSR.DLEC for error types. If the matched Rx FIFO operates in overwrite mode, you must consider the boundary conditions described in Rx FIFO Overwrite Mode.

Note:

When received messages are written to one of the two Rx FIFOs or to the Rx Buffer , unmatched received IDs are stored independently of the used filter. The process of receiving filtering depends heavily on the order in which the filter elements are configured.

30.3.6.2 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

30.3.6.3 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.

30.3.6.4 ID Bit Mask Filter

ID bit mask filtering is used to filter message ID groups by masking certain bits of the received message ID. When performing ID bit mask filtering, SF1ID/EF1ID is used as the message ID filter and SF2ID/EF2ID is used as the filter ID mask (ID Mask).

The bits that are 0 in the ID mask will mask out the corresponding bits of the configured ID filter, that is, the value of the received message ID at this bit position is not compared with the value of the corresponding bit of the reception filter, only the bits of the received message ID whose corresponding mask bit is 1 are compared.

If all mask bits are 1, a match will occur only if the received message ID and the message ID filter are exactly the same. If all mask bits are 0, then all message IDs match.

30.3.6.5 Standard Message ID Filtering

This Figure 30-5 describes the filtering process for standard message IDs (11-digit IDs). Section [Standard Message ID Filter Elements] introduces the standard message ID filter elements.

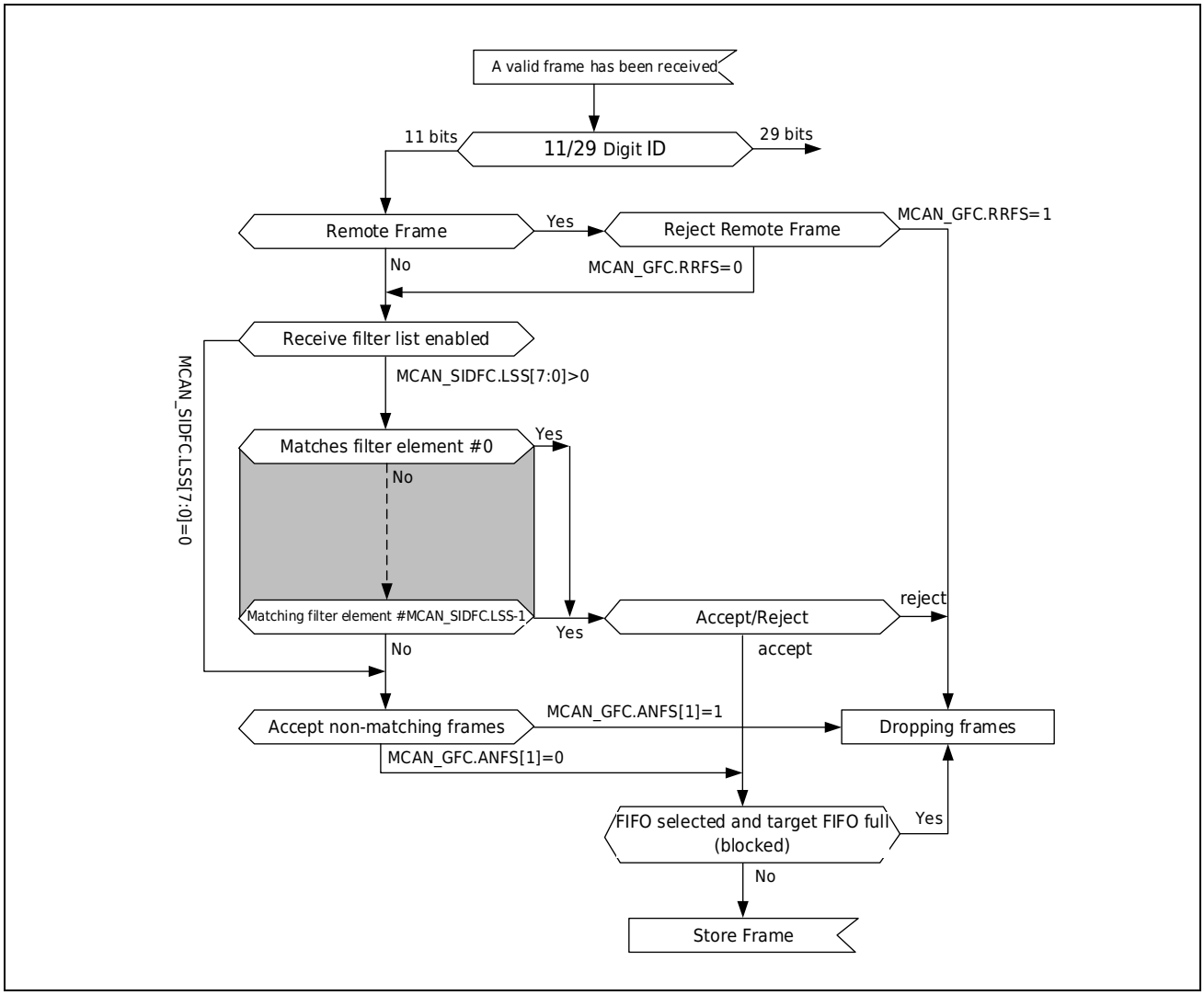


Figure 30-5 Standard Message ID Filter Path

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.

30.3.6.6 Extended Message ID Filtering

Figure 30-6 describes the filtering process for extended message IDs (29-bit IDs). Section [Extended Message ID Filter Element] introduces the extended message ID filter element.

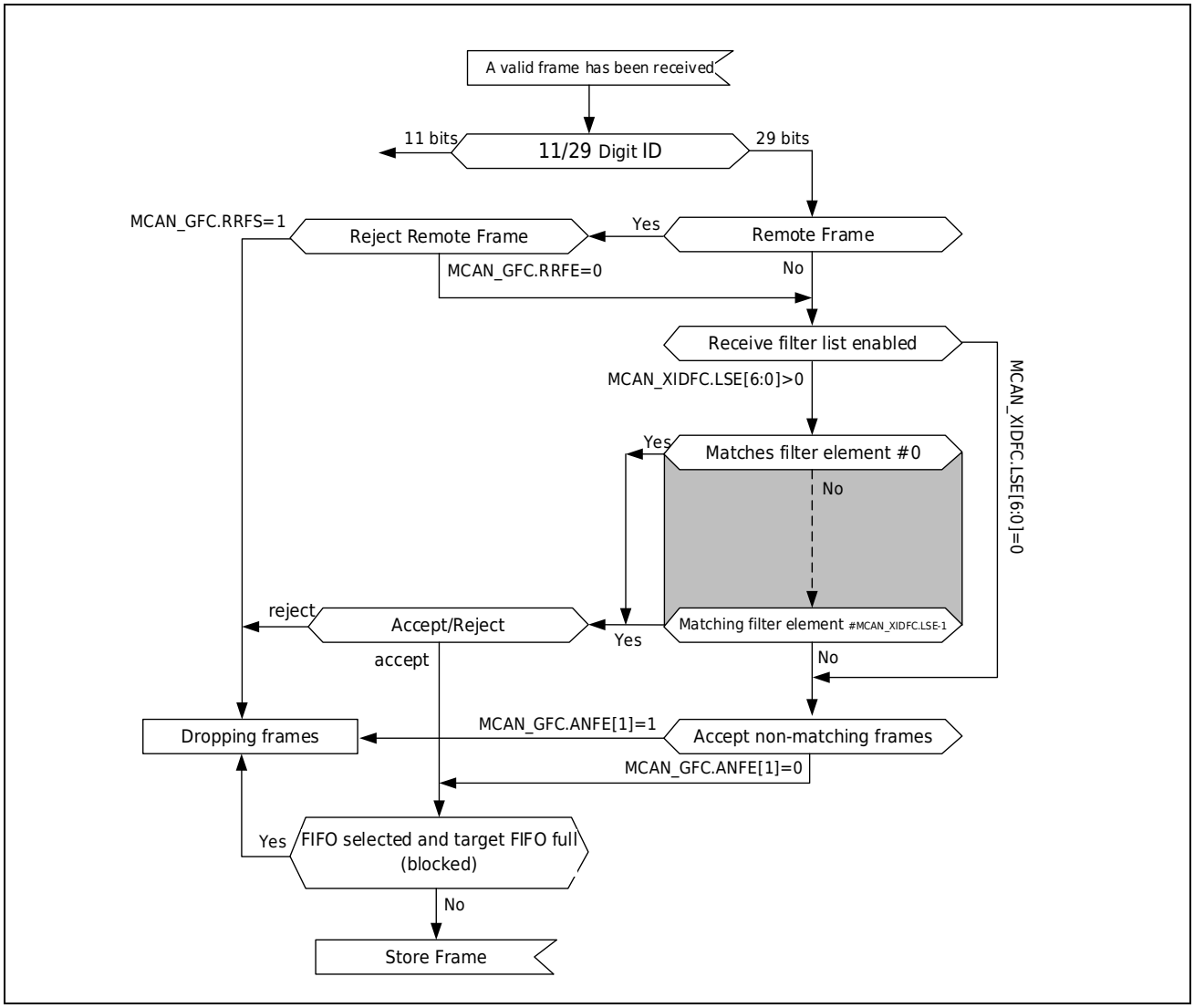


Figure 30-6 Extended Message ID Filter Path

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 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.

30.3.6.7 Rx FIFO

Rx FIFO0 and Rx 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 Rx FIFOs is done through registers MCAN_RXF0C and MCAN_RXF1C.

Messages that pass the receive filter are transferred to the Rx FIFO configured by the matching filter element. For a description of the filter mechanism that can be used for Rx FIFO0 and Rx FIFO1, see section [Receiving Filters]. Chapter [Rx Buffer and FIFO Elements] introduces the Rx Buffer and FIFO elements.

To prevent the Rx FIFO from overflowing, the Rx FIFO watermark can be used. When the Rx FIFO fill level reaches the Rx FIFO watermark configured by MCAN_RXFnC.FnWM, the interrupt flag MCAN_IR.RFnW is set. When the put index and get index of the Rx FIFO meet, the Rx FIFO full condition occurs, the flag MCAN_RXFnS.FnF is set, and the interrupt flag MCAN_IR.RFnF is set.

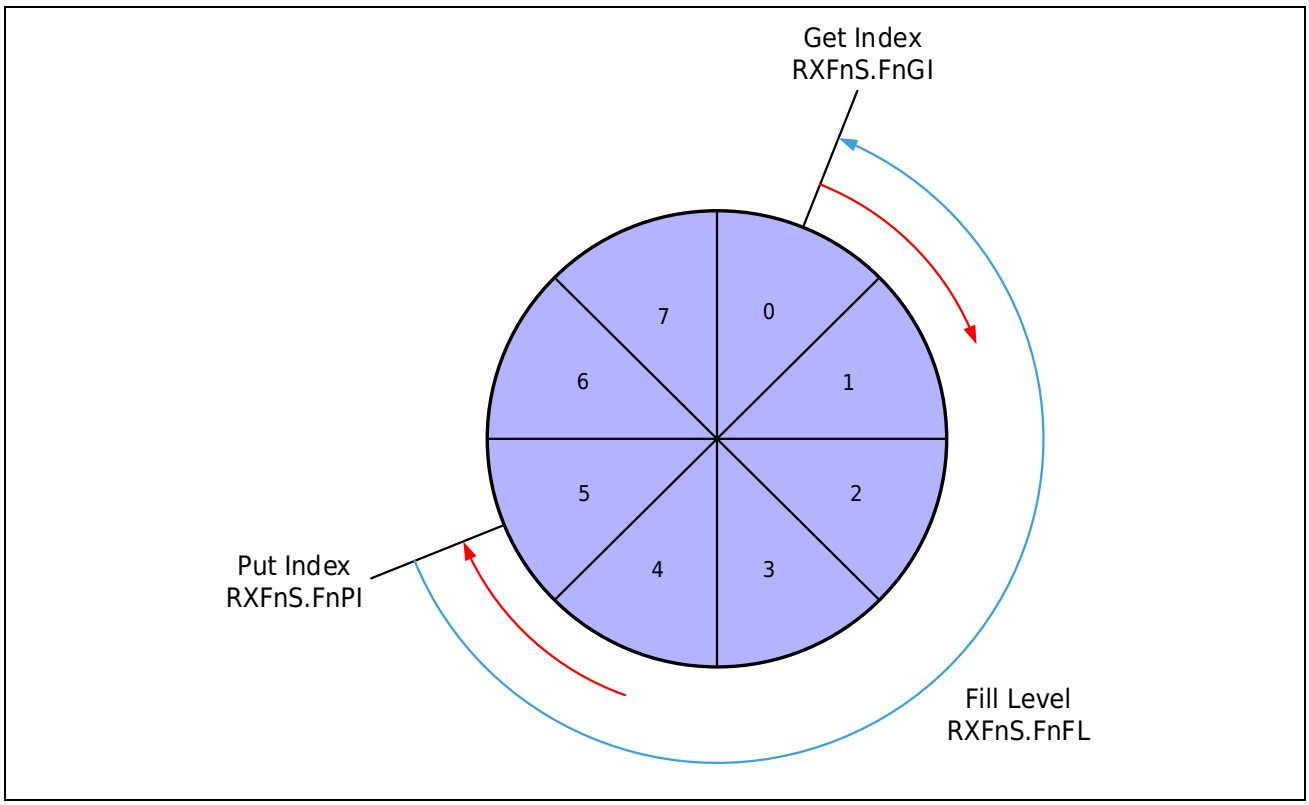


Figure 30-7 Rx FIFO Status

When reading a message from the Rx FIFO, the Rx 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 30-4 Rx Buffer and FIFO Element Size

RXESC.RBDS[2:0] RXESC.FnDS[2:0]	Data Field (Bytes)	Element Size (Words)
000	8	4
001	12	5
010	16	6
011	20	7
100	24	8
101	32	10
110	48	14
111	64	18

30.3.6.8 Rx FIFO Blocking Mode

The Rx FIFO blocking mode is configured by MCAN_RXFnC.FnOM=0, which is the default operating mode of the Rx FIFO.

When the Rx FIFO is full (MCAN_RXFnS.FnPI=MCAN_RXFnS.FnGI), the flag bit RXFnS.FnF=1 and the interrupt flag MCAN_IR.RFnF=1. At this point, the MCAN will not continue to write messages to the corresponding Rx FIFO unless messages are read from the corresponding Rx FIFO and its fetch index has been incremented.

If a message is received when the corresponding Rx FIFO is full, the message will be discarded and the message loss will be indicated by MCAN_RXFnS.RFnL=1. In addition, the interrupt flag MCAN_IR.RFnL is also set.

30.3.6.9 Rx FIFO Overwrite Mode

The Rx FIFO overwrite mode is configured by MCAN_RXFnC.FnOM=1.

When the Rx FIFO is full (MCAN_RXFnS.FnPI=MCAN_RXFnS.FnGI), the flag bit RXFnS.FnF=1. At this point, the next message received by the FIFO will overwrite the earliest received message, and both the put index and the get index will increase by 1.

If the Rx FIFO operates in overwrite mode and the Rx FIFO is full, the Rx FIFO elements should be read starting at least from the fetch index plus 1. This is because when the CPU is reading messages from the message RAM, a received message happens to be 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 acquisition index when reading data from the Rx FIFO can avoid this problem. The offset depends on how fast the CPU can access the Rx FIFO. Figure 30-8 shows that the offset of getting index when reading the Rx FIFO is 2. In this case, the two messages stored in elements 1 and 2 are lost.

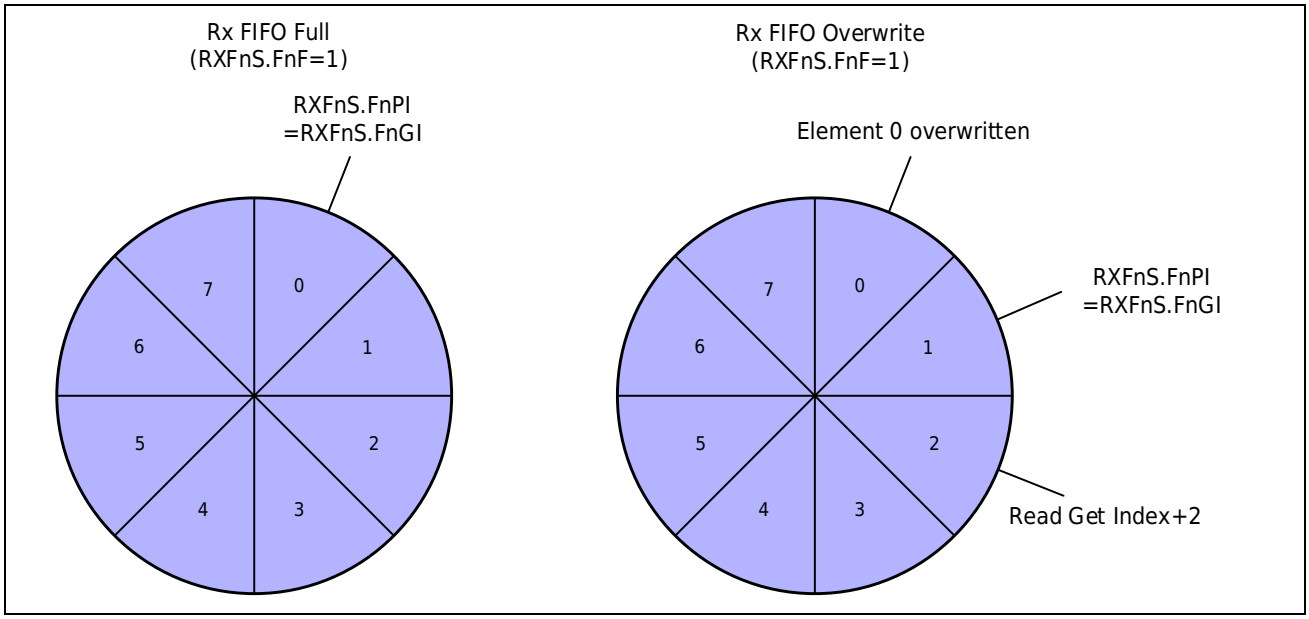


Figure 30-8 Rx FIFO Overflow Processing

After reading data from the Rx FIFO, the number of the last element read must be written into the Rx 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 Rx FIFO element number, the Rx FIFO full flag will be reset (MCAN_RXFnS.FnF = 0).

30.3.6.10 Dedicated Rx Buffer

The MCAN supports up to 64 dedicated Rx Buffer s. The starting address of the dedicated Rx Buffer is configured via RXBC.RBSA.

To use a dedicated Rx Buffer , for standard ID messages, the SFEC bit of the standard message ID filter element must be configured to 0b111 and the SFID2 bit to 0b00 (see section [Standard Message ID Filter Elements]); for extended ID messages, the EFEC bit of the extended message ID filter element must be configured to 0b111 and the EFID2 bit to 0b00 (see section [Extended Message ID Filter Element]). Table 30-5 Example of filter configuration for a dedicated Rx Buffer .

After a received message is accepted by a filter element, it is stored in the Rx Buffer in the message RAM referenced by the filter element. The storage format is the same as that of the Rx FIFO elements. Additionally, the flag MCAN_IR.DRX in the interrupt register (message stored in the dedicated Rx Buffer) is set.

Table 30-5 Rx 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	00	00 0000
1	ID Message 2	00	00 0001
2	ID Message 3	00	00 0010

After the last word of the matched received message is written into the message RAM, the corresponding new data flag in register MCAN_NDAT1/MCAN_NDAT2 will be set. As long as the new data flag remains set, the corresponding Rx Buffer is locked to prevent it from being updated with new matching messages. The new data flag must be reset by the master writing 1 to the corresponding bit.

When the new data flag of the Rx Buffer is set, the 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 Rx Buffer or Rx FIFO, or to be rejected, depending on the filter configuration.

30.3.6.11 Rx Buffer Processing

- Reset interrupt flag MCAN_IR.DRX
- Read new data register
- Reading messages from the message RAM
- Reset the new data flag of the processed message

30.3.6.12 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, the meanings of SFID1/SFID2 and EFID1/EFID2 are different (see Section [Standard Message ID Filter Elements] and Section [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 received debug messages. Table 30-6 Example configuration for debug message filter.

When storing debug messages, the corresponding new data flag and MCAN_IR.DRX will not be set, and the reception of debug messages can be monitored through MCAN_RXF1S.DMS.

Table 30-6 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	01	11 1101
1	ID Debug Message B	10	11 1110
2	ID Debug Message C	11	11 1111

30.3.7 Send Processing

The send processing unit processes the send requests of the dedicated Tx Buffer, Tx FIFO and send queue. It controls the transmission of send messages to the CAN core, puts and gets indexes into and out of the Tx Event FIFO. Up to 32 Tx Buffers can be set for message sending, and each Tx Buffer can be independently configured with the send mode (classic CAN mode or CAN FD mode).

For a description of the Tx Buffer, see section [Tx Buffer Elements]. Table 30-7 Describes the possible configurations for sending frames.

Table 30-7 Possible Configurations for Sending Frames

MCAN_CCCR		Tx Buffer Elements		Send Frame
BRSE	FDOE	FDF	BRS	
Neglect	0	Neglect	Neglect	Classic CAN
0	1	0	Neglect	Classic CAN
0	1	1	Neglect	FD without bit rate switching
1	1	0	Neglect	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 send queue buffers and supports send cancellation.

When the Tx Buffer request pending register MCAN_TXBRP is updated, the send processing unit starts a send scan to check the highest priority pending send request (the Tx Buffer with the smallest message ID). The conditions for MCAN_TXBRP to be updated are that a Tx Buffer is sent completely, or the master adds a send request by writing MCAN_TXBAR, or the master cancels a pending transmission by writing MCAN_TXBCR.

30.3.7.1 Send Pause

The Send Pause function is used in CAN systems where the 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, while in certain applications their relative arbitration priorities should be reversed. This may lead to the following situation: one ECU bursts out multiple CAN messages with higher arbitration priority, causing another ECU CAN message with 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 bus-free CAN bit times before it is allowed to start sending the next requested message. If other ECUs have pending messages, they will start sending those messages during their idle time and do not need to arbitrate with the next message from ECU-1. After receiving the message, ECU-1 can start the next transmission immediately after the received message releases the CAN bus.

This function is controlled by MCAN_CCCR.TXP. If this bit is set, after each successful message transmission, the MCAN will pause for two CAN bit times before starting the next transmission. This allows messages to be sent even if other nodes in the network have IDs with lower priority. This function disable by default (MCAN_CCCR.TXP=0). This feature allows bursts of multiple messages from a single node to be spread out more sparsely, preventing ECUs sending lower priority

messages from causing applications to incorrectly believe they are requesting too many transmissions.

30.3.7.2 Dedicated Tx Buffer

A dedicated Tx Buffer allows messages to be sent completely under the control of the master CPU. Each dedicated Tx Buffer is configured with a specific message ID. If multiple Tx Buffers are configured with the same message ID, the message in the Tx Buffer with the smallest number will be sent first. These Tx Buffers should be requested to be sent in ascending order, that is, the smallest buffer number should be requested to be sent first, or, request to send simultaneously through a single write to MCAN_TXBAR.

If there is an update to the message in the buffer, request to send it by setting MCAN_TXBAR.ARn. The message that has been requested to be sent will first be arbitrated internally with the messages in the Tx FIFO or send queue according to the message ID, and then arbitrated externally with the messages on the CAN bus, and finally sent out according to the arbitration result.

The dedicated Tx Buffer elements allocated in the Message RAM are a number of 32-bit words (4 bytes) in size. Therefore, the starting address of the dedicated Tx Buffer in the message RAM is calculated by multiplying the Tx Buffer index (0 to 31) by the element size (words), multiplied by 4 (converted to bytes), and then added to MCAN_TXBC.TBSA.

Table 30-8 Tx Buffer, FIFO (queue) Element Size

TXESC.TBDS2[2:0]	Data Field (Bytes)	Element Size (Words)
000	8	4
001	12	5
010	16	6
011	20	7
100	24	8
101	32	10
110	48	14
111	64	18

30.3.7.3 Tx FIFO

Tx FIFO mode is achieved by configuring MCAN_TXBC.TFQM to 0. The messages stored in the Tx FIFO are sent starting from the message pointed to by the get index MCAN_TXFQS.TFGI. After each transmission, the acquisition index will be incremented cyclically until the Tx FIFO is null. The Tx FIFO sends messages with the same ID from different Tx Buffers in the order in which the messages are written. MCAN obtains the number of available (free) Tx FIFO elements (TXFQS.TFFL) by calculating the difference between the get index and the put index.

New messages must be written into the Tx FIFO starting with the Tx Buffer pointed to by index MCAN_TXFQS.TFQPI. Adding a send request increments the put index to point to the next free Tx FIFO element. When the put index meets the get index, it indicates that the Tx FIFO is full (MCAN_TXFQS.TFQF=1). In this case, no further messages should be written to the Tx FIFO until the next message has been sent and the get index has been incremented.

If a single message is added to the Tx FIFO, it is requested to be sent by writing 1 to the MCAN_TXBAR bit corresponding to the index into which the Tx FIFO is placed.

If multiple (let's say n) messages are added to the Tx FIFO, they are written into n consecutive Tx Buffers starting with the put index. Then request to send through MCAN_TXBAR, and put it into the index loop to increase n. The number of Tx Buffers requested to be sent should not exceed the number of free Tx Buffers indicated by the Tx FIFO Free Level (MCAN_TXFQS.TFFL).

If the send request of the Tx Buffer pointed to by the get index has been canceled, the get index is incremented to the next Tx Buffer with a send request pending and the Tx FIFO free level (MCAN_TXFQS.TFFL) is recalculated. If a send from any other Tx Buffer is canceled, the fetch index and FIFO idle level remain unchanged.

The size of the Tx FIFO elements allocated in the Message RAM is a number of 32-bit words (4 bytes). Therefore, the starting address of the next available (free) Tx FIFO is calculated by multiplying the index MCAN_TXFQS.TFQPI (0~31) by the element size (words), multiplying it by 4, and then adding it to MCAN_TXBC.TBSA.

30.3.7.4 Send Queue

send queue mode is achieved by configuring MCAN_TXBC.TFQM to 1. Messages stored in the send queue are sent starting with the message with the smallest ID (highest priority). If multiple send queues are configured with the same message ID, the sending order cannot be predicted because the sending order is related to the numbers of the send queues storing the messages, and the numbers of these send queues depend on the current state of the index.

New messages must be written into the Tx Buffer pointed to by index MCAN_TXFQS.TFQPI. The put index always points to the smallest free buffer in the send queue. If the send queue is full (MCAN_TXFQS.TFQF = 1), the put index has no effect and no further messages should be written to the send queue until at least one requested message has been sent or a pending send request has been canceled.

The application can use register MCAN_TXBRP instead to put the index and put the message into any Tx Buffer that does not have a pending send request.

The size of the send queue buffer elements allocated in the message RAM is a number of 32-bit words (4 bytes). Therefore, the starting address of the next available (free) send queue buffer is

calculated by multiplying the put index MCAN_TXFQS.TFQPI (0~31) by the element size (words), multiplying it by 4, and then adding it to MCAN_TXBC.TBSA.

30.3.7.5 Mixing Dedicated Tx Buffers and Tx FIFOs

In this case, the Tx Buffer in the message RAM is divided into a set of dedicated Tx Buffers and a Tx FIFO. The number of dedicated Tx Buffers is configured via MCAN_TXBC.NDTB. The number of Tx Buffers allocated to the Tx FIFO is configured via MCAN_TXBC.TFQS. If MCAN_TXBC.TFQS is configured to 0, only the dedicated Tx Buffer will be used.

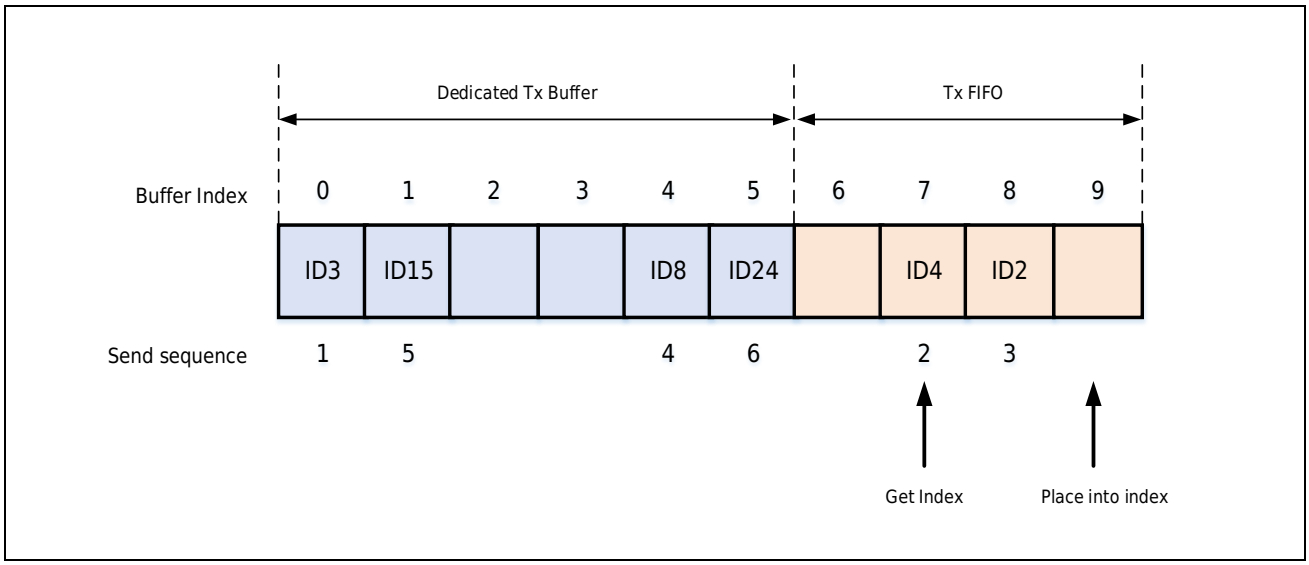


Figure 30-9 Example of Mixed Configuration of Dedicated Tx Buffer and Tx FIFO

Send priority :

- Scan all dedicated Tx Buffers with activated send requests and the Tx FIFO buffer with the earliest pending send request (the buffer pointed to by MCAN_TXEFS.TFGI)
- The buffer with the smallest message ID has the highest priority, and the message in that buffer will be sent next time.

30.3.7.6 Mixing Dedicated Tx Buffers and Send Queues

In this case, the Tx Buffer in the message RAM is divided into a set of dedicated Tx Buffers and a send queue. The number of dedicated Tx Buffers is configured via MCAN_TXBC.NDTB. The number of send queue buffers is configured via MCAN_TXBC.TFQS. If MCAN_TXBC.TFQS is configured to 0, only the dedicated Tx Buffer will be used.

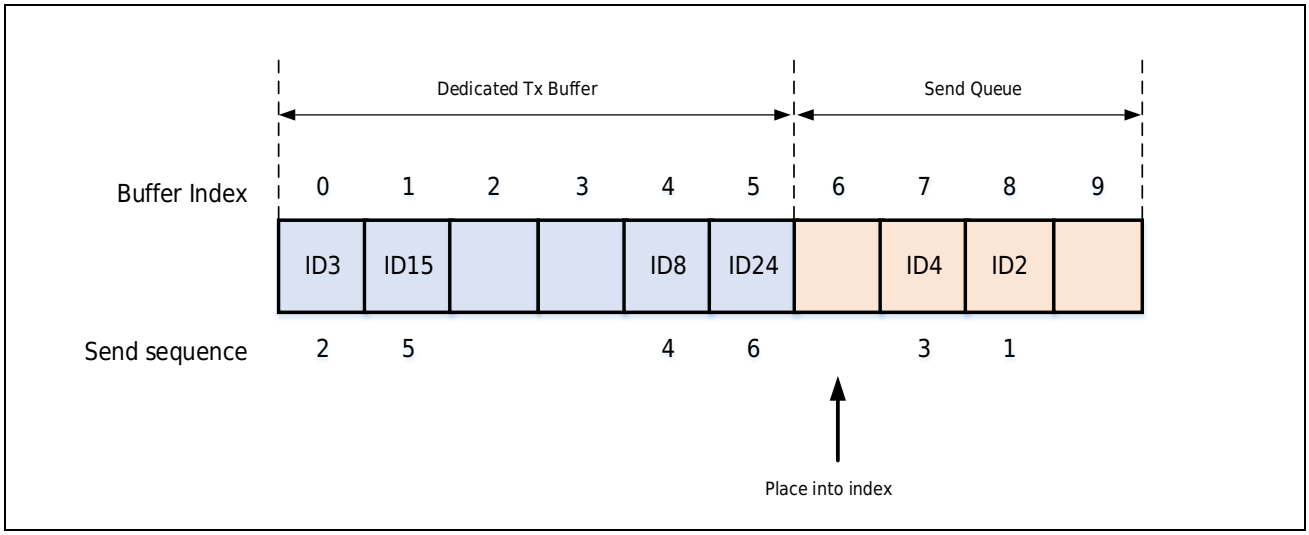


Figure 30-10 Example of Mixed Configuration of Dedicated Tx Buffer and Send Queue

Send priority :

- Scan all dedicated Tx Buffers and send queue buffers with activated send requests
- The Tx Buffer with the smallest message ID has the highest priority, and the message in that buffer will be sent next time.

30.3.7.7 Send Cancel

MCAN supports send cancel function. This functionality is particularly suitable for gateway applications and AUTOSAR-based applications. To cancel the send request of a dedicated Tx Buffer or a send queue buffer, the master must write 1 to the corresponding bit in register MCAN_TXBCR (the bit corresponding to the Tx Buffer index). Send cancel does not work in Tx FIFO mode.

After successful cancellation, the corresponding bit of register MCAN_TXBCF will be set to 1.

If a request is made to cancel a transmission while a buffer transmission is in progress, the corresponding bit in MCAN_TXBRP remains set as long as the transmission is in progress. If the transmission is successful, the corresponding bits of MCAN_TXBTO and MCAN_TXBCF will be set to 1. If the transmission is unsuccessful, it will not be sent repeatedly, and only the corresponding bit of MCAN_TXBCF will be set.

Note:

If a pending send is canceled immediately before it starts sending, there will be a short window after the cancellation during which no send will start, even if there is another message pending in that node. This allows another node to send a message that may have a lower priority than the second message in this node.

30.3.7.8 Tx Event Processing

MCAN uses the Tx Event FIFO to support the processing of Tx Events. After the MCAN sends a message on the CAN bus, the message ID and timestamp are stored in the Tx Event FIFO element.

To associate a Tx Event with a Tx Event FIFO element, the message tags in the sent Tx Buffer are copied to the Tx Event FIFO element.

The Tx Event FIFO can be configured with up to 32 elements. Section [Tx Event FIFO Element] describes the Tx Event FIFO elements.

The purpose of the Tx Event FIFO is to separate the processing of the send status information from the processing of the send message, that is, to let the Tx Buffer only save the message to be sent, and store the send status separately in the Tx Event FIFO. This has advantages, especially when dealing with dynamically managed send queues, as the Tx Buffer can be used for new messages immediately after a successful send. Before rewriting the Tx Buffer, there is no need to save the send status information of the Tx Buffer.

When MCAN_IR.TEFF indicates that the Tx Event FIFO is full, no further elements are written to the Tx Event FIFO until at least one element has been read and the Tx Event FIFO get index has been incremented. If a Tx Event occurs when the Tx Event FIFO is full, the event is discarded and the interrupt flag MCAN_IR.TEFL is set.

To prevent the Tx Event FIFO from overflowing, the Tx Event FIFO watermark can be used. When the Tx Event FIFO fill count reaches the Tx Event FIFO watermark configured by TXEFC.EFWM, the interrupt flag MCAN_IR.TEFW is set.

When reading data from the Tx Event FIFO, the Tx Event FIFO get index TXEFS.EFGI must be multiplied by 2 (the Tx Event FIFO element size is two words), then multiplied by 4, and then added to the Tx Event FIFO start address TXEFC.EFSA to get the address where the target data is located.

30.3.8 FIFO Acknowledgement Processing

The acquisition index of Rx FIFO0, Rx FIFO1 and Tx Event FIFO is controlled by writing to the corresponding FIFO confirmation index register. Please refer to Chapter [MCAN Rx FIFO0 Acknowledge Register (MCAN_RXF0A)], Chapter [MCAN Rx FIFO1 Acknowledge Register (MCAN_RXF1A)] and Chapter [MCAN Tx Event FIFO Acknowledge Register (MCAN_TXEFA)]. 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 fetch index can be updated simply by writing the index value of the last element into 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 Rx 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.

30.3.9 Message RAM

30.3.9.1 Message RAM Configuration

The width of the message RAM is 32 bits. The MCAN module can allocate up to 512 words in the Message RAM as shown in Figure 30-11. In actual applications, it is not required to configure all parts, and there is no restriction on the order of the parts.

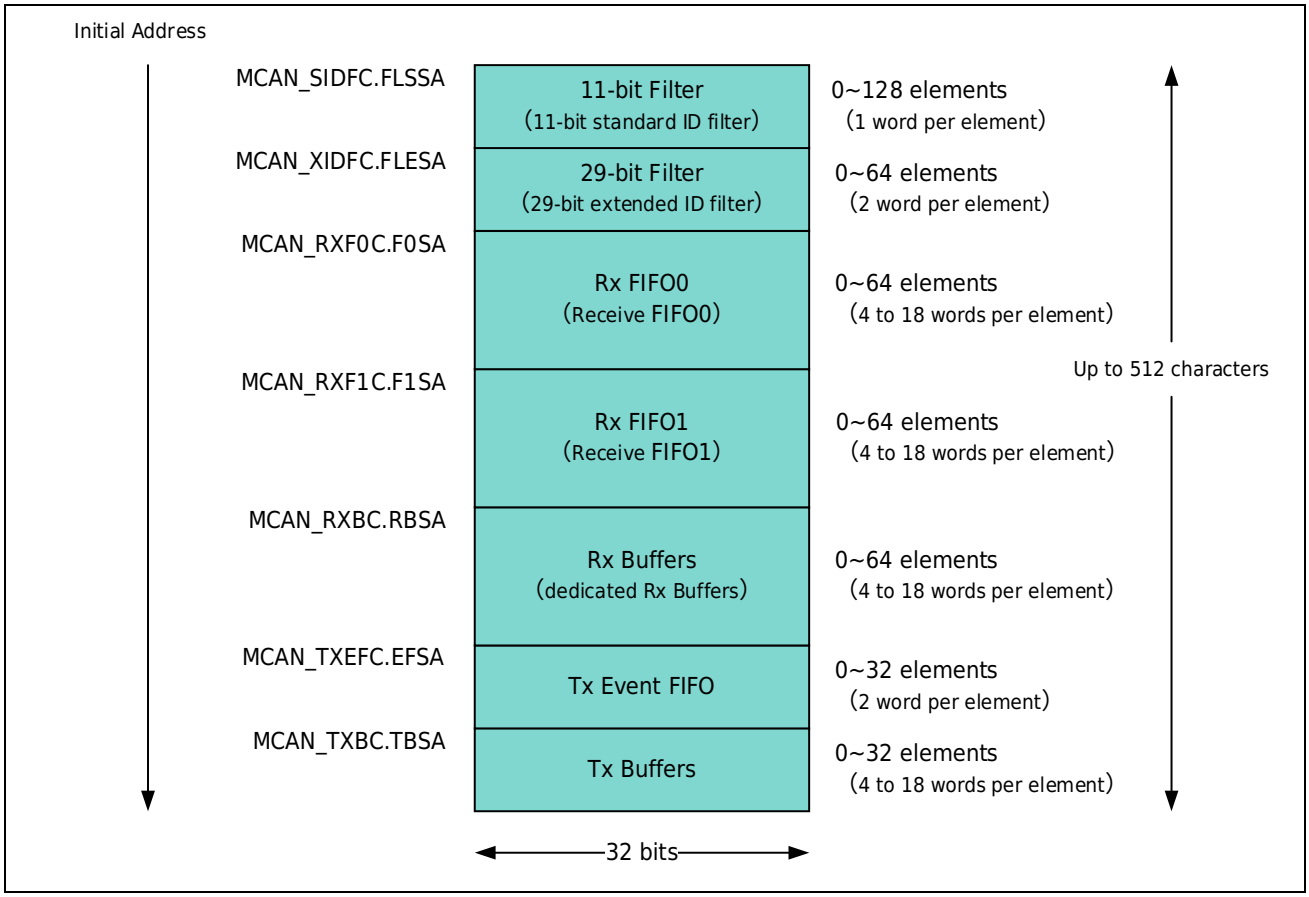


Figure 30-11 Message RAM Configuration

When MCAN accesses the message RAM, it needs to address it at a 32-bit aligned address instead of a byte aligned address. The configurable starting address is a 32-bit word address, i.e. only bits 15 to 2 of the address are evaluated, ignoring the two least significant bits.

The message RAM has ECC (Error Checking and Correcting). ECC is a one-correction-two-check code, which means it can correct one-bit error and detect two-bit errors.

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.

30.3.9.2 Rx Buffer and FIFO Elements

Up to 64 Rx Buffer s and 2 Rx FIFOs can be configured in the message RAM. Each Rx FIFO can be configured to store up to 64 messages. The structures of the Rx Buffer and Rx FIFO elements are shown in Table 30-9 and are described in detail in Table 30-10. 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 30-9 Rx Buffer and FIFO Elements

Bit	31 24							23 16		15 8	7 0
R0	ESI	XTD	RTR	ID[28:0]							
R1	ANMF	FIDX[6:0]						Res	FDF BRS DLC[3:0]	RXTS[15:0]	
R2	DB3[7:0]							DB2[7:0]		DB1[7:0]	DB0[7:0]
R3	DB7[7:0]							DB6[7:0]		DB5[7:0]	DB4[7:0]
…	…							…		…	…
Rn	DBm[7:0]							DBm-1[7:0]		DBm-2[7:0]	DBm-3[7:0]

Table 30-10 Rx Buffer and FIFO Elements Description

Bit Field	Marking	Description
R0[31]	ESI	Error State Indicator 0: The sending node is in error active state 1: The sending node is in error passive state
R0[30]	XTD	Extended Identifier Indicates to the master whether the ID of the received frame is a standard ID or an extended ID. 0: 11-bit standard ID 1: 29-bit extended ID
R0[29]	RTR	Remote Transmission Request Indicates to the master whether the received frame is a data frame or a remote frame. 0: The received frame is a data frame 1: 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.*
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-matching frames can be enabled via MCAN_GFC.ANFS and MCAN_GFC.ANFE. 0: The received frame matches the filter element pointed to by filter index FIDX 1: 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 (invalid when ANMF = 1)

Bit Field	Marking	Description
		The range is 0 to MCAN_SIDFC.LSS] minus 1 or MCAN_XIDFC.LSE minus 1.
R1[21]	FDF	FD Format 0: Classic CAN frame format 1: CAN FD frame format (new DLC encoding and CRC)
R1[20]	BRS	Bit Rate Switch 0: No bit rate switching in received frames 1: The received frame has 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
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

30.3.9.3 Tx Buffer Elements

The Tx Buffer can be configured as a dedicated Tx Buffer and a Tx FIFO (or queue). If the Tx Buffer is shared by a dedicated Tx Buffer and a Tx FIFO (or queue), the dedicated Tx Buffer is at the beginning of the Tx Buffer, followed by the Tx FIFO (or queue). The send processing unit evaluates the Tx Buffer configurations MCAN_TXBC.TFQS and MCAN_TXBC.NDTB to differentiate between the dedicated Tx Buffer and the Tx FIFO (queue). The structure of the Tx Buffer element is shown in the Table 30-11. For related descriptions, see Table 30-12. The element size can be configured via register MCAN_TXESC. The data field of a CAN FD message can be up to 64 bytes.

Table 30-11 Buffer Elements

Bit	31 24	23 16	15 8	7 0
T0	ESI / XTD / RTR	ID[28:0]		
T1	MM[7:0]	EFC / Res / FDF / BRS / DLC[3:0]	MM[15:8]	Res
T2	DB3[7:0]	DB2[7:0]	DB1[7:0]	DB0[7:0]
T3	DB7[7:0]	DB6[7:0]	DB5[7:0]	DB4[7:0]
…	…	…	…	…
Tn	DBm[7:0]	DBm-1[7:0]	DBm-2[7:0]	DBm-3[7:0]

Table 30-12 Tx Buffer Element Description

Bit Field	Marking	Description
T0[31]	ESI	Error State Indicator 0: The ESI bit of the CAN FD format depends only on the error passive flag 1: The ESI bit of the CAN FD format is sent as a recessive bit **Note:** *The ESI bit of the Tx Buffer is ORed with the error passive flag to determine the value of the ESI bit of the FD frame to be sent. As required by the CAN FD protocol specification, an error active node may choose to send a recessive ESI bit, but an error passive node will always send a recessive ESI bit.*
T0[30]	XTD	Extended Identifier 0: 11-bit standard ID 1: 29-bit extended ID
T0[29]	RTR	Remote Transmission Request 0: Send data frame 1: Send 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 Tx Buffer configuration. Copied to the Tx Event FIFO element to identify the send message status.
T1[23]	EFC	Event FIFO Control 0: Do not store Tx Events 1: Store Tx Events
T1[21]	FDF	FD Format 0: Send frames in classic CAN format 1: Send frames in CAN FD format

Bit Field	Marking	Description
T1[20]	BRS	Bit Rate Switch 0: Do not switch the bit rate when sending CAN FD frames 1: Switch bit rate when sending CAN FD frames **Note:** *The ESI, FDF and BRS bits are evaluated only if CAN FD is enabled (MCAN_CCCR.FDOE=1). The BRS bit is only evaluated if MCAN_CCCR.BRSE = 1 is additionally set.*
T1[19:16]	DLC[3:0]	Data Length Code 0~8: Classic CAN and CAN FD: The senting frame contains 0~8 data bytes 9~15: CAN: The sending frame contains 8 data bytes 9~15: CAN FD: The sending 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 Tx Buffer configuration. Copied to the Tx Event FIFO element to identify the send message status. Only valid when MCAN_CCCR.WMM=1.
T2[31:24]	DB3[7:0]	Data Byte 3 **Note:** *Depending on the element size configured in 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
…	…	…
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

30.3.9.4 Tx Event FIFO Element

Each element stores information about the sent message. By reading the Tx Event FIFO, the master CPU obtains the messages in the order they were sent. The status information of the Tx Event FIFO can be obtained from the register MCAN_TXEFS. The structure of the Tx Event FIFO element is shown in the Table 30-13. For related descriptions, see Table 30-14.

E1A: When MCAN_CCCR.WMM = 0, E1A.TXTS[15:0] holds the 16-bit timestamp generated by the internal timestamp logic of the MCAN.

E1B: When 16-bit wide message mark is enabled (MCAN_CCCR.WMM=1), E1B.MM[15:8] is the upper 8 bits of the wide message mark.

Table 30-13 Tx Event FIFO Elements

Bit	31 24				23 16				15 8	7 0
E0	ESI	XTD	RTR	ID[28:0]						
E1A	MM[7:0]				ET[1:0]	FDF	BRS	DLC[3:0]	TXTS[15:0]	
E1B	MM[7:0]				ET[1:0]	FDF	BRS	DLC[3:0]	MM[15:8]	Res

Table 30-14 Tx Event FIFO Element Description

Bit Field	Marking	Description
E0[31]	ESI	Error State Indicator 0: The sending node is in error active state 1: The sending node is in error passive state
E0[30]	XTD	Extended Identifier 0: 11-bit standard ID 1: 29-bit extended ID
E0[29]	RTR	Remote Transmission Request 0: The frame to be sent is a data frame 1: The senting 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 Tx Buffer to the Tx Event element to identify the status of the send message.
E1A/B[23:22]	ET[1:0]	Event Type 00: reserved 01: Tx Event 10: sending even after cancellation (always this value when sending in DAR mode) 11: reserved
E1A/B[21]	FDF	FD Format 0: Classic CAN frame format 1: CAN FD frame format (new DLC encoding and CRC)
E1A/B[20]	BRS	Bit Rate Switch 0: CAN FD frames are sent without bit rate switching 1: CAN FD frames are sent with bit rate switching
E1A/B[19:16]	DLC[3:0]	Data Length Code 0~8: Classic CAN and CAN FD: The senting frame contains 0~8 data bytes 9~15: Classic CAN: The senting frame contains 8 data bytes 9~15: CAN FD: The sending frame contains 12/16/20/24/32/48/64 data bytes
E1A[15:0]	TXTS[15:0]	Tx Timestamp

Bit Field	Marking	Description
		The timestamp counter value captured when the SOF (Start Of Frame) is sent. The resolution depends on the configuration of the timestamp counter prescaler MCAN_TSCC.TCP.
E1B[15:8]	MM[15:8]	High byte of Wide Message Marker Written by the CPU during Tx Buffer configuration. Copied to the Tx Event FIFO element to identify the send message status.

30.3.9.5 Standard Message ID Filter Elements

Up to 128 filter elements (each element size is 1 word) can be configured for an 11-bit standard ID. Its address is the filter element index (0~127) multiplied by 4 (converted to bytes) plus the starting address MCAN_SIDFC.FLSSA. The structure of the standard message ID element is shown in the Table 30-15. For related descriptions, see Table 30-16.

Table 30-15 Standard Message ID Filter Elements

Bit	31 24		23 16	15 8	7 0
S0	SFT [1:0]	SFEC [2:0]	SFID1[10:0]	Res	SFID2[10:0]

Table 30-16 Standard Message ID Filter Element Description

Bit Field	Marking	Description
S0[31:30]	SFT[1:0]	Standard Filter Type 00: Range filter from SFID1 to SFID2 (SFID2 ≥ SFID1) 01: Dual ID filter defined by SFID1 or SFID2 10: Classic filter: SFID1=ID, SFID2=mask 11: Disable this filter element **Note:** *When SFT = 0b11, this filter element disable 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. 000: Disable this filter element 001: If a match is found, store it in Rx FIFO0 010: If a match is found, store it in Rx FIFO1 011: If match, reject ID 100: If matched, set priority, do not store 101: If match, set priority and store into FIFO0 110: If match, set priority and store into FIFO1 111: Stored in the Rx Buffer or as a debug message, ignoring the configuration of SFT[1:0]
S0[26:16]	SFID[10:0]	Standard Filter ID 1 The first ID of the standard ID filter element. When filtering for Rx Buffer or debug messages, this bitfield defines the ID of the standard message to be stored. The received ID must match exactly, no masking 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 receiving buffer or debug messages
	SFID2[10:9]	Determines whether the received message is stored in the Rx Buffer or processed as message A, B, or C in the debug message sequence. 00: Store the message in the Rx Buffer 01: Debug message A 10: Debug message B 11: Debug message C
	SFID2[5:0]	Defines the offset (buffer index) of the Rx Buffer where matching messages are stored to the starting address RXBC.RBSA.

30.3.9.6 Extended Message ID Filter Element

Up to 64 filter elements (each element size is 2 words) 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 is shown in theTable 30-17. For related descriptions, see Table 30-18.

Table 30-17 Extended Message ID Filter Elements

Bit	31 24		23 16	15 8	7 0
F0	EFEC [2:0]		EFID1[28:0]		
F1	EFT [1:0]	Res	EFID2[28:0]		

Table 30-18 4Extended Message ID Filter Element Description

Bit Field	Marking	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. 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. 000: Disable this filter element 001: If a match is found, store it in Rx FIFO0 010: If a match is found, store it in Rx FIFO1 011: If match, reject ID 100: If matched, set priority 101: If match, set priority and store into FIFO0 110: If match, set priority and store into FIFO1 111: Stored in the Rx Buffer or as a debug message, ignoring the configuration of EFT[1:0]
F0[28:0]	EFID1[28:0]	Extended Filter ID 1 Extension ID The first ID of the filter element. When filtering for Rx Buffer or debug messages, this bitfield defines the ID of the extended message to be stored. The received ID must match exactly, using the MCAN_XIDAM masking mechanism only.
F1[31:30]	EFT[1:0]	Extended Filter Type 00: Range filter from EFID1 to EFID2 (EFID2 ≥ EFID1) 01: Dual ID filter defined by EFID1 or EFID2 10: Classic filter: EFID1=ID, EFID2=mask 11: Range filter from EFID1 to EFID2 (EFID2 ≥ EFID1), without using 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 – When EFEC = 0b111: filter for receiving buffer or debug messages
	EFID2[10:9]	Determines whether the received message is stored in the Rx Buffer or processed as message A, B, or C in the debug message sequence. 00: Store the message in the Rx Buffer 01: Debug message A 10: Debug message B 11: Debug message C
	EFID2[5:0]	Defines the offset (buffer index) of the Rx Buffer where matching messages are stored to the starting address MCAN_RXBC.RBSA.

30.4Register Description

Table 30-19 MCAN/MCANRAM Base Address

Name	Base Address	Description
MCAN1	0x4002 9000	MCAN1 base address
MCAN2	0x4002 9400	MCAN2 base address
MCANRAM	0x4007 3800	MCANRAM base address

Table 30-20 MCAN Register List

Register	Description	Offset Address	Bit Width	Reset Value
MCAN_ENDN	MCAN Byte Order Register	0x0004	32	0x8765 4321
MCAN_DBTP	MCAN Data Bit Time and Prescaler Register	0x000C	32	0x0000 0A33
MCAN_TEST	MCAN Test Register	0x0010	32	0x0000 00X0
MCAN_RWD	MCAN RAM Watchdog Register	0x0014	32	0x0000 0000
MCAN_CCCR	MCAN CC Control Register	0x0018	32	0x0000 0001
MCAN_NBTP	MCAN Nominal Bit time and Prescaler Register	0x001C	32	0x0600 0A03
MCAN_TSCC	MCAN Timestamp Counter Configuration Register	0x0020	32	0x0000 0000
MCAN_TSCV	MCAN Timestamp Counter Value Register	0x0024	32	0x0000 0000
MCAN_TOCC	MCAN Timeout Counter Configuration Register	0x0028	32	0xFFFF 0000
MCAN_TOCV	MCAN Timeout Counter Value Register	0x002C	32	0x0000 FFFF
MCAN_ECR	MCAN Error Counter Register	0x0040	32	0x0000 0000
MCAN_PSR	MCAN Protocol Status Register	0x0044	32	0x0000 0707
MCAN_TDCR	MCAN send Delay Compensation Register	0x0048	32	0x0000 0000
MCAN_IR	MCAN Interruption Register	0x0050	32	0x0000 0000
MCAN_IE	MCAN interrupt enable register	0x0054	32	0x0000 0000
MCAN_ILS	MCAN Interrupt Line Selection Register	0x0058	32	0x0000 0000
MCAN_ILE	MCAN Interrupt Line Enable Register	0x005C	32	0x0000 0000
MCAN_GFC	MCAN Global Filter Configuration Register	0x0080	32	0x0000 0000
MCAN_SIDFC	MCAN Standard ID Filter Configuration Register	0x0084	32	0x0000 0000
MCAN_XIDFC	MCAN Extended ID Filter Configuration Register	0x0088	32	0x0000 0000
MCAN_XIDAM	MCAN Extended ID and Mask Register	0x0090	32	0x1FFF FFFF
MCAN_HPMS	MCAN High Priority Message Status Register	0x0094	32	0x0000 0000
MCAN_NDAT1	MCAN New Data 1 Register	0x0098	32	0x0000 0000
MCAN_NDAT2	MCAN New Data 2 Register	0x009C	32	0x0000 0000
MCAN_RXF0C	MCAN Rx FIFO0 Configuration Register	0x00A0	32	0x0000 0000
MCAN_RXF0S	MCAN Rx FIFO0 Status Register	0x00A4	32	0x0000 0000
MCAN_RXF0A	MCAN Rx FIFO0 Acknowledge Register	0x00A8	32	0x0000 0000
MCAN_RXBC	MCAN Rx Buffer Configuration Register	0x00AC	32	0x0000 0000

Register	Description	Offset Address	Bit Width	Reset Value
MCAN_RXF1C	MCAN Rx FIFO1 Configuration Register	0x00B0	32	0x0000 0000
MCAN_RXF1S	MCAN Rx FIFO1 Status Register	0x00B4	32	0x0000 0000
MCAN_RXF1A	MCAN Rx FIFO1 Acknowledge Register	0x00B8	32	0x0000 0000
MCAN_RXESC	MCAN Rx Buffer and FIFO Element Size Configuration Registers	0x00BC	32	0x0000 0000
MCAN_TXBC	MCAN Tx Buffer Configuration Register	0x00C0	32	0x0000 0000
MCAN_TXFQS	MCAN Tx FIFO/Queue Status Register	0x00C4	32	0x0000 0000
MCAN_TXESC	MCAN Tx Buffer Element Size Configuration Register	0x00C8	32	0x0000 0000
MCAN_TXBRP	MCAN Tx Buffer Request Pending Register	0x00CC	32	0x0000 0000
MCAN_TXBAR	MCAN Tx Buffer Add Request Register	0x00D0	32	0x0000 0000
MCAN_TXBCR	MCAN Tx Buffer Cancel Request Register	0x00D4	32	0x0000 0000
MCAN_TXBTO	MCAN Tx Buffer Send Occurred Register	0x00D8	32	0x0000 0000
MCAN_TXBCF	MCAN Tx Buffer Cancel Completed Register	0x00DC	32	0x0000 0000
MCAN_TXBTIE	MCAN Tx Buffer Send Interrupt Enable Register	0x00E0	32	0x0000 0000
MCAN_TXBCIE	MCAN Tx Buffer Cancel Completed Interrupt Enable Register	0x00E4	32	0x0000 0000
MCAN_TXEFC	MCAN Tx Event FIFO Configuration Register	0x00F0	32	0x0000 0000
MCAN_TXEFS	MCAN Tx Event FIFO Status Register	0x00F4	32	0x0000 0000
MCAN_TXEFA	MCAN Tx Event FIFO Acknowledge Register	0x00F8	32	0x0000 0000

Table 30-21 MCANRAM Register List

Register	Description	Offset Address	Bit Width	Reset Value
MCANRAM_CKCR	MCAN Message RAM Check Control Register	0x00	32	0x00000000
MCANRAM_CKSR	MCAN message RAM check status register	0x04	32	0x00000000
MCANRAM_EIEN	MCAN Message RAM Error Injection Enable Register	0x08	32	0x00000000
MCANRAM_EIBIT0	MCAN Message RAM Error Injection Bit Enable Register 0	0x0C	32	0x00000000
MCANRAM_EIBIT1	MCAN Message RAM Error Injection Bit Enable Register 1	0x10	32	0x00000000
MCANRAM_ECCERRADDR	MCAN Message RAM Check Error Address Register	0x14	32	0x00000000

30.4.1 MCAN Byte Order Register (MCAN_ENDN)

MCAN Endianness Register

Reset value: 0x8765 4321

b31	b30	b29	b28	b27	b26	b25	b24
ETV[31:24]							
b23	b22	b21	b20	b19	b18	b17	b16
ETV[23:16]							
b15	b14	b13	b12	b11	b10	b9	b8
ETV[15:8]							
b7	b6	b5	b4	b3	b2	b1	b0
ETV[7:0]							

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b0	ETV[31:0]	Byte Order Test Value	The test value is 0x8765 4321.	R

30.4.2 MCAN Data Bit Time and Prescaler Register (MCAN_DBTP)

MCAN Data Bit Timing and Prescaler Register

Reset value: 0x0000 0A33

Write access to this register is possible only when both the MCAN_CCCR.CCE and MCAN_CCCR.INIT bits are set. The programmable range of the CAN bit time is 4 to 49 time slices (Time quanta, tq). The programmable range of the CAN time slice is 1 to 32 MCAN communication clock cycles (Minimum time quantum, mtq). tq = (DBRP + 1)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 sample point.

b31	b30	b29	b28	b27	b26	b25	b24
Res							
b23	b22	b21	b20	b19	b18	b17	b16
TDC	Res		DBRP[4:0]				
b15	b14	b13	b12	b11	b10	b9	b8
Res			DTSEG1[4:0]				
b7	b6	b5	b4	b3	b2	b1	b0
DTSEG2[3:0]				DSJW[3:0]			

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b24	Res	Reserved Bit	When reading, it is 0. When writing, please write 0	R
b23	TDC	Send Delay Compensation	0: Disable sending delay compensation 1: Enable send delay compensation	RWA
b22~b21	Res	Reserved Bit	When reading, it is 0. When writing, please write 0	R
b20~b16	DBRP[4:0]	Data bit rate pre-scaling	Generates the MCAN communication clock pre-division value for the bit time slice, where the bit time is a multiple of the time slice. The valid pre-scaling value is 0 to 31. When TDC=1, the range is limited to 0 to 1, and the actual value used by the hardware is the programmed value plus 1.	RWA
b15~b13	Res	Reserved Bit	When reading, it is 0. When writing, please write 0	R
b12~b8	DTSEG1[4:0]	Data time before the sampling point	The valid value is 0 to 31, and the actual value used by the hardware is the programmed value plus 1.	RWA
b7~b4	DTSEG2[3:0]	Data time after sampling point	The valid value is 1 to 15. The actual value used by the hardware is the programmed value plus 1.	RWA
b3~b0	DSJW[3:0]	Data (re)sync jump width	The valid value is 0 to 15, and the actual value used by the hardware is the programmed value plus 1.	RWA

Note:

1. When the MCAN communication clock is 8MHz, the reset value 0x0000 0A33 will configure the bit rate of the MCAN data phase to 500Kbps.
2. The CAN FD data phase bit rate configured via DBTP cannot be less than the arbitration phase bit rate configured via NBTP.

30.4.3 MCAN Test Register (MCAN_TEST)

MCAN Test Register

Reset value: 0x0000 00X0 (the reset value of bit 7 is determined by the actual level of the MCAN_RX pin)

Write access to this test register is only possible when the MCAN_CCCR.TEST bit is set to 1. When the MCAN_CCCR.TEST bit is cleared to 0, all test register functions are set to their reset values.

Loopback mode and software control of the MCAN_TX pin belong to hardware test mode. Configuring TX to anything other than 0b00 may interfere with message transmission on the CAN bus.

b31	b30	b29	b28	b27	b26	b25	b24
Res							
b23	b22	b21	b20	b19	b18	b17	b16
Res		SVAL	TXBNS[4:0]				
b15	b14	b13	b12	b11	b10	b9	b8
Res		PVAL	TXBNP[4:0]				
b7	b6	b5	b4	b3	b2	b1	b0
RX	TX[1:0]		LBCK	Res			

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b22	Res	Reserved Bit	When reading, it is 0. When writing, please write 0	R
b21	SVAL	Already effective	0: The value of TXBNS invalid 1: The value of TXBNS valid	R
b20~b16	TXBNS[4:0]	The number of the buffer that has started sending	The Tx Buffer number from which the last transmission started. It valid when the SVAL bit is set to 1, and the valid range is 0~31.	R
b15~b14	Res	Reserved Bit	When reading, it is 0. When writing, please write 0	R
b13	PVAL	Ready to use	0: TXBNP value invalid 1: TXBNP value valid	R
b12~b8	TXBNP[4:0]	The number of buffers that are ready to send	The Tx Buffer number that is ready to send. It valid when the PVAL bit is set to 1, and the valid range is 0~31.	R
b7	RX	Receive Pin	The actual value of the MCAN_RX pin monitored 0: CAN bus is dominant (MCAN_RX=0) 1: CAN bus is recessive (MCAN_RX=1)	R
b6~b5	TX[1:0]	Send pin control	00: reset value, MCAN_TX pin is controlled by the CAN core and updated at the end of the CAN bit time 01: The sampling point can be monitored on the MCAN_TX pin 10: MCAN_TX pin outputs dominant level (0) 11: MCAN_TX pin outputs recessive level (1)	RWA
b4	LBCK	Loop mode	0: reset value, disable loopback mode 1: Enable loopback mode (see [External Loopback Mode] and [Internal Loopback Mode] sections for details)	RWA
b3~b0	Res	Reserved Bit	When reading, it is 0. When writing, please write 0	R

30.4.4 MCAN RAM Watchdog Register (MCAN_RWD)

MCAN RAM Watchdog Register

Reset value: 0x0000 0000

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 successful completion of the access by activating its READY signal output, the counter is reloaded with the value of MCAN_RWD.WDC. If there is no response from the message RAM before the counter decrements to 0, the counter will stop counting and the interrupt flag MCAN_IR.WDI will be set to 1. The RAM watchdog counter is counted by the MCAN control logic clock.

b31	b30	b29	b28	b27	b26	b25	b24
Res							
b23	b22	b21	b20	b19	b18	b17	b16
Res							
b15	b14	b13	b12	b11	b10	b9	b8
WDV[7:0]							
b7	b6	b5	b4	b3	b2	b1	b0
WDC[7:0]							

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b16	Res	Reserved Bit	When reading, it is 0. When writing, please write 0	R
b15~b8	WDV[7:0]	Watchdog count value	Actual value of the message RAM watchdog counter	R
b7~b0	WDC[7:0]	Watchdog Configuration	Starting value of the message RAM watchdog counter. If the initial value 0x00 is used, the counter disable.	RWA

30.4.5 MCAN CC Control Register (MCAN_CCCR)

MCAN CC Control Register

Reset value: 0x0000 0001

For the setting conditions of each bit of the register, please refer to the [Software Initialization] section.

b31	b30	b29	b28	b27	b26	b25	b24
Res							
b23	b22	b21	b20	b19	b18	b17	b16
Res							
b15	b14	b13	b12	b11	b10	b9	b8
NISO	TXP	EFBI	PXHD	WMM	Res	BRSE	FDOE
b7	b6	b5	b4	b3	b2	b1	b0
TEST	DAR	MON	CSR	CSA	ASM	CCE	INIT

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b16	Res	Reserved Bit	When reading, it is 0. When writing, please write 0	R
b15	NISO	Non-ISO mode selection	0: Use the CAN FD frame format specified in ISO 11898-1:2015 1: Use the CAN FD frame format specified by the Bosch CAN FD specification V1.0	RWA
b14	TXP	Transfer Pause	If this bit is set to 1, MCAN will pause for two bit times after successfully sendting a frame of data before sendting the next frame of data. 0: Disable transfer pause 1: Enable transfer pause	RWA
b13	EFBI	Edge filtering during bus synchronization	0: Disable edge filtering 1: Two consecutive dominant tqs are required to detect a hard sync edge	RWA
b12	PXHD	Protocol exception handling prohibited	0: Enable protocol exception handling 1: Disable protocol exception handling **Note:** When protocol exception handling disable, the MCAN will send an error frame when a protocol exception condition is detected.	RWA
b11	WMM	Wide message tag	When using 16-bit wide message markers (WMM=1), the 16-bit internal timestamp in the Tx Event FIFO invalid. 0: Use 8-bit message tag 1: Use 16-bit message tag to replace the 16-bit timestamp in the Tx Event FIFO	RWA
b10	Res	Reserved Bit	When reading, it is 0. When writing, please write 0	R
b9	BRSE	Bit rate switching	0: Disable bit rate switching when sending 1: Enable bit rate switching when sending **Note:** When CAN FD mode invalid (FDOE=0), the setting of BRSE has no effect.	RWA
b8	FDOE	FD mode enabled	0: disable FD mode 1: Enable FD mode	RWA
b7	TEST	Test mode enable	0: Regular Mode, the test register MCAN_TEST keeps the reset value 1: Test mode, enable write access to test register MCAN_TEST	RW1A
b6	DAR	Disable automatic resend	0: Enable automatic resending of unsuccessfully sent messages 1: Disable automatic resend	RWA
b5	MON	Bus monitoring mode	The master can set the MON bit only when both the CCE and INIT bits are set. This bit can be reset by the master at any time. 0: Disable bus monitoring mode 1: Enable bus monitoring mode	RW1A
b4	CSR	Clock stop request	When clock stop has been requested, the INIT bit and then the CSA bit will be set to 1 in sequence after all pending transmission requests have been completed and the CAN bus has reached the idle state. 0: Clock stop is not requested 1: Clock stop requested	RW
b3	CSA	Clock stop confirmation	0: Clock stop request is not acknowledged 1: The clock stop request has been acknowledged and the MCAN can be set to the power-down state by stopping the control logic clock and the	R

			communication clock.	
b2	ASM	Restricted operating mode	The master can set the ASM bit only when both CCE and INIT are set. This bit can be reset by the master at any time. For instructions on restricted working mode, please refer to the [Restricted Working Mode] section. 0: Normal mode 1: Restricted working mode	RW1A
b1	CCE	Configuration Change Enable	0: CPU has no write access to protected configuration registers 1: CPU has write access rights to protected configuration registers (when INIT=1)	RWA
b0	INIT	initialization	0: Normal work 1: 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 correctly read. Therefore, before resetting the INIT value, please confirm that the previous setting value has been written.	RW

30.4.6 MCAN Nominal Bit time and Prescaler Register (MCAN_NBTP)

MCAN Nominal Bit Timing and Prescaler Register

Reset value: 0x0600 0A03

Write access to this register is possible only when both the MCAN_CCCR.CCE and MCAN_CCR.INIT bits are set to 1. The programmable range of the CAN nominal bit time is 4 to 385 time slices (Time quanta, tq). The programmable range of the CAN time slice is 1 to 512 MCAN communication clock cycles (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.

b31	b30	b29	b28	b27	b26	b25	b24
NSJW[6:0]							NBRP[8]
b23	b22	b21	b20	b19	b18	b17	b16
NBRP[7:0]							
b15	b14	b13	b12	b11	b10	b9	b8
NTSEG1[7:0]							
b7	b6	b5	b4	b3	b2	b1	b0
Res	NTSEG2[6:0]						

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b25	NSJW[6:0]	Nominal (re)synchronization jump width	The valid value is 0 to 127, and the actual value used by the hardware is the programmed value plus 1.	RWA
b24~b16	NBRP[8:0]	Nominal bit rate prescaler	Generates the MCAN communication clock pre-division value for the bit time slice, where the bit time is a multiple of the time slice. The valid pre-scaling value is 0 to 511, and the actual value used by the hardware is the programmed value plus 1.	RWA
b15~b8	NTSEG1[7:0]	Nominal time period before the sampling point	The valid value is 1 to 255. The actual value used by the hardware is the programmed value plus 1.	RWA
b7	Res	Reserved Bit	When reading, it is 0. When writing, please write 0	R
b6~b0	NTSEG2[6:0]	Nominal time period after the sampling point	The valid value is 1 to 127. The actual value used by the hardware is the programmed value plus 1.	RWA

Note:

When the MCAN clock is 8MHz, the reset value 0x0600 0A03 will configure the MCAN bit rate to 500Kbps.

30.4.7 MCAN Timestamp Counter Configuration Register (MCAN_TSCC)

MCAN Timestamp Counter Configuration Register

Reset value: 0x0000 0000

This register is used to configure the internal 16-bit timestamp counter. For an introduction to internal timestamp processing, please refer to section [Timestamp Generation].

b31	b30	b29	b28	b27	b26	b25	b24
Res							
b23	b22	b21	b20	b19	b18	b17	b16
Res				TCP[3:0]			
b15	b14	b13	b12	b11	b10	b9	b8
Res							
b7	b6	b5	b4	b3	b2	b1	b0
Res						TSS[1:0]	

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b20	Res	Reserved Bit	When reading, it is 0. When writing, please write 0	R
b19~b16	TCP[3:0]	Timestamp counter prescaler	Configure the timestamp and timeout counter time units to 1 to 16 times the CAN bit time. The effective setting value is 0~15, and the actual value used by the hardware is the programmed value plus 1.	RWA
b15~b2	Res	Reserved Bit	When reading, it is 0. When writing, please write 0	R
b1~b0	TSS[1:0]	Timestamp selection	00: The value of the timestamp counter is always 0x0000 01: The timestamp counter value is incremented according to TCP 10: Setting prohibited 11: The value of the timestamp counter is always 0x0000 (equivalent to the set value "00")	RWA

30.4.8 MCAN Timestamp Counter Value Register (MCAN_TSCV)

MCAN Timestamp Counter Value Register

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24
Res							
b23	b22	b21	b20	b19	b18	b17	b16
Res							
b15	b14	b13	b12	b11	b10	b9	b8
TSC[15:8]							
b7	b6	b5	b4	b3	b2	b1	b0
TSC[7:0]							

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b16	Res	Reserved Bit	When reading, it is 0. When writing, please write 0	R
b15~b0	TSC[15:0]	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, and 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. A write operation clears the counter. **Note:** 1. "Wraparound" means that the value of the timestamp counter changes from non-zero to zero, and it is not cleared due to the write operation of MCAN_TSCV. 2. When accessing by byte, writing any byte of this register will cause the timestamp register to be cleared.	RWC

30.4.9 MCAN Timeout Counter Configuration Register (MCAN_TOCC)

MCAN Timeout Counter Configuration Register

Reset value: 0xFFFF 0000

For an introduction to the timeout counter, please refer to chapter [Timeout Counter].

b31	b30	b29	b28	b27	b26	b25	b24
TOP[15:8]							
b23	b22	b21	b20	b19	b18	b17	b16
TOP[7:0]							
b15	b14	b13	b12	b11	b10	b9	b8
Res							
b7	b6	b5	b4	b3	b2	b1	b0
Res					TOS[1:0]		ETOC

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b16	TOP[15:0]	Timeout period	The starting value of the timeout counter (down counter) configures the timeout period.	RW
b15~b3	Res	Reserved Bit	When reading, it is 0. When writing, please write 0	R
b2~b1	TOS[1:0]	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 decrementing downward. When the timeout counter is controlled by one of the FIFOs, an null 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. 00: Continuous mode 01: Timeout controlled by the Tx Event FIFO 10: Timeout controlled by Rx FIFO0 11: Timeout controlled by Rx FIFO1	RWA
b0	ETOC	Timeout counter enable	0: Disable timeout counter 1: Enable timeout counter	RWA

30.4.10 MCAN Timeout Counter Value Register (MCAN_TOCV)

MCAN Timeout Counter Value Register

Reset value: 0x0000 FFFF

b31	b30	b29	b28	b27	b26	b25	b24
Res							
b23	b22	b21	b20	b19	b18	b17	b16
Res							
b15	b14	b13	b12	b11	b10	b9	b8
TOC[15:8]							
b7	b6	b5	b4	b3	b2	b1	b0
TOC[7:0]							

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b16	Res	Reserved Bit	When reading, it is 0. When writing, please write 0	R
b15~b0	TOC[15:0]	Timeout Counter	The timeout counter is decremented by 1 to 16 times the CAN bit time, and the specific multiple is determined by the setting value of MCAN_TSCC.TCP. When the counter decrements to zero, the interrupt flag MCAN_IR.TOO will be set to 1 and the counter will stop counting. The start and reset (or restart) conditions can be set through MCAN_TOCC.TOS. **Note:** For byte access, when MCAN_TOCC.TOS = 0b00, a write to any byte of this register will preset the time-out counter to the value configured by MCAN_TOCC.TOP.	RW

30.4.11 MCAN Error Counter Register (MCAN_ECR)

MCAN Error Counter Register

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24
Res							
b23	b22	b21	b20	b19	b18	b17	b16
CEL[7:0]							
b15	b14	b13	b12	b11	b10	b9	b8
RP	REC[6:0]						
b7	b6	b5	b4	b3	b2	b1	b0
TEC[7:0]							

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b24	Res	Reserved Bit	When reading, it is 0. When writing, please write 0	R
b23~b16	CEL[7:0]	CAN error logging	When a CAN protocol error causes the send error counter TEC or the receive error counter REC to increment, the error record counter CEL will also 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 increases as REC and TEC increase. 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:** For byte accesses, reading byte 2 of the register clears CEL, and reading byte 3, 1, or 0 has no effect.	RC
b15	RP	Receive Error Passive	0: Receive error counter is below error passive level 128 1: The receive error counter has reached error passive level 128	R
b14~b8	REC[6:0]	Receive error counter	The actual status of the receive error counter. The count value range is 0~127.	R
b7~b0	TEC[7:0]	Send error counter	The actual status of the send error counter. The count value range is 0~255.	R

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.

30.4.12 MCAN Protocol Status Register (MCAN_PSR)

MCAN Protocol Status Register

Reset value: 0x0000 0707

b31	b30	b29	b28	b27	b26	b25	b24
Res							
b23	b22	b21	b20	b19	b18	b17	b16
Res	TDCV[6:0]						
b15	b14	b13	b12	b11	b10	b9	b8
Res	PXE	RFDF	RBRS	RESI	DLEC[2:0]		
b7	b6	b5	b4	b3	b2	b1	b0
BO	EW	EP	ACT[1:0]		LEC[2:0]		

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b23	Res	Reserved Bit	When reading, it is 0. When writing, please write 0	R
b22~b16	TDCV[6:0]	Send delay compensation value	The actual sending delay compensation value. The valid value is 0~127 mtq. The position of the 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 stage (Data phase), the SSP position is the number of mtqs between the starting point of the sent bit and the Secondary Sample Point.	R
b15	Res	Reserved Bit	When reading, it is 0. When writing, please write 0	R
b14	PXE	Protocol abnormal events	0: No protocol exception event has occurred since the last read access 1: A protocol exception event has occurred Access type is RX: Reset on read access. **Note:** For byte access, reading byte 0 of the register will clear PXE, and reading bytes 3/2/1 will have no effect.	R
b13	RFDF	CAN FD message received	This bit is set regardless of acceptance filtering. 0: No CAN FD formatted message has been received since the last time it was cleared by the CPU 1: Received a CAN FD format message with the FDF flag set to 1 **Note:** When accessing by byte, reading byte 0 of the register will clear RFDF, and reading bytes 3/2/1 will have no effect.	R
b12	RBRS	BRS flag of the last received CAN FD message	This bit is set together with RFDF and is independent of the acceptance filtering. 0: The BRS flag of the last received CAN FD message was not set 1: The BRS flag of the last received CAN FD message was set **Note:** For byte access, reading byte 0 of the register will clear RBRS, and reading bytes 3/2/1 has no effect.	R
b11	RESI	ESI flag of the last received CAN FD message	This bit is set in conjunction with RFDF, regardless of the acceptance filter. 0: The ESI flag of the last received CAN FD message was not set 1: The ESI flag of the last received CAN FD message was set **Note:** When accessing bytes, reading byte 0 of the register will clear RESI, and reading bytes 3/2/1 will have no effect.	R
b10~b8	DLEC[2:0]	The last error code in the data phase	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 that of LEC. DLEC will be cleared when a CAN FD frame is sent (received or sent) without errors with the BRS flag set. **Note:** For byte access, reading byte 0 of the register will set DLEC to 0b111, reading bytes 3/2/1 has no effect.	R
b7	BO	Buss_Off Status	0: MCAN is not in Bus_Off state 1: MCAN is in Bus_Off state	R
b6	EW	Warning status	0: Both error counters are less than the error warning (Error_Warning) upper limit of 96 1: At least one error counter has reached the Error Warning (Error_Warning) upper limit of 96	R
b5	EP	Error passive	0: MCAN is in Error_Active state. The MCAN participates in bus communications normally and sends active error flags when an error is detected.	R

			1: MCAN is in Error_Passive state	
b4~b3	ACT[1:0]	Communication status	Monitor the communication status of MCAN. 00: Synchronizing, the node is synchronizing CAN communication 01: Idle, the node is neither a receiving node nor a sending node 10: Receive, the node works as a receiving node 11: Send, the node works as a sending node **Note:** Protocol exception events will set ACT to 0b00	R
b2~b0	LEC[2:0]	Last error code	0: No error. No errors have occurred since a successful reception or transmission reset the LEC. 1: Stuff Error. In the received message, more than 5 identical bits appear in a certain part, which is not allowed. 2: Form Error. The fixed-format portion of the received frame had a format error. 3: ACK Error. The message sent by MCAN is not acknowledged by other nodes. 4: Bit1 Error (Bit1 Error). During the message transmission process (except the arbitration field), the device sends a recessive level (logical value 1), but the detected bus level is dominant (logical value 0). 5: Bit0 error. During the message transmission process (including ACK bit, active error flag, overload flag), the device sends a dominant level (logic value 0 of data or ID), but the detected bus level is recessive (logic value 1). During bus Bus_Off recovery, this state is set each time an 11-bit recessive bit sequence is detected. This allows the CPU to monitor the progress of the bus recovery sequence (indicating that the bus is not stuck in the dominant state or continuously disturbed). 6: CRC Error. The CRC of the received message has an error. The received CRC and the CRC calculated using the received data do not match. 7: No Change. Any read access to the Protocol Status Register will reinitialize the LEC to 7. When LEC displays a value of 7, it indicates that no CAN bus event has been detected since the last time the CPU read accessed the protocol status register. **Note:** 1. When a CAN FD frame with the BRS flag set has reached the data phase, the next CAN event (error or valid frame) will be shown in the DLEC instead of the LEC. An error with fixed stuff bits in the CAN FD CRC sequence will show up as a Form Error instead of a Stuff Error. 2. The Bus_Off recovery sequence (please refer to ISO11898-1:2015) cannot be shortened by setting or clearing MCAN_CCCR.INIT. If the device enters Bus_Off state, it will automatically set MCAN_CCCR.INIT, stopping all bus activities. 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 counters 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, so that the CPU can 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. 3. During byte operations, reading byte 0 of the register will set LEC to 0b111, and reading bytes 3/2/1 has no effect.	RS

30.4.13 MCAN Send Delay Compensation Register (MCAN_TDCR)

MCAN Transmitter Delay Compensation Register

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24
Res							
b23	b22	b21	b20	b19	b18	b17	b16
Res							
b15	b14	b13	b12	b11	b10	b9	b8
Res	TDCO[6:0]						
b7	b6	b5	b4	b3	b2	b1	b0
Res	TDCF[6:0]						

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b15	Res	Reserved Bit	When reading, it is 0. When writing, please write 0	R
b14~b8	TDCO[6:0]	Send Delay Compensation Offset SSP	The offset value is the distance between the delay measured from the MCAN_TX pin to the MCAN_RX pin and the Second Sample Point (SSP). The effective setting value is 0~127mtq.	RWA
b7	Res	Reserved Bit	When reading, it is 0. When writing, please write 0	R
b6~b0	TDCF[6:0]	Send compensation window width delay filter	Defines the minimum value of the SSP position. For the measurement of the send delay, a dominant edge on the MCAN_RX pin that would advance the SSP position is ignored. When the TDCF setting value is greater than TDCO, this function is enabled. The effective setting value is 0~127mtq.	RWA

30.4.14 MCAN Interrupt Register (MCAN_IR)

MCAN Interrupt Register

Reset value: 0x0000 0000

When one of the conditions in the following list is detected, the corresponding flag bit will be set. The flag remains set until it is cleared by the master. All flag bits are cleared by writing 1, and have no effect by writing 0. A hard reset clears the registers to zero. Whether an interrupt occurs is controlled by register MCAN_IE; which interrupt line generates the interrupt is controlled by register MCAN_ILS.

b31	b30	b29	b28	b27	b26	b25	b24
Res		ARA	PED	PEA	WDI	BO	EW
b23	b22	b21	b20	b19	b18	b17	b16
EP	ELO	BEU	BEC	DRX	TOO	MRAF	TSW
b15	b14	b13	b12	b11	b10	b9	b8
TEFL	TEFF	TEFW	TEFN	TFE	TCF	TC	HPM
b7	b6	b5	b4	b3	b2	b1	b0
RF1L	RF1F	RF1W	RF1N	RF0L	RF0F	RF0W	RF0N

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b30	Res	Reserved Bit	When reading, it is 0. When writing, please write 0	R
b29	ARA	Access to Reserved Address	Access to Reserved Address 0: No access to the reserved address 1: A reserved address has been accessed	RW1C
b28	PED	Data phase protocol error (using data bit time)	Protocol Error in Data Phase (Data Bit Time is used) 0: No data phase protocol errors were detected 1: Data phase protocol error detected (MCAN_PSR.DLEC≠0/7)	RW1C
b27	PEA	Protocol error during arbitration phase (using nominal bit time)	Protocol Error in Arbitration Phase (Nominal Bit Time is used) 0: No arbitration phase protocol errors were detected 1: A protocol error was detected during the arbitration phase (MCAN_PSR.LEC≠0/7)	RW1C
b26	WDI	watchdog interrupt	Watchdog Interrupt 0: No message RAM watchdog event occurred 1: A message RAM watchdog event occurs due to the loss of the READY signal	RW1C
b25	BO	Bus_Off Status	Bus_Off Status 0: Bus_Off status has not changed 1: Bus_Off state has changed	RW1C
b24	EW	Warning status	Warning Status 0: Error warning (Error_Warning) status has not changed 1: Error Warning (Error_Warning) status has changed	RW1C
b23	EP	Error passive	Error Passive 0: Error_Passive state has not changed 1: Error_Passive state has changed	RW1C
b22	ELO	Error record overflow	Error Logging Overflow 0: CAN error record counter has not overflowed 1: CAN error record counter has overflowed	RW1C
b21	BEU	Uncorrected error bit detection flag	Bit Error Uncorrected 0: No error was detected when reading from the message RAM 1: When MCANRAM_CKCR.ECCMOD is not 0, a bit error was detected and not corrected when reading from the message RAM **Note:** BEU is set while MCAN_CCCR.INIT is set	RW
b20	BEC	Corrected error bit detection flag	Bit Error Corrected 0: No error bit detected when reading from the message RAM 1: When MCANRAM_CKCR.ECCMOD is not 0, a bit error was detected and corrected when reading from the message RAM	RW

b19	DRX	Messages are stored in a dedicated Rx Buffer	Message stored to Dedicated Rx Buffer This flag is set each time a received message has been stored into the dedicated Rx Buffer . 0: The Rx Buffer is not updated 1: At least one message has been stored in the Rx Buffer	RW1C
b18	TOO	Timeout occurred	Timeout Occurred 0: No timeout occurred 1: A timeout has occurred	RW1C
b17	MRAF	Message RAM Access Failure	Message RAM Access Failure Set conditions during receive processing: 1) Before receiving the arbitration field of the subsequent message, the reception filtering or storage of the received message has not been completed. In this case, the reception filtering or message storage is terminated and the reception processing unit will start processing subsequent messages. 2) Unable to write message to message RAM, message storage will be terminated. In both cases, the FIFO put index is not updated, the new data flag of the dedicated Rx Buffer is not set, and the partially stored message is overwritten when the next message is stored at that location. Set conditions during the send process: 1) The sending processing unit cannot read the message from the message RAM in time. In this case, message sending will be terminated. In case of a failure to access the sending processing unit, the MCAN is switched to restricted operating mode (see section [Restricted Working Mode] for details). To exit restricted operating mode, the master CPU must reset MCAN_CCCR.ASM. 0: No message RAM access failure occurred 1: A message RAM access failure occurred	RW1C
b16	TSW	Timestamp wrapping	Timestamp Wraparound 0: The timestamp counter does not wrap around 1: The timestamp counter has wrapped	RW1C
b15	TEFL	Tx Event FIFO element lost	Tx Event FIFO Element Lost 0: Tx Event FIFO elements are not lost 1: The Tx Event FIFO element is lost; this bit is also set when writing to a Tx Event FIFO with a space size of 0	RW1C
b14	TEFF	Tx Event FIFO full	Tx Event FIFO Full 0: The Tx Event FIFO is not full 1: The Tx Event FIFO is full	RW1C
b13	TEFW	Tx Event FIFO reaches watermark	Tx Event FIFO Watermark Reached 0: The fill level of the Tx Event FIFO is below the watermark 1: The fill level of the Tx Event FIFO reaches the watermark	RW1C
b12	TEFN	Send new event FIFO entry	Tx Event FIFO New Entry 0: No change in the Tx Event FIFO 1: The send processing unit has written a Tx Event FIFO element	RW1C
b11	TFE	Transmit FIFO is empty	Transmit FIFO is empty (Tx FIFO Empty 0: The Tx FIFO is not null 1: The Tx FIFO is null	RW1C
b10	TCF	Transmission Cancellation Finished	Transmission Cancellation Finished 0: Send cancel is not completed 1: Send cancel completed	RW1C
b9	TC	Send Complete	Transmission Completed 0: Sending is not completed 1: Sending completed	RW1C
b8	HPM	High priority messages	High Priority Message 0: No high priority message received 1: A high priority message has been received	RW1C
b7	RF1L	Rx FIFO1 message lost	Rx FIFO 1 Message Lost 0: Rx FIFO1 message is not lost 1: Rx FIFO1 message is lost; this bit is also set when writing to Rx FIFO1 with space size 0	RW1C
b6	RF1F	Rx FIFO1 is full	Rx FIFO1 is full (Rx FIFO 1 Full) 0: Rx FIFO1 is not full 1: Rx FIFO1 is full	RW1C
b5	RF1W	Rx FIFO1 reaches the water level	Rx FIFO 1 Watermark Reached 0: The fill level of Rx FIFO1 is below the watermark 1: The fill level of Rx FIFO1 reaches the watermark	RW1C
b4	RF1N	Receive new message from FIFO1	Rx FIFO1 New Message (Rx FIFO 1 New Message) 0: No new message is written to Rx FIFO1 1: A new message is written into Rx FIFO1	RW1C
b3	RF0L	Rx FIFO0 message lost	Rx FIFO 0 Message Lost 0: Rx FIFO0 message is not lost 1: The Rx FIFO0 message is lost; this bit is also set when writing to the Rx FIFO0 with a space size of 0	RW1C
b2	RF0F	Rx FIFO0 is full	Rx FIFO0 is full (Rx FIFO 0 Full) 0: Rx FIFO0 is not full 1: Rx FIFO0 is full	RW1C
b1	RF0W	Rx FIFO0 reaches the	Rx FIFO 0 Watermark Reached	RW1C

		water level	0: The fill level of Rx FIFO0 is below the watermark 1: The fill level of Rx FIFO0 reaches the watermark	
b0	RF0N	Receive new message from FIFO0	Rx FIFO0 New Message (Rx FIFO 0 New Message) 0: No new message is written to Rx FIFO0 1: A new message is written into Rx FIFO0	RW1C

30.4.15 MCAN Interrupt Enable Register (MCAN_IE)

MCAN Interrupt Enable Register

Reset value: 0x0000 0000

The settings in the interrupt enable register determine which status changes in the interrupt register (MCAN_IR) will be indicated on the interrupt line.

b31	b30	b29	b28	b27	b26	b25	b24
Res		ARAE	PEDE	PEAE	WDIE	BOE	EWE
b23	b22	b21	b20	b19	b18	b17	b16
EPE	ELOE	BEUE	BECE	DRXE	TOOE	MRAFE	TSWE
b15	b14	b13	b12	b11	b10	b9	b8
TEFLE	TEFFE	TEFWE	TEFNE	TFEE	TCFE	TCE	HPME
b7	b6	b5	b4	b3	b2	b1	b0
RF1LE	RF1FE	RF1WE	RF1NE	RF0LE	RF0FE	RF0WE	RF0NE

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b30	Res	Reserved Bit	When reading, it is 0. When writing, please write 0	R
b29	ARAE	Access reserved address interrupt enable	Access to Reserved Address Enable 0: disable interrupt 1: enable interrupt	RW
b28	PEDE	Data phase protocol error interrupt enable	Protocol Error in Data Phase Enable 0: disable interrupt 1: enable interrupt	RW
b27	PEAE	Arbitration phase protocol error interrupt enable	Protocol Error in Arbitration Phase Enable 0: disable interrupt 1: enable interrupt	RW
b26	WDIE	Watchdog interrupt enable	Watchdog Interrupt Enable 0: disable interrupt 1: enable interrupt	RW
b25	BOE	Bus_Off Status Interrupt Enable	Bus_Off Status Interrupt Enable 0: disable interrupt 1: enable interrupt	RW
b24	EWE	Warning Status Interrupt Enable	Warning Status Interrupt Enable 0: disable interrupt 1: enable interrupt	RW
b23	EPE	Error Passive Interrupt Enable	Error Passive Interrupt Enable 0: disable interrupt 1: enable interrupt	RW
b22	ELOE	Error Logging Overflow Interrupt Enable	Error Logging Overflow Interrupt Enable 0: disable interrupt 1: enable interrupt	RW
b21	BEUE	Bit Error Uncorrected Interrupt Enable	Bit Error Uncorrected Interrupt Enable 0: disable interrupt 1: enable interrupt	RW
b20	BECE	Bit Error Corrected Interrupt Enable	Bit Error Corrected Interrupt Enable 0: disable interrupt 1: enable interrupt	
b19	DRXE	Message storage to dedicated Rx Buffer interrupt enable	Message stored to Dedicated Rx Buffer Interrupt Enable 0: disable interrupt 1: enable interrupt	RW
b18	TOOE	Timeout Occurred Interrupt Enable	Timeout Occurred Interrupt Enable 0: disable interrupt 1: enable interrupt	RW
b17	MRAFE	Message RAM Access Failure Interrupt Enable	Message RAM Access Failure Interrupt Enable 0: disable interrupt 1: enable interrupt	RW
b16	TSWE	Timestamp Wraparound Interrupt Enable	Timestamp Wraparound Interrupt Enable 0: disable interrupt 1: enable interrupt	RW

b15	TEFLE	Tx Event FIFO Element Lost Interrupt Enable	Tx Event FIFO Element Lost Interrupt Enable 0: disable interrupt 1: enable interrupt	RW
b14	TEFFE	Tx Event FIFO Full Interrupt Enable	Tx Event FIFO Full Interrupt Enable 0: disable interrupt 1: enable interrupt	RW
b13	TEFWE	Tx Event FIFO reaches water level interrupt enable	Tx Event FIFO Watermark Reached Interrupt Enable 0: disable interrupt 1: enable interrupt	RW
b12	TEFNE	Tx Event FIFO New Entry Interrupt Enable	Tx Event FIFO New Entry Interrupt Enable 0: disable interrupt 1: enable interrupt	RW
b11	TFEE	Tx FIFO null Interrupt Enable	Tx FIFO Empty Interrupt Enable 0: disable interrupt 1: enable interrupt	RW
b10	TCFE	Send cancel completed interrupt enable	Transmission Cancellation Finished Interrupt Enable 0: disable interrupt 1: enable interrupt	RW
b9	TCE	Transmit complete interrupt enable	Transmission Completed Interrupt Enable 0: disable interrupt 1: enable interrupt	RW
b8	HPME	High Priority Message Interrupt Enable	High Priority Message Interrupt Enable 0: disable interrupt 1: enable interrupt	RW
b7	RF1LE	Rx FIFO 1 Message Lost Interrupt Enable	Rx FIFO 1 Message Lost Interrupt Enable 0: disable interrupt 1: enable interrupt	RW
b6	RF1FE	Rx FIFO 1 Full Interrupt Enable	Rx FIFO 1 Full Interrupt Enable 0: disable interrupt 1: enable interrupt	RW
b5	RF1WE	Rx FIFO 1 Watermark Reached Interrupt Enable	Rx FIFO 1 Watermark Reached Interrupt Enable 0: disable interrupt 1: enable interrupt	RW
b4	RF1NE	Rx FIFO 1 New Message Interrupt Enable	Rx FIFO 1 New Message Interrupt Enable 0: disable interrupt 1: enable interrupt	RW
b3	RF0LE	Rx FIFO 0 Message Lost Interrupt Enable	Rx FIFO 0 Message Lost Interrupt Enable 0: disable interrupt 1: enable interrupt	RW
b2	RF0FE	Rx FIFO 0 Full Interrupt Enable	Rx FIFO 0 Full Interrupt Enable 0: disable interrupt 1: enable interrupt	RW
b1	RF0WE	Rx FIFO 0 Watermark Reached Interrupt Enable	Rx FIFO 0 Watermark Reached Interrupt Enable 0: disable interrupt 1: enable interrupt	RW
b0	RF0NE	Rx FIFO 0 New Message Interrupt Enable	Rx FIFO 0 New Message Interrupt Enable 0: disable interrupt 1: enable interrupt	RW

30.4.16 MCAN Interrupt Line Selection Register (MCAN_ILS)

MCAN Interrupt Line Select Register

Reset value: 0x0000 0000

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.

b31	b30	b29	b28	b27	b26	b25	b24
Res		ARAL	PEDL	PEAL	WDIL	BOL	EWL
b23	b22	b21	b20	b19	b18	b17	b16
EPL	ELOL	BEUL	BECL	DRXL	TOOL	MRAFL	TSWL
b15	b14	b13	b12	b11	b10	b9	b8
TEFLL	TEFFL	TEFWL	TEFNL	TFEL	TCFL	TCL	HPML
b7	b6	b5	b4	b3	b2	b1	b0
RF1LL	RF1FL	RF1WL	RF1NL	RF0LL	RF0FL	RF0WL	RF0NL

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b30	Res	Reserved Bit	When reading, it is 0. When writing, please write 0	R
b29	ARAL	Access to Reserved Address Line	Access to Reserved Address Line 0: Select interrupt line 0 1: Select interrupt line 1	RW
b28	PEDL	Protocol Error in Data Phase Line	Protocol Error in Data Phase Line 0: Select interrupt line 0 1: Select interrupt line 1	RW
b27	PEAL	Protocol Error in Arbitration Phase Line	Protocol Error in Arbitration Phase Line 0: Select interrupt line 0 1: Select interrupt line 1	RW
b26	WDIL	Watchdog Interrupt Line	Watchdog Interrupt Line 0: Select interrupt line 0 1: Select interrupt line 1	RW
b25	BOL	Bus_Off Status Interrupt Line	Bus_Off Status Interrupt Line 0: Select interrupt line 0 1: Select interrupt line 1	RW
b24	EWL	Warning Status Interrupt Line	Warning Status Interrupt Line 0: Select interrupt line 0 1: Select interrupt line 1	RW
b23	EPL	Error Passive Interrupt Line	Error Passive Interrupt Line 0: Select interrupt line 0 1: Select interrupt line 1	RW
b22	ELOL	Error logging interrupt line	Error Logging Overflow Interrupt Line 0: Select interrupt line 0 1: Select interrupt line 1	RW
b21	BEUL	Bit Error Uncorrected Interrupt Enable	Bit Error Uncorrected Interrupt Enable 0: Disable interrupt 1: Enable interrupt	RW
b20	BECL	Bit Error Corrected Interrupt Enable	Bit Error Corrected Interrupt Enable 0: Disable interrupt 1: Enable interrupt	
b19	DRXL	Message stored to Dedicated Rx Buffer Interrupt Line	Message stored to Dedicated Rx Buffer Interrupt Line 0: Select interrupt line 0 1: Select interrupt line 1	RW
b18	TOOL	Timeout Occurred Interrupt Line	Timeout Occurred Interrupt Line 0: Select interrupt line 0 1: Select interrupt line 1	RW
b17	MRAFL	Message RAM Access Failure Interrupt Line	Message RAM Access Failure Interrupt Line 0: Select interrupt line 0 1: Select interrupt line 1	RW

b16	TSWL	Timestamp Wraparound Interrupt Line	Timestamp Wraparound Interrupt Line 0: Select interrupt line 0 1: Select interrupt line 1	RW
b15	TEFLL	Tx Event FIFO Element Lost Interrupt Line	Tx Event FIFO Element Lost Interrupt Line 0: Select interrupt line 0 1: Select interrupt line 1	RW
b14	TEFFL	Tx Event FIFO Full Interrupt Line	Tx Event FIFO Full Interrupt Line 0: Select interrupt line 0 1: Select interrupt line 1	RW
b13	TEFWL	Tx Event FIFO Watermark Reached Interrupt Line	Tx Event FIFO Watermark Reached Interrupt Line 0: Select interrupt line 0 1: Select interrupt line 1	RW
b12	TEFNL	Tx Event FIFO New Entry Interrupt Line	Tx Event FIFO New Entry Interrupt Line 0: Select interrupt line 0 1: Select interrupt line 1	RW
b11	TFEL	Tx FIFO null Interrupt Line	Tx FIFO Empty Interrupt Line 0: Select interrupt line 0 1: Select interrupt line 1	RW
b10	TCFL	Transmission Cancellation Finished Interrupt Line	Transmission Cancellation Finished Interrupt Line 0: Select interrupt line 0 1: Select interrupt line 1	RW
b9	TCL	Transmission Completed Interrupt Line	Transmission Completed Interrupt Line 0: Select interrupt line 0 1: Select interrupt line 1	RW
b8	HPML	High Priority Message Interrupt Line	High Priority Message Interrupt Line 0: Select interrupt line 0 1: Select interrupt line 1	RW
b7	RF1LL	Rx FIFO 1 Message Lost Interrupt Line	Rx FIFO 1 Message Lost Interrupt Line 0: Select interrupt line 0 1: Select interrupt line 1	RW
b6	RF1FL	Rx FIFO 1 Full Interrupt Line	Rx FIFO 1 Full Interrupt Line 0: Select interrupt line 0 1: Select interrupt line 1	RW
b5	RF1WL	Rx FIFO 1 Watermark Reached Interrupt Line	Rx FIFO 1 Watermark Reached Interrupt Line 0: Select interrupt line 0 1: Select interrupt line 1	RW
b4	RF1NL	(Rx FIFO 1 New Message Interrupt Line	(Rx FIFO 1 New Message Interrupt Line 0: Select interrupt line 0 1: Select interrupt line 1	RW
b3	RF0LL	Rx FIFO 0 Message Lost Interrupt Line	Rx FIFO 0 Message Lost Interrupt Line 0: Select interrupt line 0 1: Select interrupt line 1	RW
b2	RF0FL	Rx FIFO 0 Full Interrupt Line	Rx FIFO 0 Full Interrupt Line 0: Select interrupt line 0 1: Select interrupt line 1	RW
b1	RF0WL	Rx FIFO 0 Watermark Reached Interrupt Line	Rx FIFO 0 Watermark Reached Interrupt Line 0: Select interrupt line 0 1: Select interrupt line 1	RW
b0	RF0NL	Rx FIFO 0 New Message Interrupt Line	Rx FIFO 0 New Message Interrupt Line 0: Select interrupt line 0 1: Select interrupt line 1	RW

30.4.17 MCAN Interrupt Line Enable Register (MCAN_ILE)

MCAN Interrupt Line Enable Register

Reset value: 0x0000 0000

Each of the two interrupt lines connected to the CPU can be enabled or disabled by setting EINT0 and EINT1 respectively.

b31	b30	b29	b28	b27	b26	b25	b24
Res							
b23	b22	b21	b20	b19	b18	b17	b16
Res							
b15	b14	b13	b12	b11	b10	b9	b8
Res							
b7	b6	b5	b4	b3	b2	b1	b0
Res						EINT1	EINT0

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b2	Res	Reserved Bit	When reading, it is 0. When writing, please write 0	R
b1	EINT1	Interrupt line 1 enable	0: Interrupt line 1 disable 1: Interrupt line 1 enable	RW
b0	EINT0	Interrupt line 0 enable	0: Interrupt line 0 disable 1: Interrupt line 0 enable	RW

30.4.18 MCAN Global Filter Configuration Register (MCAN_GFC)

MCAN Global Filter Configuration Register

Reset value: 0x0000 0000

Global settings for message ID filtering. Global filter configuration, controls the filtering path of standard messages and extended messages, see Figure 30-5 andFigure 30-6 for details.

b31	b30	b29	b28	b27	b26	b25	b24
Res							
b23	b22	b21	b20	b19	b18	b17	b16
Res							
b15	b14	b13	b12	b11	b10	b9	b8
Res							
b7	b6	b5	b4	b3	b2	b1	b0
Res		ANFS[1:0]		ANFE[1:0]		RRFS	RRFE

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b6	Res	Reserved Bit	When reading, it is 0. When writing, please write 0	R
b5-b4	ANFS[1:0]	Accept non-matching standard frames	Defines how to handle received 11-bit standard ID messages that do not match any element of the filter list. 00: Accept mismatched standard frames in Rx FIFO0 01: Unmatched standard frame received in Rx FIFO1 10: Reject non-matching standard frames 11: Reject non-matching standard frames	RWA
b3~b2	ANFE[1:0]	Accept unmatched extended frames	Defines how to handle received messages with a 29-bit extension ID that does not match any element of the filter list. 00: Accept mismatched extended frame in Rx FIFO0 01: Unmatched extended frame accepted in Rx FIFO1 10: Reject unmatched extended frames 11: Reject unmatched extended frames	RWA
b1	RRFS	Reject Remote Standard Frame	0: Filter remote frames with 11-bit standard ID 1: reject all remote frames with 11-bit standard ID	RWA
b0	RRFE	Reject remote extension frames	0: Filter remote frames with 29-bit extended ID 1: reject all remote frames with 29-bit extension ID	RWA

30.4.19 MCAN Standard ID Filter Configuration Register (MCAN_SIDFC)

MCAN Standard ID Filter Configuration Register

Reset value: 0x0000 0000

Settings for 11-bit standard ID filter. Standard ID filter configuration, controls the filtering path of standard messages, see Figure 30-5.

b31	b30	b29	b28	b27	b26	b25	b24
Res							
b23	b22	b21	b20	b19	b18	b17	b16
LSS[7:0]							
b15	b14	b13	b12	b11	b10	b9	b8
FLSSA[15:8]							
b7	b6	b5	b4	b3	b2	b1	b0
FLSSA[7:2]						Res	

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b24	Res	Reserved Bit	When reading, it is 0. When writing, please write 0	R
b23~b16	LSS[7:0]	Standard ID filter list size	0: No standard ID filter 1~128: Number of standard ID filter elements >128: The set value greater than 128 is interpreted as 128	RWA
b15~b2	FLSSA[15:2]	Standard ID filter list start address	Set the start address of the standard ID filter list (32-bit word address, seeFigure 30-11)	RWA
b1~b0	Res	Reserved Bit	When reading, it is 0. When writing, please write 0	R

30.4.20 MCAN Extended ID Filter Configuration Register (MCAN_XIDFC)

MCAN Extended ID Filter Configuration Register

Reset value: 0x0000 0000

Settings for 29-bit extended ID filter. Extension ID filter configuration controls the filtering path of extension messages, seeFigure 30-6.

b31	b30	b29	b28	b27	b26	b25	b24
Res							
b23	b22	b21	b20	b19	b18	b17	b16
Res	LSE[6:0]						
b15	b14	b13	b12	b11	b10	b9	b8
FLESA[15:8]							
b7	b6	b5	b4	b3	b2	b1	b0
FLESA[7:2]						Res	

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b23	Res	Reserved Bit	When reading, it is 0. When writing, please write 0	R
b22~b16	LES[7:0]	Extended ID filter list size	0: No extended ID filter 1~64: Number of extended ID filter elements >64: A setting value greater than 64 is interpreted as 64	RWA
b15~b2	FLESA[15:2]	Extended ID filter list start address	The starting address of the extended ID filter list (32-bit word address, see Figure 30-11)	RWA
b1~b0	Res	Reserved Bit	When reading, it is 0. When writing, please write 0	R

30.4.21 MCAN Extended ID AND Mask Register (MCAN_XIDAM)

MCAN Extended ID AND Mask Register

Reset value: 0x1FFF FFFF

b31	b30	b29	b28	b27	b26	b25	b24
Res			EIDM[28:24]				
b23	b22	b21	b20	b19	b18	b17	b16
EIDM[23:16]							
b15	b14	b13	b12	b11	b10	b9	b8
EIDM[15:8]							
b7	b6	b5	b4	b3	b2	b1	b0
EIDM[7:0]							

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b27	Res	Reserved Bit	When reading, it is 0. When writing, please write 0	R
b28~b0	EIDM[28:0]	Extended ID Mask	For extended frame reception filtering, the 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 invalid.	RWA

30.4.22 MCAN High Priority Message Status Register (MCAN_HPMS)

MCAN High Priority Message Status Register

Reset value: 0x0000 0000

Message ID filter elements that are configured to generate priority events will update this register on each match. This can be used to monitor the status of incoming high priority messages and provide quick access to them.

b31	b30	b29	b28	b27	b26	b25	b24
Res							
b23	b22	b21	b20	b19	b18	b17	b16
Res							
b15	b14	b13	b12	b11	b10	b9	b8
FLST	FIDX[6:0]						
b7	b6	b5	b4	b3	b2	b1	b0
MSI[1:0]		BIDX[5:0]					

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b16	Res	Reserved Bit	When reading, it is 0. When writing, please write 0	R
b15	FLST	Filter List	Indicates the list of filters that matched. 0: Standard filter list 1: Extend the filter list	R
b14~b8	FIDX[6:0]	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.	R
b7~b6	MSI[1:0]	Message memory indicator	00: FIFO not selected 01: FIFO message lost 10: The message is stored in FIFO0 11: Message stored in FIFO1	R
b5~b0	BIDX[5:0]	Buffer Index	Index of the Rx FIFO element where the message is stored. Only valid when MSI[1]=1.	R

30.4.23 MCAN New Data 1 Register (MCAN_NDAT1)

MCAN New Data 1 Register

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24
ND31	ND30	ND29	ND28	ND27	ND26	ND25	ND24
b23	b22	b21	b20	b19	b18	b17	b16
ND23	ND22	ND21	ND20	ND19	ND18	ND17	ND16
b15	b14	b13	b12	b11	b10	b9	b8
ND15	ND14	ND13	ND12	ND11	ND10	ND9	ND8
b7	b6	b5	b4	b3	b2	b1	b0
ND7	ND6	ND5	ND4	ND3	ND2	ND1	ND0

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b0	ND31~0	New data	This register holds the new data flag for Rx Buffer s 0 to 31. When a receive frame updates the Rx Buffer , the corresponding flag bit is set to 1. The flag bit remains set until it is cleared by the master. A flag bit is cleared by writing 1 to the corresponding bit; writing 0 has no effect. A hard reset clears the registers to zero. 0: The Rx Buffer is not updated 1: The Rx Buffer is updated with new messages	RW

30.4.24 MCAN New Data 2 Register (MCAN_NDAT2)

MCAN New Data 2 Register

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24
ND63	ND62	ND61	ND60	ND59	ND58	ND57	ND56
b23	b22	b21	b20	b19	b18	b17	b16
ND55	ND54	ND53	ND52	ND51	ND50	ND49	ND48
b15	b14	b13	b12	b11	b10	b9	b8
ND47	ND46	ND45	ND44	ND43	ND42	ND41	ND40
b7	b6	b5	b4	b3	b2	b1	b0
ND39	ND38	ND37	ND36	ND35	ND34	ND33	ND32

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b0	ND63~32	New data	This register holds the new data flag for Rx Buffer s 32 to 63. When a receive frame updates the Rx Buffer , the corresponding flag bit is set to 1. The flag bit remains set until it is cleared by the master. A flag bit is cleared by writing 1 to the corresponding bit; writing 0 has no effect. A hard reset clears the registers to zero. 0: The Rx Buffer is not updated 1: The Rx Buffer is updated with new messages	RW

30.4.25 MCAN Rx FIFO0 Configuration Register (MCAN_RXF0C)

MCAN Rx FIFO 0 Configuration Register

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24
F0OM	F0WM[6:0]						
b23	b22	b21	b20	b19	b18	b17	b16
Res	F0S[6:0]						
b15	b14	b13	b12	b11	b10	b9	b8
F0SA[15:8]							
b7	b6	b5	b4	b3	b2	b1	b0
F0SA[7:2]						Res	

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31	F0OM	Rx FIFO0 Working Mode	FIFO0 can work in blocking mode or overwrite mode. 0: Rx FIFO0 is in blocking mode 1: Rx FIFO0 is in overwrite mode	RWA
b30~b24	F0WM[6:0]	Rx FIFO0 water level	0: Disable Rx FIFO0 water level interrupt 1~64: Level of Rx FIFO0 waterline interrupt (MCAN_IR.RF0W) >64: Disable Rx FIFO0 water level interrupt	RWA
b23	Res	Reserved Bit	When reading, it is 0. When writing, please write 0	R
b22~b16	F0S[6:0]	Rx FIFO0 size	0: No Rx FIFO0 1~64: Number of Rx FIFO0 elements >64: A setting value greater than 64 is interpreted as 64 The Rx FIFO0 elements are indexed from 0 to F0S minus 1.	RWA
b15~b2	F0SA[15:2]	Rx FIFO0 starting address	The starting address of Rx FIFO0 in the message RAM (32-bit word address, seeFigure 30-11)	RWA
b1~b0	Res	Reserved Bit	When reading, it is 0. When writing, please write 0	R

30.4.26 MCAN Rx FIFO0 Status Register (MCAN_RXF0S)

MCAN Rx FIFO 0 Status Register

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24
Res						RF0L	F0F
b23	b22	b21	b20	b19	b18	b17	b16
Res		F0PI[5:0]					
b15	b14	b13	b12	b11	b10	b9	b8
Res		F0GI[5:0]					
b7	b6	b5	b4	b3	b2	b1	b0
Res	F0FL[6:0]						

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b26	Res	Reserved Bit	When reading, it is 0. When writing, please write 0	R
b25	RF0L	Rx 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 0: No message is lost in receiving FIFO0 1: Rx FIFO0 has a message lost; this bit is also set when writing to the Rx FIFO0 with a space size of 0 **Note:** When MCAN_RXF0C.F0OM=1, the earliest message is overwritten and this flag is not set.	R
b24	F0F	Rx FIFO0 full	0: Rx FIFO0 is not full 1: Rx FIFO0 is full	R
b23~b22	Res	Reserved Bit	When reading, it is 0. When writing, please write 0	R
b15~b8	F0PI[5:0]	Receive FIFO0 put in index	Rx FIFO0 write index pointer, range 0~63	R
b15~b14	Res	Reserved Bit	When reading, it is 0. When writing, please write 0	R
b13~b8	F0GI[5:0]	Rx FIFO0 to get the index	Rx FIFO0 read index pointer, range 0~63	R
b7	Res	Reserved Bit	When reading, it is 0. When writing, please write 0	R
b6~b0	F0FL[6:0]	Rx FIFO0 fill level	The number of elements stored in Rx FIFO0, ranging from 0 to 64	R

30.4.27 MCAN Rx FIFO0 Acknowledge Register (MCAN_RXF0A)

MCAN Rx FIFO 0 Acknowledge Register

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24
Res							
b23	b22	b21	b20	b19	b18	b17	b16
Res							
b15	b14	b13	b12	b11	b10	b9	b8
Res							
b7	b6	b5	b4	b3	b2	b1	b0
Res				F0AI[5:0]			

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b6	Res	Reserved Bit	When reading, it is 0. When writing, please write 0	R
b5~b0	F0AI[5:0]	Rx FIFO0 acknowledgement index	After the master has read a message or sequence of messages from Rx FIFO0, it must write the buffer index of the last element read from Rx FIFO0 to F0AI. This operation sets the Rx FIFO0 get index MCAN_RXF0S.F0GI to F0AI plus 1 and updates the FIFO0 fill level MCAN_RXF0S.F0FL	RW

30.4.28 MCAN Rx Buffer Configuration Register (MCAN_RXBC)

MCAN Rx Buffer Configuration Register

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24
Res							
b23	b22	b21	b20	b19	b18	b17	b16
Res							
b15	b14	b13	b12	b11	b10	b9	b8
RBSA[15:8]							
b7	b6	b5	b4	b3	b2	b1	b0
RBSA[7:2]						Res	

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b16	Res	Reserved Bit	When reading, it is 0. When writing, please write 0	R
b15~b2	RBSA[15:2]	The starting address of the Rx Buffer	Configure the starting address of the Rx Buffer in the message RAM (32-bit address), also used to reference debug messages A, B, C	RWA
b1~b0	Res	Reserved Bit	When reading, it is 0. When writing, please write 0	R

30.4.29 MCAN Rx FIFO1 Configuration Register (MCAN_RXF1C)

MCAN Rx FIFO 1 Configuration Register

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24
F1OM	F1WM[6:0]						
b23	b22	b21	b20	b19	b18	b17	b16
Res	F1S[6:0]						
b15	b14	b13	b12	b11	b10	b9	b8
F1SA[15:8]							
b7	b6	b5	b4	b3	b2	b1	b0
F1SA[7:2]						Res	

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31	F1OM	Rx FIFO1 Operation Mode	FIFO1 can work in blocking mode or overwrite mode. 0: Rx FIFO1 is in blocking mode 1: Rx FIFO1 is in overwrite mode	RWA
b30~b24	F1WM[6:0]	Rx FIFO1 watermark	0: Disable Rx FIFO1 water level interrupt 1~64: Level of Rx FIFO1 waterline interrupt (MCAN_IR.RF1W) >64: Disable Rx FIFO1 water level interrupt	RWA
b23	Res	Reserved Bit	When reading, it is 0. When writing, please write 0	R
b22~b16	F1S[6:0]	Rx FIFO1 size	0: No Rx FIFO1 1~64: Number of Rx FIFO1 elements >64: A setting value greater than 64 is interpreted as 64 The Rx FIFO1 elements are indexed from 0 to F1S minus 1.	RWA
b15~b2	F1SA[15:2]	Rx FIFO1 start address	The starting address of Rx FIFO1 in the message RAM (32-bit word address, seeFigure 30-11)	RWA
b1~b0	Res	Reserved Bit	When reading, it is 0. When writing, please write 0	R

30.4.30 MCAN Rx FIFO1 Status Register (MCAN_RXF1S)

MCAN Rx FIFO 1 Status Register

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24
DMS[1:0]		Res				RF1L	F1F
b23	b22	b21	b20	b19	b18	b17	b16
Res		F1PI[5:0]					
b15	b14	b13	b12	b11	b10	b9	b8
Res		F1GI[5:0]					
b7	b6	b5	b4	b3	b2	b1	b0
Res	F1FL[6:0]						

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b30	DMS[1:0]	Debug message status	00: Idle state, waiting to receive debug messages 01: Debug message A received 10: Debug messages A and B received 11: Debug messages A, B, C received	R
b29~b28	Res	Reserved Bit	When reading, it is 0. When writing, please write 0	R
b25	RF1L	Rx FIFO1 message lost	0: No elements are lost in Rx FIFO1 1: An element of Rx FIFO1 is lost; this bit is also set when a write operation is performed on Rx FIFO1 with a space size of 0 **Note:** When MCAN_RXF1C.F1OM=1, the earliest message is overwritten and this flag is not set.	R
b24	F1F	Rx FIFO1 full	0: Rx FIFO1 is not full 1: Rx FIFO1 is full	R
b23~b22	Res	Reserved Bit	When reading, it is 0. When writing, please write 0	R
b15-b8	F1PI[5:0]	Rx FIFO1 puts index	Rx FIFO1 write index pointer, range 0~63	R
b15~b14	Res	Reserved Bit	When reading, it is 0. When writing, please write 0	R
b13~b8	F1GI[5:0]	Rx FIFO1 get index	Rx FIFO1 read index pointer, range 0~63	R
b7	Res	Reserved Bit	When reading, it is 0. When writing, please write 0	R
b6~b0	F1FL[6:0]	Rx FIFO1 fill level	The number of elements stored in Rx FIFO1, ranging from 0 to 64	R

30.4.31 MCAN Rx FIFO1 Acknowledge Register (MCAN_RXF1A)

MCAN Rx FIFO 1 Acknowledge Register

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24
Res							
b23	b22	b21	b20	b19	b18	b17	b16
Res							
b15	b14	b13	b12	b11	b10	b9	b8
Res							
b7	b6	b5	b4	b3	b2	b1	b0
Res		F1AI[5:0]					

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b6	Res	Reserved Bit	When reading, it is 0. When writing, please write 0	R
b5~b0	F1AI[5:0]	Rx FIFO1 acknowledgement index	After the master has read a message or sequence of messages from Rx FIFO1, it must write the buffer index of the last element read from Rx FIFO1 to F1AI. This operation sets the Rx FIFO1 get index MCAN_RXF1S.F1GI to F1AI plus 1 and updates the FIFO1 fill level MCAN_RXF1S.F1FL	RW

30.4.32 MCAN Rx Buffer and FIFO Element Size Configuration Register (MCAN_RXESC)

MCAN Rx Buffer/ FIFO Element Size Configuration Register

Reset value: 0x0000 0000

Configure the Rx Buffer and the number of data bytes for the Rx FIFO elements. Data fields larger than 8 bytes are only applicable in CAN FD mode.

b31	b30	b29	b28	b27	b26	b25	b24
Res							
b23	b22	b21	b20	b19	b18	b17	b16
Res							
b15	b14	b13	b12	b11	b10	b9	b8
Res					RBDS[2:0]		
b7	b6	b5	b4	b3	b2	b1	b0
Res	F1DS[2:0]			Res	F0DS[2:0]		

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b11	Res	Reserved Bit	When reading, it is 0. When writing, please write 0	R
b10~b8	RBDS[2:0]	Rx Buffer data field size	000: 8 bytes 001: 12 bytes 010: 16 bytes 011: 20 bytes 100: 24 bytes 101: 32 bytes 110: 48 bytes 111: 64 bytes	RWA
b7	Res	Reserved Bit	When reading, it is 0. When writing, please write 0	R
b6~b4	F1DS[2:0]	Rx FIFO1 field size	000: 8 bytes 001: 12 bytes 010: 16 bytes 011: 20 bytes 100: 24 bytes 101: 32 bytes 110: 48 bytes 111: 64 bytes	RWA
b3	Res	Reserved Bit	When reading, it is 0. When writing, please write 0	R
b2~b0	F0DS[2:0]	Rx FIFO0 data field size	000: 8 bytes 001: 12 bytes 010: 16 bytes 011: 20 bytes 100: 24 bytes 101: 32 bytes 110: 48 bytes 111: 64 bytes	RWA

30.4.33 MCAN Tx Buffer Configuration Register (MCAN_TXBC)

Reset value: 0x0000 0000

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.

b31	b30	b29	b28	b27	b26	b25	b24
Res	TFQM	TFQS[5:0]					
b23	b22	b21	b20	b19	b18	b17	b16
Res		NDTB[5:0]					
b15	b14	b13	b12	b11	b10	b9	b8
TBSA[13:6]							
b7	b6	b5	b4	b3	b2	b1	b0
TBSA[5:0]						Res	

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31	Res	Reserved Bit	When reading, it is 0. When writing, please write 0	R
b30	TFQM	Tx FIFO/Queue Mode	0: Tx FIFO mode 1: Send queue mode	RW
b29~b24	TFQS[5:0]	Tx FIFO/queue size	0: No Tx FIFO or queue 1-32: Buffer size of the Tx FIFO queue >32: A setting value greater than 32 is interpreted as 32	RW
b23~b22	Res	Reserved Bit	When reading, it is 0. When writing, please write 0	R
b21~b16	NDTB[5:0]	Dedicated Tx Buffer size	0: No dedicated Tx Buffer 1-32: dedicated Tx Buffer size >32: A setting value greater than 32 is interpreted as 32	RW
b15~b2	TBSA[13:0]	Tx Buffer start address	The starting address of the Tx Buffer portion in the message RAM (32-bit word address, see Figure 30-11)	RW
b1~b0	Res	Reserved Bit	When reading, it is 0. When writing, please write 0	R

Note:

The sum of TFQS and NDTB shall not be greater than 32. MCAN does not monitor this configuration error. The Tx Buffer portion of the message RAM starts with a dedicated Tx Buffer.

30.4.34 MCAN Tx FIFO/Queue Status Register (MCAN_TXFQS)

MCAN Tx FIFO/Queue Status Register

Reset value: 0x0000 0000

The status of the Tx FIFO or queue is related to the pending send request in register MCAN_TXBRP. Therefore, the effect of adding or canceling a request may be delayed due to a send scan that is running (TXBRP has not yet been updated).

b31	b30	b29	b28	b27	b26	b25	b24
Res							
b23	b22	b21	b20	b19	b18	b17	b16
Res		TFQF	TFQPI[4:0]				
b15	b14	b13	b12	b11	b10	b9	b8
Res			TFGI[4:0]				
b7	b6	b5	b4	b3	b2	b1	b0
Res		TFFL[5:0]					

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b22	Res	Reserved Bit	When reading, it is 0. When writing, please write 0	R
b21	TFQF	Tx FIFO/queue full	0: The Tx FIFO or queue is not full 1: The Tx FIFO or queue is full	R
b20~b16	TFQPI[4:0]	Tx FIFO/queue put index	Tx FIFO or queue index pointer, range 0-31	R
b15~b13	Res	Reserved Bit	When reading, it is 0. When writing, please write 0	R
b12~b8	TFGI[4:0]	Tx FIFO Get Index	The read index pointer of the Tx FIFO, ranging from 0 to 31. If it is configured as send queue mode (MCAN_TXBC.TFQM=1), the value of this bit is 0.	R
b7~b6	Res	Reserved Bit	When reading, it is 0. When writing, please write 0	R
b5~b0	TFFL[5:0]	Tx FIFO idle level	The number of consecutive idle Tx FIFO elements starting from TFGI, in the range 0-32. If it is configured as send queue mode (MCAN_TXBC.TFQM=1), the value of this bit is 0.	R

Note:

In the case where a dedicated Tx Buffer is configured in combination with a Tx FIFO or a send queue, the put index and get index refer to the Tx Buffer number starting from the first dedicated Tx Buffer. For example, for a combined configuration with 12 dedicated Tx Buffers and 20 Tx FIFO buffers, put index 15 to point to the 4th buffer of the Tx FIFO.

30.4.35 MCAN Tx Buffer Element Size Configuration Register (MCAN_TXESC)

MCAN Tx Buffer Element Size Configuration Register

Reset value: 0x0000 0000

Configure the number of data bytes for a Tx Buffer element. Data fields larger than 8 bytes are only applicable in CAN FD mode.

b31	b30	b29	b28	b27	b26	b25	b24
Res							
b23	b22	b21	b20	b19	b18	b17	b16
Res							
b15	b14	b13	b12	b11	b10	b9	b8
Res							
b7	b6	b5	b4	b3	b2	b1	b0
Res					TBDS[2:0]		

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b3	Res	Reserved Bit	When reading, it is 0. When writing, please write 0	R
b2~b1	TBDS[2:0]	Tx Buffer data field size	000: 8 bytes 001: 12 bytes 010: 16 bytes 011: 20 bytes 100: 24 bytes 101: 32 bytes 110: 48 bytes 111: 64 bytes	RW

Note:

If the data length of the Tx Buffer element (configured by DLC) is greater than the data length configured by TBDS, the undefined data bytes in the Tx Buffer will be sent as 0xCC (filling bytes).

30.4.36 MCAN Transmit Buffer Request Pending Register (MCAN_TXBRP)

MCAN Tx Buffer Request Pending Register

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24
TRP31	TRP30	TRP29	TRP28	TRP27	TRP26	TRP25	TRP24
b23	b22	b21	b20	b19	b18	b17	b16
TRP23	TRP22	TRP21	TRP20	TRP19	TRP18	TRP17	TRP16
b15	b14	b13	b12	b11	b10	b9	b8
TRP15	TRP14	TRP13	TRP12	TRP11	TRP10	TRP9	TRP8
b7	b6	b5	b4	b3	b2	b1	b0
TRP7	TRP6	TRP5	TRP4	TRP3	TRP2	TRP1	TRP0

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b0	TRP31~0	Send request pending flag	Each Tx Buffer has its own send request pending bit. These bits are set via register MCAN_TXBAR and reset after the requested transmission has been completed or the requested transmission has been canceled via register MCAN_TXBCR. Only the TXBRP bit corresponding to the Tx Buffer configured by MCAN_TXBC will be set. When the TXBRP bit is set, a send scan is started (see section [Send Processing]) to check the highest priority pending send request (i.e. the Tx Buffer with the smallest message ID). Canceling a send request resets the corresponding TXBRP bit. If a transmission has already started when a cancel is requested, the TXBRP bit will be reset at the end of the transmission regardless of whether the transmission was successful or not. When the TXBRP bit is reset, the corresponding cancel request bit is also reset immediately. After a cancellation is requested, the following indicates that the cancellation is complete via register MCAN_TXBCF: • After the transmission is successful and the corresponding MCAN_TXBTO bit is set • Canceled when sending has not started • Send aborted due to arbitration failure • An error occurred during frame transmission In DAR mode, all sends are automatically canceled after failure. For all failed transmissions, the corresponding MCAN_TXBCF bit will be set. 0: No send request pending 1: Send request pending **Note:** The TXBRP bits set during a send scan are ignored during this send scan. During this period, if a request is made to cancel these transmissions, the transmissions are canceled immediately and the corresponding TXBRP bits are reset.	R

30.4.37 MCAN Tx Buffer Add Request Register (MCAN_TXBAR)

MCAN Tx Buffer Add Request Register

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24
AR31	AR30	AR29	AR28	AR27	AR26	AR25	AR24
b23	b22	b21	b20	b19	b18	b17	b16
AR23	AR22	AR21	AR20	AR19	AR18	AR17	AR16
b15	b14	b13	b12	b11	b10	b9	b8
AR15	AR14	AR13	AR12	AR11	AR10	AR9	AR8
b7	b6	b5	b4	b3	b2	b1	b0
AR7	AR6	AR5	AR4	AR3	AR2	AR1	AR0

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b0	AR31~0	Add Request	Each Tx Buffer has its own add request bit. Writing 1 sets the corresponding add request bit, and writing 0 has no effect. The master can set up send requests for multiple Tx Buffers at one time. Only the TXBAR bit corresponding to the Tx Buffer configured by MCAN_TXBC will be set. These bits are reset immediately when a send scan is not running, otherwise they remain set until the send scan is complete. 0: No send request is added 1: Send request has been added **Note:** If the Tx Buffer of the add request already has a send request pending (the corresponding bit of the MCAN_TXBRP register is set), the add request will be ignored.	RW

30.4.38 MCAN Tx Buffer Cancel Request Register (MCAN_TXBCR)

MCAN Tx Buffer Add Request Register

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24
CR31	CR30	CR29	CR28	CR27	CR26	CR25	CR24
b23	b22	b21	b20	b19	b18	b17	b16
CR23	CR22	CR21	CR20	CR19	CR18	CR17	CR16
b15	b14	b13	b12	b11	b10	b9	b8
CR15	CR14	CR13	CR12	CR11	CR10	CR9	CR8
b7	b6	b5	b4	b3	b2	b1	b0
CR7	CR6	CR5	CR4	CR3	CR2	CR1	CR0

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b0	CR31~0	Cancel Request	Each Tx Buffer has its own cancel request bit. Writing 1 sets the corresponding cancel request bit, and writing 0 has no effect. The master can set cancel requests for multiple Tx Buffers at one time. Only the TXBCR bit corresponding to the Tx Buffer configured by MCAN_TXBC will be set. These bits remain set until the corresponding MCAN_TXBRP bit is reset. 0: No cancellation request pending 1: A cancel request is pending	RW

30.4.39 MCAN Tx Buffer Send Occurred Register (MCAN_TXBTO)

MCAN Tx Buffer Transmission Occurred Register

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24
TO31	TO30	TO29	TO28	TO27	TO26	TO25	TO24
b23	b22	b21	b20	b19	b18	b17	b16
TO23	TO22	TO21	TO20	TO19	TO18	TO17	TO16
b15	b14	b13	b12	b11	b10	b9	b8
TO15	TO14	TO13	TO12	TO11	TO10	TO9	TO8
b7	b6	b5	b4	b3	b2	b1	b0
TO7	TO6	TO5	TO4	TO3	TO2	TO1	TO0

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b0	TO31~0	Sending has occurred	Each Tx Buffer has its own send-has-occurred flag. When the Tx Buffer is completed and the corresponding bit of MCAN_TXBRP is cleared, the corresponding send 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. 0: Not sent 1: Sent	R

30.4.40 MCAN Tx Buffer Cancel Completed Register (MCAN_TXBCF)

MCAN Tx Buffer Cancellation Finished Register

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24
CF31	CF30	CF29	CF28	CF27	CF26	CF25	CF24
b23	b22	b21	b20	b19	b18	b17	b16
CF23	CF22	CF21	CF20	CF19	CF18	CF17	CF16
b15	b14	b13	b12	b11	b10	b9	b8
CF15	CF14	CF13	CF12	CF11	CF10	CF9	CF8
b7	b6	b5	b4	b3	b2	b1	b0
CF7	CF6	CF5	CF4	CF3	CF2	CF1	CF0

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b0	CF31~0	Cancellation Completed	Each Tx Buffer has its own cancel completed flag. When the cancellation requested through MCAN_TXBCR is completed and the corresponding bit of MCAN_TXBRP is cleared, the corresponding cancellation completed flag is set. If the corresponding bit of MCAN_TXBRP is not set at the time of cancellation, the cancel completed flag will be set immediately. When writing 1 to the corresponding bit of MCAN_TXBAR to request a new transmission, the corresponding cancel completed flag is reset. 0: No Tx Buffer cancel 1: The Tx Buffer has been canceled	R

30.4.41 MCAN Tx Buffer Send Interrupt Enable Register (MCAN_TXBTIE)

MCAN Tx Buffer Transmission Interrupt Enable Register

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24
TIE31	TIE30	TIE29	TIE28	TIE27	TIE26	TIE25	TIE24
b23	b22	b21	b20	b19	b18	b17	b16
TIE23	TIE22	TIE21	TIE20	TIE19	TIE18	TIE17	TIE16
b15	b14	b13	b12	b11	b10	b9	b8
TIE15	TIE14	TIE13	TIE12	TIE11	TIE10	TIE9	TIE8
b7	b6	b5	b4	b3	b2	b1	b0
TIE7	TIE6	TIE5	TIE4	TIE3	TIE2	TIE1	TIE0

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b0	TIE31~0	Transmit interrupt enable	Each Tx Buffer has its own send Interrupt Enable Bit. 0: Disable sending interrupt 1: Enable send interrupt	RW

30.4.42 MCAN Tx Buffer Cancel Completed Interrupt Enable Register (MCAN_TXBCIE)

MCAN Tx Buffer Cancellation Finished Interrupt Enable Register

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24
CFIE31	CFIE30	CFIE29	CFIE28	CFIE27	CFIE26	CFIE25	CFIE24
b23	b22	b21	b20	b19	b18	b17	b16
CFIE23	CFIE22	CFIE21	CFIE20	CFIE19	CFIE18	CFIE17	CFIE16
b15	b14	b13	b12	b11	b10	b9	b8
CFIE15	CFIE14	CFIE13	CFIE12	CFIE11	CFIE10	CFIE9	CFIE8
b7	b6	b5	b4	b3	b2	b1	b0
CFIE7	CFIE6	CFIE5	CFIE4	CFIE3	CFIE2	CFIE1	CFIE0

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b0	CFIE31~0	Cancel the completed interrupt enable	Each Tx Buffer has its own cancel completed Interrupt Enable Bit. 0: Disable cancellation of completed interrupts 1: Enable cancel completed interrupt	RW

30.4.43 MCAN Tx Event FIFO Configuration Register (MCAN_TXEFC)

MCAN Tx Event FIFO Configuration Register

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24
Res		EFWM[5:0]					
b23	b22	b21	b20	b19	b18	b17	b16
Res		EFS[5:0]					
b15	b14	b13	b12	b11	b10	b9	b8
EFSA[15:8]							
b7	b6	b5	b4	b3	b2	b1	b0
EFSA[7:2]						Res	

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b30	Res	Reserved Bit	When reading, it is 0. When writing, please write 0	R
b29~b24	EFWM[5:0]	Event FIFO watermark	0: Disable waterline interruption 1~32: The level of the Tx Event FIFO waterline interrupt (MCAN_IR.TEFW) >32: water level interruption is prohibited	RWA
b23~b22	Res	Reserved Bit	When reading, it is 0. When writing, please write 0	R
b21~b16	EFS[5:0]	Event FIFO size	0: Disable Tx Event FIFO 1-32: Number of elements in the Tx Event FIFO >32: A setting value greater than 32 is interpreted as 32 The index of the Tx Event FIFO element is 0 to EFS minus 1	RWA
b15~b2	EFSA[15:2]	Event FIFO start address	Starting address of the event FIFO in the message RAM (32-bit word address, seeFigure 30-11)	RWA
b1~b0	Res	Reserved Bit	When reading, it is 0. When writing, please write 0	R

30.4.44 MCAN Tx Event FIFO Status Register (MCAN_TXEFS)

MCAN Tx Event FIFO Status Register

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24
Res						TEFL	EFF
b23	b22	b21	b20	b19	b18	b17	b16
Res			EFPI[4:0]				
b15	b14	b13	b12	b11	b10	b9	b8
Res			EFGI[4:0]				
b7	b6	b5	b4	b3	b2	b1	b0
Res		EFFL[5:0]					

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b26	Res	Reserved Bit	When reading, it is 0. When writing, please write 0	R
b25	TEFL	Tx 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 0: No elements are lost in the Tx Event FIFO 1: An element of the Tx Event FIFO is lost; this bit is also set when a write operation is performed on a Tx Event FIFO with a space size of 0.	R
b24	EFF	Event FIFO is full	0: The Tx Event FIFO is not full 1: The Tx Event FIFO is full	R
b23~b21	Res	Reserved Bit	When reading, it is 0. When writing, please write 0	R
b20~b16	EFPI[4:0]	Event FIFO puts index	Tx Event FIFO write index pointer, range 0~31	R
b15~b13	Res	Reserved Bit	When reading, it is 0. When writing, please write 0	R
b12~b8	EFGI[4:0]	Event FIFO Get Index	Tx Event FIFO read index pointer, range 0~31	R
b7~b6	Res	Reserved Bit	When reading, it is 0. When writing, please write 0	R
b5~b0	EFFL[5:0]	Event FIFO fill level	The number of elements stored in Rx FIFO0, ranging from 0 to 32	R

30.4.45 MCAN Tx Event FIFO Acknowledge Register (MCAN_TXEFA)

MCAN Tx Event FIFO Acknowledge Register

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24
Res							
b23	b22	b21	b20	b19	b18	b17	b16
Res							
b15	b14	b13	b12	b11	b10	b9	b8
Res							
b7	b6	b5	b4	b3	b2	b1	b0
Res			EFAI[4:0]				

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b5	Res	Reserved Bit	When reading, it is 0. When writing, please write 0	R
b4~b0	EFAI[4:0]	Event FIFO Acknowledgement Index	After the master reads an element or sequence of elements from the Tx Event FIFO, it must write the index of the last element read from the event FIFO to the EFAI. This operation sets the Tx Event FIFO get index MCAN_TXEFS.EFGI to EFAI plus 1 and updates the event FIFO fill level MCAN_TXEFS.EFFL.	RW

30.4.46 MCAN Message RAM Check Control Register (MCANRAM_CKCR)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res														ECCMOD[1:0]	

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b2	Res	Reserved Bit	Read as "0", write as "0"	RW
b1~b0	ECCMOD[1:0]	ECC check enable bit of message RAM	00: disable ECC check function 01: If 1 bit error, ECC error correction, No 1-bit error flag is generated, and no interrupt is generated; If 2 bits are wrong, ECC error detection, Generates a 2-bit error flag and generates an interrupt. 10: If 1 bit error, ECC error correction, Generates a 1-bit error flag, but does not generate an interrupt; If 2 bits are wrong, ECC error detection, Generates a 2-bit error flag and generates an interrupt. 11: If 1 bit error, ECC error correction, Generate a 1-bit error flag and generate an interrupt; If 2 bits are wrong, ECC error detection, Generates a 2-bit error flag and generates an interrupt.	RW

Note:

– When ECCMOD is not set to 0, you must set ECCMOD before accessing data, and then initialize the RAM space used in words.

30.4.47 MCAN Message RAM Check Status Register (MCANRAM_CKSR)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res														RAM2ERR	RAM1ERR

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b2	Res	Reserved Bit	Read as "0", write as "0"	RW
b1	RAM_2ERR	Message RAM ECC 2-bit error flag	0: No 2-bit ECC error occurs 1: 2-bit ECC error occurs	RW (Note 1)
b0	RAM_1ERR	Message RAM ECC 1-bit error flag	0: No 1-bit ECC error occurs 1: 1-bit ECC error occurs	RW (Note 1)

Note 1: Writing 1 clears 0.

30.4.48 MCAN Message RAM Error Injection Enable Register (MCANRAM_EIEN)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res															EIEN

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b1	Res	Reserved Bit	Read as "0", write as "0"	RW
b0	EIEN	Fault injection enable	0: Fault injection disabled 1: Error injection invalid	RW

30.4.49 MCAN Message RAM Error Injection Bit Enable Register 0 (MCANRA_EIBIT0)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
EIBIT[31:16]															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
EIBIT[15:0]															

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b0	EIBIT[31:0]	Error injection bit enable	Controls whether bits 0 to 31 of the message RAM output data are inverted 0: Not negated 1: Negate	RW

30.4.50 MCAN Message RAM Error Injection Bit Enable Register 1 (MCANRAM_EIBIT1)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res									EIBIT[38:32]						

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b7	Res	Reserved Bit	Read as "0", write as "0"	RW
b6~b0	EIBIT[38:32]	Error injection bit enable	Controls whether bits 32 to 38 of the message RAM output data are inverted 0: Not negated 1: Negate	RW

30.4.51 MCAN Message RAM Check Error Address Register (MCANRAM_ECCERRADDR)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res						ECCERRADDR [9:0]									

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b10	Res	Reserved Bit	Read as "0", write as "0"	RW
b9~b0	ECCERRADDR [8:0]	Checksum error address	Records the access address when a 1-bit or 2-bit ECC error occurs in the message RAM	R

30.5Precautions

30.5.1 CAN Usage Notes

Before using either CAN, you need to turn on the clocks of both CANs.

30.5.2 CAN Bus Anti-Jamming 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 the stable operation of the CAN bus network.

The following are several commonly used CAN bus anti-interference measures (including but not limited to)

1) Increase electrical isolation of CAN bus interface
2) Common transceiver signal ground
3) Use shielded twisted pair cables and ground them properly
4) Improve twisted pair degree of CAN transmission line
5) Add signal protector
6) Improve network topology
7) Application layer software anti-jamming mechanism

30.5.3 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 .

31 CRC Operation (CRC)

31.1 Main Features

- One implementation standard: ISO/ IEC13239
- Two encoding methods: CRC-16, CRC-32
- Three write bit widths: 8bit, 16bit, 32bit
- Two working modes: CRC encoding mode, CRC check mode
- The generating polynomial of CRC-16 is $x^{16}+x^{12}+x^5+1$, and the 16-bit initial value is 0xFFFF
- CRC-32 polynomial: $X^{32}+X^{26}+X^{23}+X^{22}+X^{16}+X^{12}+X^{11}+X^{10}+X^8+X^7+X^5+X^4+X^2+X+1$, and the initial value of 32 bits is 0xFFFF FFFF

31.2 Functional Description

The functions of this module include:

- CRC coding and CRC check;
- 3 Bitwidth access modes 8 bits, 16 bits, 32 bits:

Examples of 8-bit width input data are 0x00, 0x11, 0x22, 0x33, 0x44, 0x55, 0x66, 0x77;

Examples of 16 bit width input data are 0x1100, 0x3322, 0x5544, 0x7766;

The input data examples for 32-bit width are 0x3322 1100, 0x7766 5544;

31.2.1 Functional Block Diagram

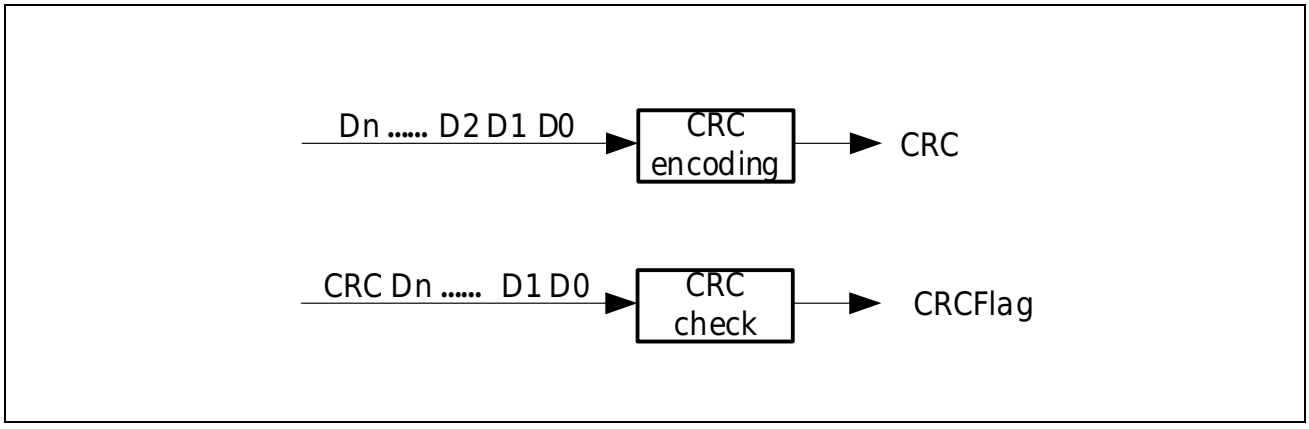


Figure 31-1 CRC Application Diagram

31.2.2 CRC16 Encoding Mode

CRC coding is to encode the original data to calculate its CRC value. The procedure is as follows:

1. Write 0b0 to CRC_CR.CR and select CRC16;
2. Write CRC_RESLT[15:0] and initialize CRC calculation.
3. Write the original data to be encoded to the CRC _ DATregister in 8-bit/16-bit/32-bit organization;

Note: Write CRC value according to 8-bit organization, write low bit first, write high bit later;

4. Read CRC_RESLT[15:0]to get 16-bit CRC coding value.

31.2.3 CRC 16 Check Mode

Check mode can check if the encoded data has been tampered with. The operation flow is as follows:

1. Write 0b0 to CRC_CR.CR and select CRC16;
2. Write CRC_RESLT[15:0] and initialize CRC calculation.
3. Write the data to be checked into the CRC_DAT register in 8-bit/16-bit/32-bit format.

   Note: Write CRC value according to 8-bit organization, write low bit first, write high bit later;
4. The checksum is written into the CRC_DAT register with a width of 8/16 bits.
5. Read CRC _ CR.FLGregister, 1 indicates check success, and 0 indicates check failure.

31.2.4 CRC32 Encoding Mode

CRC coding is to encode the original data to calculate its CRC value. The procedure is as follows:

1. Write 0b1 to CRC_CR.CR and select CRC32;
2. Write 0xFFFF FFFF to CRC_RESLT[31:0] to initialize CRC calculation;
3. Write the original data to be encoded to the CRC _ DATregister in 8-bit/16-bit/32-bit organization;

   Note: Write CRC value according to 8-bit organization, write low bit first, write high bit later;
4. Read CRC_RESLT[31:0]to get 32-bit CRC coding value.

31.2.5 CRC 32 check mode

Check mode can check if the encoded data has been tampered with. The operation flow is as follows:

1. Write 0b1 to CRC_CR.CR and select CRC32;
2. Write 0xFFFF FFFF to CRC_RESLT[31:0] to initialize CRC calculation;
3. Write the data to be checked into the CRC_DAT register in 8-bit/16-bit/32-bit format.

   Note: Write CRC value according to 8-bit organization, write low bit first, write high bit later;
4. The check code is organized in 8-bit/16-bit/32-bit mode and written into the CRC_DAT register;
5. Read CRC _ CR.FLGregister, 1 indicates check success, and 0 indicates check failure.

31.3Register Description

Table 31-1 CRC Base Address

Name	Base Address	Description
CRC	0x4000 8C00	CRC Base Address

Table 31-2 CRC Register List

Register	Description	Offset Address	Bit Width	Reset Value
CRC_CR	CRC control register	0x00	32	0x0000 0001
CRC_RESLT	CRC result register	0x04	32	0x0000 0000
CRC_DAT	CRC data register	0x80~0xFF	32	0x0000 0000

31.3.1 Control Register (CRC _ CR)

Reset value: 0x0000 0001

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res														FLAG	CR

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b30~b2	Res	Reserved Bit	Read at “0”, write at “0”	RW
b1	FLAG	Check result bit	0: Current check error 1: Current check is correct	R
b0	CR	Operational control bit	0：CRC16 1：CRC32	RW

31.3.2 Results Register (CRC _ RESLT)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
RESULT[31:16]															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
RESULT[15:0]															

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b0	RESULT[31:0]	Result bit	When CRC 16 is selected, RESULT [15: 0]; When CRC 32 was selected, RESULT [31: 0] was selected.	RW

31.3.3 Data Register (CRC _ DAT)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
CRC_DAT[31:16]															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
CRC_DAT[15:0]															

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b0	CRC_DAT[31:0]	Data Register	This register is used to input data to be computed; The address of this register is a range (0x80 ~ 0xFF), and it is considered to operate on any address in that range. This register reads as all 0s.	W

32 Debug Controller (DBGC)

This product refers to the following ARM technical documents:

- Cortex M4 Technical Reference Manual r0p1
- Arm Debug Interface Architecture Specification version 5
- CoreSight Components Technical Reference Manual version 1.0

32.1 Overview

The core of this MCU is Cortex-M4, which contains hardware for advanced debugging functions. With these debugging features, you can stop the kernel when fetching fingers (instruction breakpoints) or accessing data (data breakpoints). When the kernel stops, you can query the internal state of the kernel and the external state of the system. After the query completes, the kernel and system are restored and program execution is restored.

Two debugging interfaces are provided:

- Serial Debugging Tracking Interface SWD
- Parallel debug trace interface JTAG

32.2 DBGC System Block Diagram

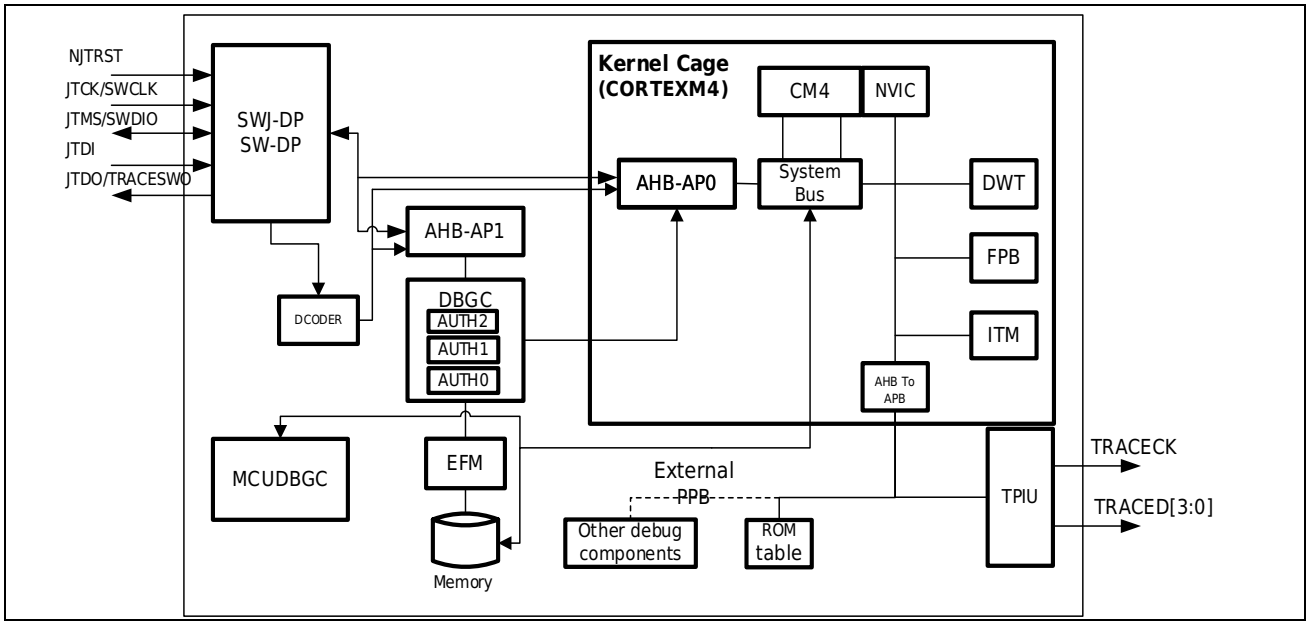


Figure 32-1 Commissioning Control System

ARM Cortex ™ -M4 + cores provide integrated on-chip debugging support. It includes:

- SWJ-DP: SWD/JTAG debug port
- AHB-AP: AHB Access Port
- ITM: Instruction Tracking Unit
- FPB: Flash instruction breakpoint

- DWT: Data breakpoint trigger
- TPIU: Trace Port Unit Interface (available on larger packages where corresponding pins are mapped)
- Flexible debugging pin assignment

Note:

– For more information on the debug features supported by the ARM Cortex-M4, see the Cortex-M4F-r0p1 Technical Reference Manual and the CoreSight Design Suite r0p1 Technical Reference Manual.

32.3SWJ-DP Debug Port (SWD and JTAG)

The MCU core integrates the SWD/JTAG debug port (SWJ-DP). This port is an ARM standard CoreSight debug port with JTAG-DP (5 pins) interface and SW-DP (2 pins) interface.

- The JTAG Debug Port (JTAG-DP) provides a 5-pin standard JTAG interface for connection to the AHB-AP port.
- The Serial Wire Debug Port (SW-DP) provides a 2-pin (clock+data) interface for connecting to the AHB-AP port.

In SWJ-DP, the 2 JTAG pins of SW-DP are multiplexed with some of the 5 JTAG pins of JTAG-DP.

In the figure below, JTDO reuses TRACESWO and TDO. This means that asynchronous trace can only be implemented on SW-DP, not on JTAG-DP.

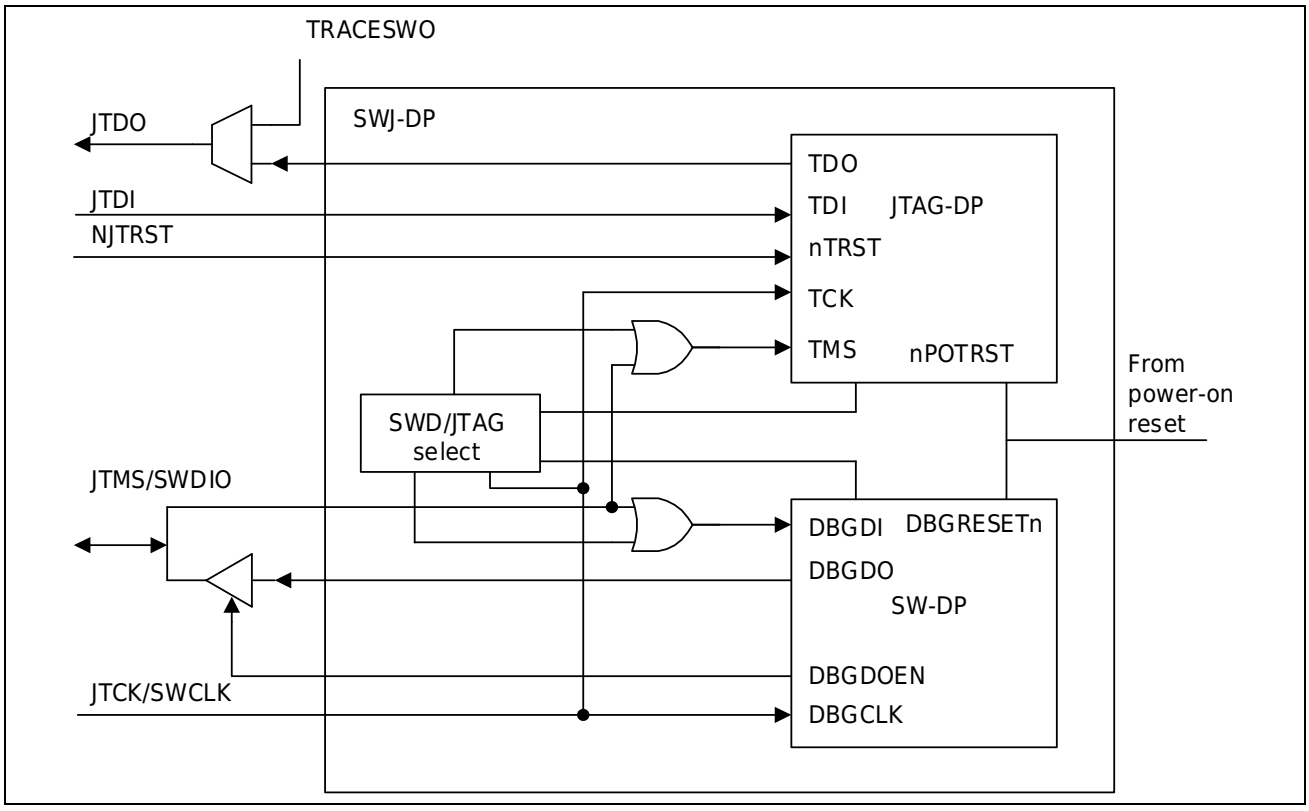


Figure 32-2 Commissioning Control System

32.3.1 Switching Mechanism of JTAG-DP or SW-DP

The default debug interface is the JTAG-DP interface.

If a debug tool wants to switch to SW-DP, it must provide a dedicated JTAG sequence on JTMS (SWDIO)/JTCK (SWCLK) for disabling JTAG-DP and enabling SW-DP. This allows access to the SW-DP using only the SWCLK and SWDIO pins.

The sequence is:

1) Output JTMS (SWDIO) = 1 signal for more than 50 JTCK cycles
2) Output 16 JTMS (SWDIO) signals 0111_1001_1110_0111 (MSB)
3) Output JTMS (SWDIO) = 1 signal for more than 50 JTCK cycles

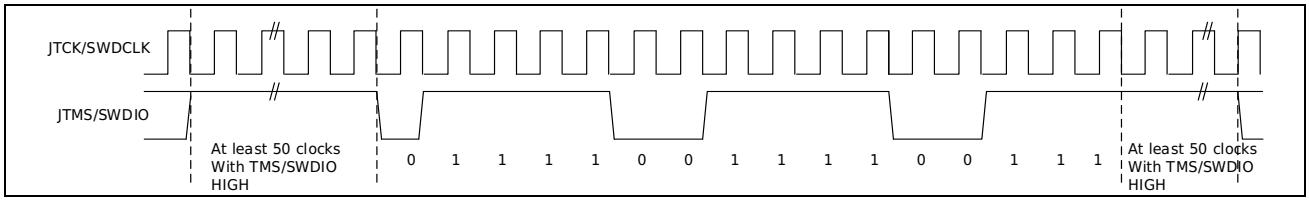


Figure 32-3 JTAG-DP to SW-DP Switching Timing

32.4Pinout and Debug Port Pinout

Depending on the MCU package, there are different effective pin counts. Therefore, some pin-related functions may vary from package to package.

32.4.1 SWJ Debug Port Pin

5 common I/O ports of MCU can be used as SWJ-DP interface pins.

Table 32-1 SWJ Debug Port Pins

SWJ-DP Pin Name	JTAG Debug Port		SW debug port	
	Types	Description	Types	Debug allocation
JTMS/SWDIO	Input	JTAG mode selection	Input output	Serial data input/output
JTCK/SWCLK	Input	JTAG Clock	Input	Serial clock
JTDI	Input	JTAG data input	-	-
JTDO/TRACESWO	Output	JTAG data output	-	TRACESWO (if asynchronous tracing is enabled)
NJTRST	Input	JTAG reset	-	-

32.4.2 Flexible SWJ-DP Pin Assignment

After reset (power-up or pin reset), all 5 pins for SWJ-DP are designated as dedicated pins, available for immediate use by debug tools (note that trace outputs are not assigned unless explicitly programmed) . However, the MCU can disable some or all of the SWJ-DP ports, thereby freeing the associated pins for use as GPIOs. For more details on how to disable the SWJ-DP port pin, see the chapter [Universal IO (GPIO)]Special Control Register PSPCR.

Table 32-2 Flexible SWJ-DP Pin Assignment

Available Debug Ports	Assigned SWJ IO Pins				
	JTMS/ SWDIO	JTCK/ SWCLK	JTDI	JTDO	NJTRST
All SWJ (JTAG-DP+SW-DP) - reset state	✓	✓	✓	✓	✓
Disable JTAG-DP and enable SW-DP	✓	✓	Releasable	Releasable	Releasable
Disable JTGA-DP and Disable SW-DP	Releasable	Releasable	Releasable	Releasable	Releasable

32.4.3 Internal Pull-ups on JTAG Pins

According to the JTAG IEEE standard, it must be ensured that the JTAG input pins are not floating because these pins are directly connected to the MCU internals for controlling the debug functions. Special attention must also be paid to the JTCK/SWCLK pin, which is used directly for the debug control clock function. In order to avoid IO level floating, the MCU has built-in internal pull-up resistors on the JTAG pins:

- NJTRST: Internal pull-up
- JTDI: Internal pull-up
- JTMS/SWDIO: internal pull-up
- JTCK/SWCLK: Internal pull-up
- JTDO: Internal pull-up

When the debugger is not connected, the user software can release the JTAG IO as a normal I/O port by setting the GPIO special control register. Since the internal pull-up of the chip is a weak pull-up of <100K ohms, it is recommended to use an external pull-up of 10K ohms.

32.4.4 Use the Serial Interface and Release Unused Debug Pins for GPIO

Some GPIOs can be released when using SWD, user software must change the GPIO configuration in the GPIO control register, so that the corresponding pins can be released as GPIO.

When debugging, the master does the following:

- In system reset state, assign all SWJ pins (JTAG-DP+SW-DP).
- In system reset state, the debug master sends a JTAG sequence to switch from JTAG-DP to SW-DP.
- Still in the system reset state, the debug master sets a breakpoint at the reset address.
- The reset signal is released and the core stops at the reset address.
- From this debug port switch to SW-DP. The other JTAG pins are then reassigned as GPIOs by user software.

Note:

– For user software design, when the debug pins need to be released, these pins are still in input pull-up (NJTRST, JTMS, JTDI, JTCK and JTDO) during the period after reset until the user software releases the pins.

32.5Register Description

Table 32-3 DBGC Base Address

Name	Base Address	Description
DBGC	0xE004 2000	DBGC Base Address

Table 32-4 DBGC Register List

Register	Description	Offset Address	Bit Width	Reset Value	Access master
CHIPID	Chip-specific flag register	0x000C	32	0x5848 0334	Debug IDE
MCUDBGSTAT	DBG status register	0x001C	32	0x0000 0000	CPU/Debug IDE*
MCUSTPCTL	Peripheral debugging pause register	0x0020	32	0x0000 003B	CPU/Debug IDE*
MCUTRACECTL	Debug Component Configuration Registers	0x0024	32	0x0000 0000	CPU/Debug IDE*
MCUSTPCTL2	Peripheral debugging pause register 2	0x0028	32	0x0000 0000	CPU/Debug IDE*

Note:

– Registers are located in the PPB area and can only be accessed by the CPU in privileged mode.

32.5.1 Chip Specific Identification Register (CHIPID)

Used to indicate company information and product model information.

Reset value: 0x5848 0334

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
CHIPID[31:16]															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
CHIPID[15:0]															

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b0	CHIPID[31:0]	Chip Specific Flag	CHIPID[31:16]: The corresponding ASCII code of XH, 0x5848 CHIPID[15:0]: Product model, 0x0334	R

32.5.2 DBG State Register (MCUDBGSTAT)

DBG debugs power-on status check register.

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res														CDBGPWRUPACK	CDBGPWRUPREQ

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b2	Res	Reserved Bit	Read at “0”, write at “0”	RW
b1	CDBGPWRUPACK	Power-on feedback of the debugger	0: No feedback 1: Commissioning Power-on Feedback	RW
b0	CDBGPWRUPREQ	Debugger power-on request	0: No power-on request 1: Power-on request	RW

32.5.3 Peripheral Debugging Pause Register (MCUSTPCTL)

The peripheral module pauses control when the CPU is in the debug state.

Reset value: 0x0000 003B

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res							M24STP	M23STP	M22STP	M21STP	M20STP	M19STP	M18STP	M17STP	M16STP
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
M15STP	Res		M12STP	Res				M07STP	M06STP	Res			RTCSTP	WDTSTP	SWDTSTP

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b25	Res	Reserved Bit	Read at “0”, write at “0”	RW
b24	M24STP	TMR6_4 Count pause signal	0: Counter still counts even if the kernel stops 1: Counter pause counting when kernel stops	RW
b23	M23STP	TMR6_3 Count pause signal	0: Counter still counts even if the kernel stops 1: Counter pause counting when kernel stops	RW
b22	M22STP	TMR6_2 Count pause signal	0: Counter still counts even if the kernel stops 1: Counter pause counting when kernel stops	RW
b21	M21STP	TMR6_1 Count pause signal	0: Counter still counts even if the kernel stops 1: Counter pause counting when kernel stops	RW
b20	M20STP	HRPWM_6 Count pause signal	0: Counter still counts even if the kernel stops 1: Counter pause counting when kernel stops	RW
b19	M19STP	HRPWM_5 Count pause signal	0: Counter still counts even if the kernel stops 1: Counter pause counting when kernel stops	RW
b18	M18STP	HRPWM_4 Count pause signal	0: Counter still counts even if the kernel stops 1: Counter pause counting when kernel stops	RW
b17	M17STP	HRPWM_3 Count pause signal	0: Counter still counts even if the kernel stops 1: Counter pause counting when kernel stops	RW
b16	M16STP	HRPWM_2 Count pause signal	0: Counter still counts even if the kernel stops 1: Counter pause counting when kernel stops	RW
b15	M15STP	HRPWM_1 Count pause signal	0: Counter still counts even if the kernel stops 1: Counter pause counting when kernel stops	RW
b14~ b13	Res	Reserved Bit	Read at “0”, write at “0”	RW
b12	M12STP	TMR4 Count pause signal	0: Counter still counts even if the kernel stops 1: Counter pause counting when kernel stops	RW
b11~b8	Res	Reserved Bit	Read at “0”, write at “0”	RW
b7	M07STP	TMR0_2 Count pause signal	0: Counter still counts even if the kernel stops 1: Counter pause counting when kernel stops	RW
b6	M06STP	TMR0_1 Count pause signal	0: Counter still counts even if the kernel stops 1: Counter pause counting when kernel stops	RW
b5~b3	Res	Reserved Bit	All bits are "1" when reading, and all bits are written "1" when writing	RW
b2	RTCSTP	RTC count pause signal	0: RTC counter still counts even if the kernel is stopped 1: RTC counter pause count when kernel stops	RW
b1	WDTSTP	WDT count pause signal	0: WDT counter still counts even if the kernel is stopped 1: WDT counter pause count when kernel stops	RW
b0	SWDTSTP	SWDT count pause signal	0: SWDTcounter still counts even if the kernel stops 1: SWDTcounter pause count when kernel stops	RW

32.5.4 Debug Component Configuration Register (MCUTRACECTL)

Configure the TRACE output pin through this register.

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res													TRACE_IOEN	TRACE_MODE	

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b3	Res	Reserved Bit	Read at “0”, write at “0”	RW
b2	TRACE_IOEN	TRACE pin output control	0: Synchronous tracking pin output disabled 1: Synchronous trace pin output permission	RW
b1~b0	TRACE_MODE	TRACED output pin control	00: asynchronous tracking 01： Sync tracking 1 bit TRACED[0] 10: Synchronous tracking 2-bit TRACED[1:0] 11: Synchronous tracking 4-bit TRACED[3:0]	RW

32.5.5 Peripheral Debugging Pause Register 2 (MCUSTPCTL2)

The peripheral module pauses control when the CPU is in the debug state.

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res											M36STP	M35STP	M34STP	M33STP	M32STP

Bit/Bitfield	Marking	Bit Name	Function	Read and Write
b31~b15	Res	Reserved Bit	Read at “0”, write at “0”	RW
b4	M36STP	TMRA_5 Count pause signal	0: Counter still counts even if the kernel stops 1: Counter pause counting when kernel stops	RW
b3	M35STP	TMRA_4 Count pause signal	0: Counter still counts even if the kernel stops 1: Counter pause counting when kernel stops	RW
b2	M34STP	TMRA_3 Count pause signal	0: Counter still counts even if the kernel stops 1: Counter pause counting when kernel stops	RW
b1	M33STP	TMRA_2 Count pause signal	0: Counter still counts even if the kernel stops 1: Counter pause counting when kernel stops	RW
b0	M32STP	TMRA_1 Count pause signal	0: Counter still counts even if the kernel stops 1: Counter pause counting when kernel stops	RW

32.6SW Debug Port

32.6.1 Introduction to SW Agreement

This synchronous serial protocol uses two pins:

- SWCLK: clock from master to slave
- SWDIO: Bidirectional

LSB is ahead when transferring data.

For SWCLK and SWDIO, the line needs to be pulled up on the circuit board (recommended to use 10K ohms).

32.7TPIU (Trace Port Interface Unit)

32.7.1 Introduction

The TPIU is the bridge between the ITM and the on-chip trace data.

The output data stream is encapsulated into a trace source ID, which is then captured by the trace port analyzer (TPA).

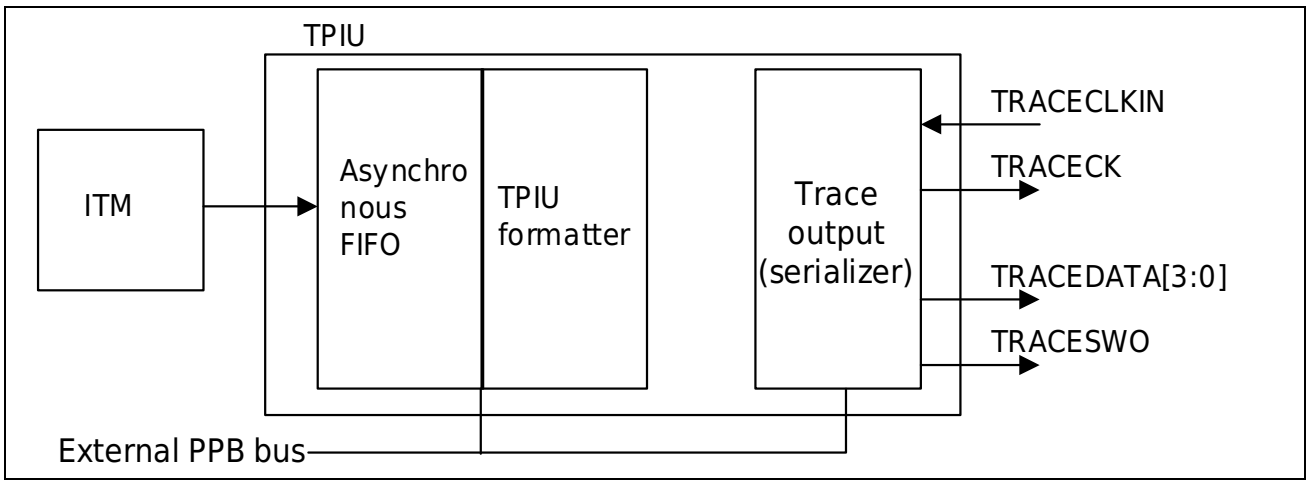


Figure 32-4 TPIU Block Diagram

32.7.2 TRACE Pin Allocation

- Asynchronous mode

Asynchronous mode requires 1 extra pin and is available in all packages. Asynchronous mode is only available when using serial line mode (not available in JTAG mode).

TPIU Pin Name	Track Asynchronous Mode	
	Types	Description
TRACESWO	Output	TRACE asynchronous data output

- synchronous mode

Synchronous mode requires 2 to 5 extra pins, depending on the length of the data being traced, and is only available on larger packages. In addition, synchronous mode is available in both JTAG mode and serial mode, and can provide higher bandwidth output capability than asynchronous trace.

TPIU Pin Mame	Track Sync Mode	
	Types	Description
TRACECK	Output	TRACE clock
TRACED[3:0]	Output	TRACE synchronous data output, can be 1, 2 or 4.

TPIU TRACE pin assignment

By default, these pins are not assigned. These pins can be configured by setting the TRACE_IOEN and TRACE_MODE bits in the MCU Debug Component Configuration Register (MCUTRACECTL). This configuration must be done by the debug master or CPU.

Also, the number of pins to assign depends on the trace configuration (asynchronous trace or synchronous trace).

- Asynchronous mode: requires 1 extra pin
- Synchronous mode: 5 additional pins required
  - TRACECK
  - TRACED[0] (if the port data length is configured as 1, 2 or 4)
  - TRACED[1] (if the port data length is configured as 2 or 4)
  - TRACED[2] (if the port data length is configured as 4)
  - TRACED[3] (if the port data length is configured as 4)

To assign the TRACE pins, the debug master must program bits TRACE_IOEN and TRACE_MODE[1:0] of the MCU debug configuration register (MCUTRACECTL). By default the TRACE pin is not assigned.

This register is mapped on the external PPB bus and is reset by power-on (not pin reset). This register can be written by the debugger in the pin reset state.

TPIU Pin Usage	Assigned TRACE IO Pins					
	JTDO/TRACESWO	TRACECK	TRACED[0]	TRACED[1]	TRACED[2]	TRACED[3]
No tracking (default state) TRACE_IOEN =0 TRACE_MODE=XX	Freed*	Freed	Freed	Freed	Freed	Freed
Asynchronous tracking TRACE_IOEN =1 TRACE_MODE=00	TRACESWO	Freed	Freed	Freed	Freed	Freed
Sync tracking 1 bit TRACE_IOEN =1 TRACE_MODE=01	Freed*	TRACECK	TRACED[0]	Freed	Freed	Freed
Sync tracking 2 bit TRACE_IOEN =1 TRACE_MODE=10	Freed*	TRACECK	TRACED[0]	TRACED[1]	Freed	Freed

TPIU Pin Usage	Assigned TRACE IO Pins					
	JTDO/TRACESWO	TRACECK	TRACED[0]	TRACED[1]	TRACED[2]	TRACED[3]
Sync tracking 4 bit TRACE_IOEN =1 TRACE_MODE=11	Freed*	TRACECK	TRACED[0]	TRACED[1]	TRACED[2]	TRACED[3]

Note:

– *When using serial mode, release this pin. But when using JTAG, this pin is assigned to TDO.*

32.7.3 MCU Internal TRACECLKIN Connection

In this MCU, the clock TRACECLKIN of TPIU is connected to the internal clock. The default clock of the MCU is the internal MRC oscillator. The frequency in reset state is different from the frequency after reset release. The reason is that since the default MRC calibration value is adopted in the system reset state, the MRC calibration value will be updated every time the system reset is released. Therefore, the trace port analyzer (TPA) should not enable trace (using the TRACE_IOEN bit) in the system reset state, because the bit width of the sync frame packet in the reset state is different from the packet after reset.

32.7.4 TPIU Register

The TPIU APB register can only be read from or written to when bit TRCENA in the Debug Exception and Monitor Control Register (DEMCR) is set. Otherwise, these registers will read zero (the output of this bit enables the clock to the TPIU).

32.7.5 TPIU Configuration Example

- Set bit TRCENA in the Debug Exception and Monitor Control (DEMCR)
- Write desired value to TPIU current port size register (default 0x1 for 1-bit port size)
- Write 0x102 to TPIU formatter and refresh control register (default value)
- Write TPIU select pin protocol to select synchronous mode or asynchronous mode. Example: 0x2 means asynchronous NRZ mode (similar to USART)
- Write 0x20 to the MCUTRACECTL control register (bit TRACE_IOEN) to assign TRACE I/O for asynchronous mode.
- Send TPIU synchronization packet at this time (FF_FF_FF_7F)
- Configure the ITM and write to the ITM stimulus register to output a value

Version Revision History

Version Number	Revision Date	Modified Content
Rev1.00	2025/11/27	First edition release.